



Oracle VM VirtualBox®

Manuel de l'utilisateur

Version 4.2.51

© 2004-2013 Oracle Corporation

<http://www.virtualbox.org>

Contents

1	Premiers pas	11
1.1	À quoi sert la virtualisation ?	12
1.2	Un peu de terminologie	12
1.3	Aperçu des fonctionnalités	13
1.4	Systèmes d'exploitation hôtes supportés	16
1.5	Installer VirtualBox et les packs d'extension	17
1.6	Démarrer VirtualBox	18
1.7	Créer votre première machine virtuelle	19
1.8	Lancer votre machine virtuelle	22
1.8.1	Démarrage d'une nouvelle VM pour la première fois	22
1.8.2	Capturer et rendre le clavier et la souris	23
1.8.3	Taper des caractères spéciaux	24
1.8.4	Changer de média amovible	25
1.8.5	Redimensionner la fenêtre de la machine	25
1.8.6	Sauvegarder l'état de la machine	26
1.9	Utiliser des groupes de VMs	27
1.10	Instantanés	28
1.10.1	Prendre, restaurer et effacer des instantanés	29
1.10.2	Contenu d'un instantané	30
1.11	Configuration d'une machine virtuelle	31
1.12	Supprimer des machines virtuelles	32
1.13	Cloner des machines virtuelles	32
1.14	Importer et exporter des machines virtuelles	33
1.15	Paramètres globaux	35
1.16	Interfaces alternatives	35
2	Détails sur l'installation	37
2.1	Installation sur des hôtes Windows	37
2.1.1	Prérequis	37
2.1.2	Effectuer l'installation	37
2.1.3	Désinstallation	38
2.1.4	Installation sans efforts	38
2.2	Installation sur des hôtes Mac OS X	39
2.2.1	Effectuer l'installation	39
2.2.2	Désinstallation	39
2.2.3	Installation sans efforts	39
2.3	Installing sur des hôtes Linux	39
2.3.1	Prérequis	39
2.3.2	Le module noyau de VirtualBox	40
2.3.3	Effectuer l'installation	41
2.3.4	Le groupe vboxusers	44
2.3.5	Démarrer VirtualBox sur Linux	45
2.4	Installation sur les hôtes Solaris	45
2.4.1	Effectuer l'installation	45
2.4.2	Le groupe vboxuser	46
2.4.3	Démarrer VirtualBox sur Solaris	46

Contents

2.4.4	Désinstallation	46
2.4.5	Installation sans efforts	46
2.4.6	Configurer un espace pour exécuter VirtualBox	47
3	Configurer des machines virtuelles	48
3.1	Systèmes d'exploitation invités supportés	48
3.1.1	Inviés Mac OS X	49
3.1.2	Invités 64 bits	49
3.2	Matériel émulé	50
3.3	Paramètres généraux	51
3.3.1	Onglet “Base”	51
3.3.2	Onglet “Avancé”	51
3.3.3	Onglet “Description”	52
3.4	Paramètres système	52
3.4.1	Onglet “Carte mère”	52
3.4.2	Onglet “Processeur”	54
3.4.3	Onglet “Accélération”	54
3.5	Paramètres d'affichage	55
3.6	Paramètres du stockage	56
3.7	Paramètres de son	58
3.8	Paramètres réseau	58
3.9	Ports série	59
3.10	Support USB	60
3.10.1	Paramètres USB	60
3.10.2	Notes d'implémentation pour les hôtes Windows et Linux	62
3.11	Dossiers partagés	62
3.12	Autre firmware (EFI)	62
3.12.1	Modes graphiques dans EFI	63
4	Les suppléments invité	64
4.1	Introduction	64
4.2	Installer et maintenir les suppléments invité	66
4.2.1	Suppléments invité pour Windows	66
4.2.2	Suppléments invité pour Linux	69
4.2.3	Suppléments invité pour Solaris	73
4.2.4	Suppléments invité pour OS/2	74
4.3	Dossiers partagés	75
4.3.1	Montage manuel	76
4.3.2	Montage automatique	77
4.4	L'accélération graphique matérielle	77
4.4.1	Accélération 3D matérielle (OpenGL and Direct3D 8/9)	77
4.4.2	L'accélération matérielle 2D pour les invités Windows	79
4.5	Fenêtres transparentes	79
4.6	Propriétés invité	80
4.7	Contrôle de l'invité	82
4.8	Transfert de mémoire	82
4.8.1	Faire du ballon avec la mémoire	83
4.8.2	Fusion de page	83
5	Le stockage virtuel	85
5.1	Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS	85
5.2	Fichiers images de disque (VDI, VMDK, VHD, HDD)	88
5.3	Le gestionnaire de médias virtuels	88
5.4	Modes spéciaux d'écriture d'images	90

Contents

5.5	Images de différenciation	92
5.6	Cloner des images de disque	94
5.7	Mise en cache des E/S dans l'hôte	95
5.8	Limiter la bande passante des images de disque	96
5.9	Support des CD/DVD	96
5.10	Serveurs iSCSI	97
6	Le réseau virtuel	98
6.1	Matériel réseau virtuel	98
6.2	Introduction aux modes réseaux	99
6.3	Network Address Translation (NAT)	100
6.3.1	Configurer la redirection de ports avec NAT	100
6.3.2	Démarrer avec PXE avec NAT	101
6.3.3	Limites du NAT	101
6.4	Réseau Bridged	102
6.5	Réseau interne	103
6.6	Réseau Host-only	104
6.7	Réseau en tunnel UDP	105
6.8	Réseau VDE	106
6.9	Limiter la bande passante des E/S réseaux	107
6.10	Améliorer les performances réseaux	108
7	Machines virtuelles distantes	109
7.1	Affichage distant (VRDP support)	109
7.1.1	Visualiseurs RDP tiers classiques	109
7.1.2	VBoxHeadless, le serveur de bureau distant	110
7.1.3	Pas à pas : créer une machine virtuelle sur un serveur headless	111
7.1.4	USB distant	113
7.1.5	Authentification RDP	113
7.1.6	Chiffrement RDP	115
7.1.7	Connexions multiples au serveur VRDP	116
7.1.8	Avoir plusieurs moniteurs distants	116
7.1.9	Redirection graphique VRDP	116
7.1.10	Personnalisation du VRDP	117
7.2	Téléportation	117
8	VBoxManage	119
8.1	Introduction	119
8.2	Aperçu des commandes	120
8.3	Options générales	127
8.4	VBoxManage list	127
8.5	VBoxManage showvminfo	128
8.6	VBoxManage registervm / unregistervm	129
8.7	VBoxManage createvm	130
8.8	VBoxManage modifyvm	130
8.8.1	Paramètres généraux	130
8.8.2	Paramètres réseaux	133
8.8.3	Options du port série, du son, du presse-papier, du bureau distant et de l'USB	135
8.8.4	Paramètres de la machine distante	136
8.8.5	Paramètres de téléportation	136
8.9	VBoxManage clonevm	137
8.10	VBoxManage import	138
8.11	VBoxManage export	139

Contents

8.12	VBoxManage startvm	139
8.13	VBoxManage controlvm	140
8.14	VBoxManage discardstate	141
8.15	VBoxManage adoptstate	141
8.16	VBoxManage snapshot	141
8.17	VBoxManage closemedium	142
8.18	VBoxManage storageattach	142
8.19	VBoxManage storagectl	145
8.20	VBoxManage bandwidthctl	145
8.21	VBoxManage showhdinfo	146
8.22	VBoxManage createhd	146
8.23	VBoxManage modifyhd	147
8.24	VBoxManage clonehd	148
8.25	VBoxManage convertfromraw	149
8.26	VBoxManage getextradata/setextradata	149
8.27	VBoxManage setproperty	150
8.28	VBoxManage usbfilter add/modify/remove	150
8.29	VBoxManage sharedfolder add/remove	151
8.30	VBoxManage guestproperty	151
8.31	VBoxManage guestcontrol	152
8.32	VBoxManage debugvm	156
8.33	VBoxManage metrics	157
8.34	VBoxManage hostonlyif	158
8.35	VBoxManage dhcpserver	158
8.36	VBoxManage extpack	159
9	Sujets avancés	161
9.1	VBoxSDL, l'afficheur simplifié de VM	161
9.1.1	Introduction	161
9.1.2	Étiquetage sécurisé avec VBoxSDL	161
9.1.3	Libérer les modificateurs avec VBoxSDL sur Linux	162
9.2	Identifications automatiques dans l'invité	163
9.2.1	Identification automatique dans un invité Windows	163
9.2.2	Identifications automatisées à un invité Linux/Unix	164
9.3	Configuration avancées pour les invités Windows	168
9.3.1	Préparation automatique du système Windows	168
9.4	Configuration avancée pour les invités Linux et Solaris	168
9.4.1	Paramétrage manuel des services sélectionnés sur l'invité Linux	168
9.4.2	Paramétrage approfondi des pilotes graphique et souris de l'invité	169
9.5	Montage de processeur à chaud	170
9.6	PCI passthrough	171
9.7	Configuration d'affichage avancée	172
9.7.1	Résolutions VESA personnalisées	172
9.7.2	Configuration de la résolution maximum des invités quand on utilise l'interface graphique	173
9.8	Configuration avancée du stockage	173
9.8.1	Utiliser un disque dur brut de l'hôte à partir de l'invité	173
9.8.2	Configuration des vendor product data (VPD) du disque dur	175
9.8.3	Accès à des cibles iSCSI via le réseau interne	176
9.9	Lancer plus de 128 VMs sur des hôtes Linux	177
9.10	Lancer plus de 120 VMs sur les hôtes Solaris	177
9.10.1	Solution temporaire quand VirtualBox est en fonction	177
9.10.2	Solution permanente, exige que l'utilisateur se re-connecte	177
9.11	Commandes de base pour utiliser les ports série	178

Contents

9.12	Peaufiner le moteur NAT de VirtualBox	178
9.12.1	Configurer l'adresse d'une interface réseau NAT	178
9.12.2	Configurer le serveur d'amorçage (prochain serveur) d'une interface réseau NAT	179
9.12.3	Peaufiner les tampons TCP/IP pour NAT	179
9.12.4	Associer des sockets à une interface spécifique	179
9.12.5	Activer le proxy DNS en mode NAT	180
9.12.6	Utiliser le résolveur de l'hôte comme proxy DNS en mode NAT	180
9.12.7	Configurer des aliases pour le moteur NAT	181
9.13	Configurer les informations DMI du BIOS	181
9.13.1	Informations DMI du BIOS (type 0)	181
9.13.2	Informations système DMI (type 1)	181
9.13.3	Informations carte mère DMI (type 2)	182
9.13.4	Boîtier système DMI ou chassis (type 3)	182
9.13.5	Informatiions DMI du processeur (type 4)	182
9.13.6	Chaînes OEM DMI (type 11)	182
9.14	Configurer la table ACPI personnalisée	183
9.15	Peaufiner les horloges et la synchronisation du temps	183
9.15.1	Configurer le time stamp counter (TSC) (horodateur) de l'invité pour refléter l'heure de l'exécution	183
9.15.2	Accélérer ou ralentir l'horloge de l'invité	183
9.15.3	Peaufiner les paramètres de synchronisation du temps des suppléments invité	184
9.15.4	Désactiver la synchronisation des suppléments invité	184
9.16	Installer le pilote du réseau bridgé alternatif sur les invités Solaris 11	184
9.17	Échantillons de VNIC VirtualBox pour les VLANs sur les hôtes Solaris 11	185
9.18	Configurer plusieurs interfaces réseaux host-only sur les hôtes Solaris	186
9.19	Configurer le CoreDumper sur les hôtes Solaris	186
9.20	Déverrouiller l'interface graphique du gestionnaire de VirtualBox	187
9.20.1	Personnalisation de l'interface graphique	187
9.20.2	Personnalisation de la touche hôte	187
9.20.3	Action puand la VM s'arrête	188
9.21	Démarrer le service Web de VirtualBox automatiquement	188
9.21.1	Linux : démarrer le service web via init	189
9.21.2	Solaris: démarrer le service web par SMF	189
9.21.3	Mac OS X : démarrer le service web par launchd	190
9.22	VirtualBox Watchdog	190
9.22.1	Contrôle du jeu de ballon de mémoire	190
9.22.2	Détection de l'isolement de l'hôte	191
9.22.3	Plus d'informations	192
9.22.4	Linux : démarrer le service watchdog via init	192
9.22.5	Solaris : démarrer le service watchdog via SMF	193
9.23	Autres packs d'extension	193
9.24	Démarrer des machines virtuelles lors de l'amorçage du système	194
9.24.1	Linux : démarrer le service autostart par init	194
9.24.2	Solaris : démarrer le service autostart par SMF	194
9.24.3	Mac OS X : démarrer le service autostart par launchd	195
10	Sous-basements techniques	196
10.1	Où VirtualBox stocke ses fichiers	196
10.1.1	Machines créées par VirtualBox version 4.0 ou supérieur	196
10.1.2	Machines créées par des versions de VirtualBox antérieures à 4.0	197
10.1.3	Données globales de configuration	197
10.1.4	Résumé des des modifications de la configuration de 4.0	198

10.1.5	Fichiers XML de VirtualBox	198
10.2	Exécutables et composants de VirtualBox	199
10.3	Virtualisation matérielle vs. logicielle	201
10.4	Détails sur la virtualisation logicielle	202
10.5	Détails sur la virtualisation matérielle	204
10.6	Vagation nestée et VPIDs	205
11	Interfaces de programmation de VirtualBox	207
12	Dépannage	208
12.1	Procédures et outils	208
12.1.1	Catégoriser et isoler des problèmes	208
12.1.2	Recueillir des informations de débogage	209
12.1.3	Le débogueur de VM intégré	210
12.1.4	Format du cœur d'une VM	212
12.2	Général	212
12.2.1	L'invité affiche des erreurs IDE/SATA pour les images basées sur un fichier sur un système de fichiers hôte lent	212
12.2.2	Réponse aux requêtes de flush IDE/SATA de l'invité	213
12.2.3	Faibles performances dues à la gestion d'énergie de l'hôte	214
12.2.4	GUI : l'option d'accélération graphique est grise	214
12.3	Invités Windows	214
12.3.1	Écrans bleus Windows après avoir changé la configuration d'une VM	214
12.3.2	Écran bleu sur Windows 0x101 si SMP est activé (IPI timeout)	215
12.3.3	Échecs d'installation de Windows 2000	215
12.3.4	Comment garder les informations d'un écran bleu des invités Windows	215
12.3.5	Pas de réseau dans les invités Windows Vista	216
12.3.6	Les invités Windows peuvent provoquer une forte charge du processeur	216
12.3.7	Temps d'accès élevés aux dossiers partagés	216
12.3.8	La tablette USB coordonne mal dans les invités Windows 98	216
12.3.9	Les invités Windows sont retirés du domaine Active Directory après la restauration d'un instantané	216
12.3.10	Restauration de d3d8.dll et de d3d9.dll	217
12.4	Invités Linux et X11	218
12.4.1	Les invités Linux peuvent entraîner une forte charge du processeur	218
12.4.2	Processeurs AMD Barcelona	218
12.4.3	Versions bugguées du noyau Linux Linux 2.6	218
12.4.4	Presse-papier partagé, redimensionnement automatique et bureau transparent dans les invités X11	219
12.5	Invités Solaris	219
12.5.1	Les versions inférieures à Solaris 10 plantent en mode 64 bits	219
12.6	Hôte Windows	219
12.6.1	Problème du serveur VBoxSVC out-of-process COM	219
12.6.2	Changements de CD/DVD non reconnus	220
12.6.3	Réponse lente en utilisant le client RDP de Microsoft	220
12.6.4	Lancer un initiateur et une cible iSCSI sur un seul système	220
12.6.5	Adaptateurs réseaux bridgés absents	221
12.6.6	L'adaptateur réseau Host-only ne peut pas être créé	221
12.7	Hôtes Linux	221
12.7.1	Le module du noyau Linux refuse de se charger	221
12.7.2	Lecteur CD/DVD de l'hôte Linux non trouvé	221
12.7.3	Lecteur CD/DVD non trouvé sur l'hôte Linux (distributions anciennes)	222
12.7.4	Disquette non trouvée sur un hôte Linux	222

Contents

12.7.5	Messages d'erreur étranges de l'IDE invité lors de l'écriture sur un CD/DVD	222
12.7.6	Problème de l'IPC VBoxSVC	223
12.7.7	L'USB ne fonctionne pas	223
12.7.8	Noyaux PAX/grsec	224
12.7.9	pool vmalloc du noyau Linux dépassé	224
12.8	Hôtes Solaris	224
12.8.1	Ne peut pas démarrer de VM, pas assez de mémoire contiguë	224
12.8.2	La VM s'arrête avec des erreurs de dépassement de mémoire sur les hôtes Solaris 10	224
13	Guide de sécurité	226
13.1	Aperçu	226
13.1.1	Principes généraux de sécurité	226
13.2	Installation et configuration sécurisées	226
13.2.1	Aperçu de l'installation	226
13.2.2	Configuration post installation	227
13.3	Fonctions de sécurité	227
13.3.1	Le modèle de sécurité	227
13.3.2	Configuration sécurisée des machines virtuelles	227
13.3.3	Configurer et utiliser l'authentification	229
13.3.4	Opérations potentiellement non sécurisées	229
13.3.5	Chiffrement	229
14	Limites connues	231
14.1	Fonctions expérimentales	231
14.2	Problèmes connus	231
15	Historique des changements	235
15.1	Version xx.xx.0 (xxxx-xx-xx)	235
15.2	Version 4.2.0 (2012-09-13)	235
15.3	Version 4.1.18 (2012-06-06)	237
15.4	Version 4.1.16 (2012-05-22)	238
15.5	Version 4.1.14 (2012-04-13)	238
15.6	Version 4.1.12 (2012-04-03)	239
15.7	Version 4.1.10 (2012-03-13)	239
15.8	Version 4.1.8 (2011-12-19)	241
15.9	Version 4.1.6 (2011-11-04)	241
15.10	Version 4.1.4 (2011-10-03)	242
15.11	Version 4.1.2 (2011-08-15)	244
15.12	Version 4.1.0 (2011-07-19)	245
15.13	Version 4.0.14 (2011-10-13)	248
15.14	Version 4.0.12 (2011-07-15)	249
15.15	Version 4.0.10 (2011-06-22)	250
15.16	Version 4.0.8 (2011-05-16)	251
15.17	Version 4.0.6 (2011-04-21)	252
15.18	Version 4.0.4 (2011-02-17)	254
15.19	Version 4.0.2 (2011-01-18)	256
15.20	Version 4.0.0 (2010-12-22)	257
15.21	Version 3.2.12 (2010-11-30)	259
15.22	Version 3.2.10 (2010-10-08)	261
15.23	Version 3.2.8 (2010-08-05)	263
15.24	Version 3.2.6 (2010-06-25)	264
15.25	Version 3.2.4 (2010-06-07)	266

Contents

15.26 Version 3.2.2 (2010-06-02)	267
15.27 Version 3.2.0 (2010-05-18)	268
15.28 Version 3.1.8 (2010-05-10)	270
15.29 Version 3.1.6 (2010-03-25)	271
15.30 Version 3.1.4 (2010-02-12)	273
15.31 Version 3.1.2 (2009-12-17)	275
15.32 Version 3.1.0 (2009-11-30)	276
15.33 Version 3.0.12 (2009-11-10)	278
15.34 Version 3.0.10 (2009-10-29)	279
15.35 Version 3.0.8 (2009-10-02)	280
15.36 Version 3.0.6 (2009-09-09)	281
15.37 Version 3.0.4 (2009-08-04)	284
15.38 Version 3.0.2 (2009-07-10)	285
15.39 Version 3.0.0 (2009-06-30)	286
15.40 Version 2.2.4 (2009-05-29)	288
15.41 Version 2.2.2 (2009-04-27)	290
15.42 Version 2.2.0 (2009-04-08)	291
15.43 Version 2.1.4 (2009-02-16)	293
15.44 Version 2.1.2 (2009-01-21)	295
15.45 Version 2.1.0 (2008-12-17)	298
15.46 Version 2.0.8 (2009-03-10)	299
15.47 Version 2.0.6 (2008-11-21)	300
15.48 Version 2.0.4 (2008-10-24)	301
15.49 Version 2.0.2 (2008-09-12)	302
15.50 Version 2.0.0 (2008-09-04)	304
16 Matériaux tiers et licences	305
16.1 Matérp,1x	305
16.2 Licences	307
16.2.1 GNU General Public License (GPL)	307
16.2.2 GNU Lesser General Public License (LGPL)	311
16.2.3 Mozilla Public License (MPL)	316
16.2.4 MIT License	322
16.2.5 X Consortium License (X11)	322
16.2.6 zlib license	322
16.2.7 OpenSSL license	323
16.2.8 Slirp license	323
16.2.9 liblzf license	324
16.2.10 libpng license	324
16.2.11 lwIP license	325
16.2.12 libxml license	325
16.2.13 libxslt licenses	325
16.2.14 gSOAP Public License Version 1.3a	326
16.2.15 Chromium licenses	331
16.2.16 curl license	333
16.2.17 libgd license	333
16.2.18 BSD license from Intel	334
16.2.19 libjpeg License	334
16.2.20 x86 SIMD extension for IJG JPEG library license	335
16.2.21 FreeBSD license	336
16.2.22 NetBSD license	336
17 Informations sur la confidentialité de VirtualBox	337

Contents

Glossary

338

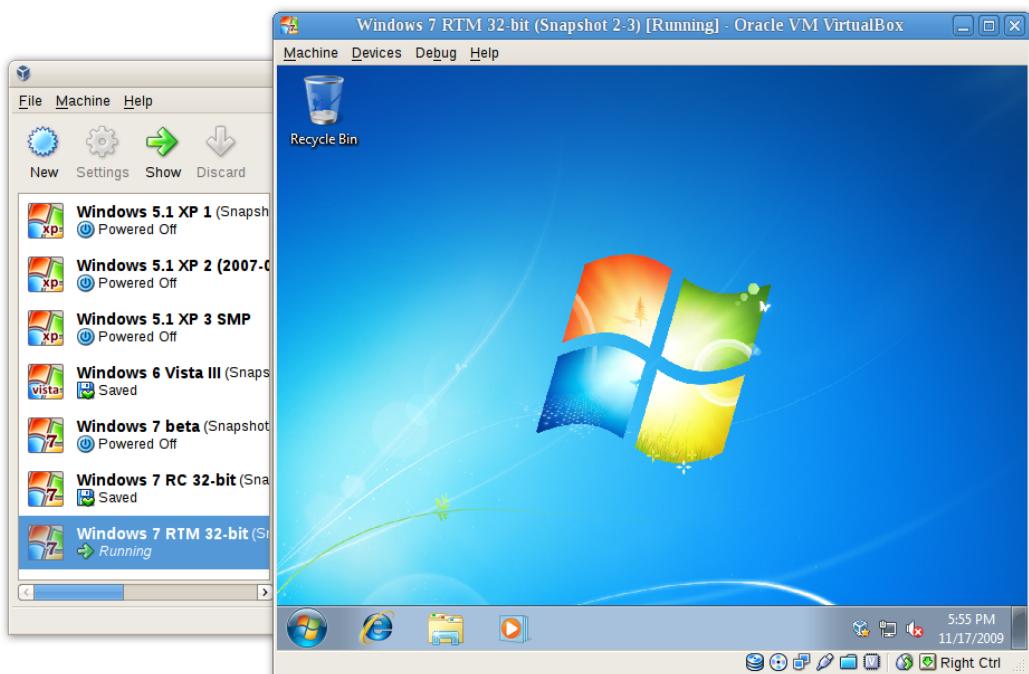
1 Premiers pas

Bienvenue à Oracle VM VirtualBox!

VirtualBox est une application de virtualisation de plateformes croisées. Qu'est-ce que cela veut dire ? D'une part, il s'installe sur vos ordinateurs existants basés sur Intel ou AMD, qu'ils soient sous les systèmes d'exploitation Windows, Mac, Linux ou Solaris. D'autre part, il augmente la capacité de votre ordinateur existant pour qu'il puisse lancer plusieurs systèmes d'exploitation en même temps (dans plusieurs machines virtuelles). Donc, vous pouvez par exemple lancer Windows and Linux sur votre Mac, lancer Windows Server 2008 sur votre serveur Linux, lancer Linux sur votre PC Windows and ainsi de suite pour toutes vos applications. Vous pouvez installer and lancer autant de machines virtuelles que vous voulez – la seule limite pratique étant votre espace disque and la mémoire.

VirtualBox est résolument simple bien que très puissant. Il peut se lancer partout, depuis de petits systèmes embarqués jusqu'aux machines de bureau en passant par des déploiements en datacenter ou même des environnements en nuages.

L'impression d'écran suivante vous montre la manière dont VirtualBox lance Windows 7 dans la fenêtre d'une machine virtuelle en étant installé sur un Mac :



Dans ce manuel de l'utilisateur, ne allons simplement commencer par une introduction rapide à la virtualisation and sur la façon de lancer votre première machine virtuelle avec l'interface graphique de VirtualBox, facile à utiliser. Les chapitres suivants entreront beaucoup plus dans les détails en traitant d'outils and de fonctionnalités plus puissants, mais heureusement, il n'est pas nécessaire de lire tout le manuel de l'utilisateur avant de pouvoir utiliser VirtualBox.

Vous pouvez trouver un résumé des possibilités de VirtualBox au chapitre [1.3, Aperçu des fonctionnalités](#), page [13](#). Pour ceux qui utilisent déjà VirtualBox and qui voudraient seulement voir les nouveautés de cette version, une liste détaillée existe au chapitre [15, Historique des changements](#), page [235](#).

1.1 À quoi sert la virtualisation ?

Les techniques and les fonctionnalités offertes par VirtualBox servent dans plusieurs scenari :

- **Lancer plusieurs systèmes d'exploitation en même temps.** VirtualBox vous permet d'exécuter plus d'un système d'exploitation en même temps. De cette façon, vous pouvez lancer des logiciels écrits pour un système d'exploitation dans un autre (par exemple un logiciel Windows sur Linux ou Mac) sans devoir redémarrer pour l'utiliser. Comme vous pouvez configurer les types de matériels “virtuels” connectés à chaque système d'exploitation, vous pouvez installer un vieux système d'exploitation tel que DOS ou OS/2 même si le matériel de votre machine physique n'est plus supporté par ce système d'exploitation.
- **Installation plus facile de logiciels.** Les éditeurs de logiciels peuvent utiliser des machines virtuelles pour y mettre des configurations de logiciels. Par exemple, l'installation d'une solution complète de serveur de messagerie sur une vraie machine peut être une tâche très fastidieuse. Avec VirtualBox, vous pouvez emballer une configuration aussi complexe (appelé alors souvent un environnement applicatif) dans une machine virtuelle. L'installation and l'exécution d'un serveur de messagerie devient aussi facile que d'importer un environnement applicatif dans VirtualBox.
- **Tester and réparer une récupération d'accident.** Une fois installés, on peut considérer une machine virtuelle and ses disques durs virtuels comme un “conteneur” qu'on peut, au choix, geler, réveiller, copier, sauvegarder and transporter entre hôtes. Au-delà de cela, en utilisant une autre fonctionnalité de VirtualBox appelé les “instantanés”, vous pouvez sauvegarder un état en particulier d'une machine virtuelle and revenir à cet état si nécessaire. De cette manière, vous pouvez librement essayer un environnement informatique. Si quelque chose ne va pas (par exemple, si un logiciel ne se comporte pas bien après l'installation ou si un invité a un virus), on peut facilement revenir à un instantané récent and éviter de sauvegarder and de restaurer fréquemment. On peut créer autant d'instantanés que vous voulez, ce qui permet de voyager dans le temps des machines virtuelles en arrière and en avant. Vous pouvez effacer des instantanés alors qu'une machine virtuelle est en fonction pour gagner de l'espace disque.
- **Consolider une infrastructure.** La virtualisation peut réduire significativement les coûts en matériel and électricité. La plupart du temps, les ordinateurs actuels n'utilisent qu'une partie de leur puissance potentielle and tournent avec une faible charge système moyenne. On gaspille donc beaucoup de ressources matérielles and énergétiques. Am lieu de lancer autant d'ordinateurs physiques, qui ne sont que partiellement utilisés, on peut emballer de nombreuses machines virtuelles sur quelques hôtes puissants and équilibrer les charges entre elles.

1.2 Un peu de terminologie

Quand on parle de virtualisation (mais aussi pour comprendre les chapitres suivants de cette documentation), acquérir un peu de terminologie fondamentale aide, en particulier les termes suivants :

Système d'exploitation hôte (OS hôte). C'est le système d'exploitation de l'ordinateur physique sur lequel VirtualBox a été installé. Il existe des versions de VirtualBox pour des hôtes Windows, Mac OS X, Linux and Solaris ; pour des détails, merci de voir le chapitre [1.4, Systèmes d'exploitation hôtes supportés](#), page 16.

La plupart du temps, ce manuel de l'utilisateur traite de toutes les versions de VirtualBox ensemble. Il peut y avoir des différences entre les plateformes, nous les soulignerons là où il faut.

Système d'exploitation invité (OS invité). C'est le système d'exploitation en fonction dans la machine virtuelle. En théorie, VirtualBox peut lancer tous les systèmes d'exploitation x86 (DOS, Windows, OS/2, FreeBSD, OpenBSD), mais pour s'approcher le plus possible de la performance d'origine du code invité sur votre machine, nous avons dû faire beaucoup d'optimisations qui sont spécifiques à certains systèmes d'exploitation. Donc si votre système d'exploitation favori peut être un invité, nous en supportons et optimisons officiellement quelques-uns sélectionnés (cependant, cela inclut ceux les plus courants).

Voir le chapitre [3.1, Systèmes d'exploitation invités supportés](#), page [48](#) pour des détails.

Machine virtuelle (VM). C'est l'environnement spécial créé par VirtualBox pour votre système d'exploitation invité qui s'exécute. Autrement dit, vous lancer votre système d'exploitation invité "dans" une VM. Normalement, une VM apparaîtra sur le bureau de votre ordinateur comme une fenêtre, mais elle peut être affichée en mode plein écran, ou à distance sur un autre ordinateur, selon les interfaces de VirtualBox que vous utilisez.

De manière plus abstraite, en interne, VirtualBox conçoit une VM comme un ensemble de paramètres qui déterminent son comportement. Parmi eux, on a les paramètres matériels (combien de mémoire devrait avoir la VM, quels disques durs devrait virtualiser VirtualBox via quels fichiers conteneurs, quels CDs, lesquels sont montés, etc.) ainsi que des informations d'état (si la machine est en fonction, sauvegardée, ses instantanés, etc.). Ces paramètres apparaissent dans la fenêtre du gestionnaire de VirtualBox ainsi que par le programme en ligne de commande `VBoxManage` ; voir le chapitre [8, VBoxManage](#), page [119](#). Autrement dit, une VM est aussi ce que vous pouvez voir dans la boîte de dialogue de ses paramètres.

Suppléments Invité. Ceci renvoie aux paquets logiciels spéciaux qui sont inclus avec VirtualBox mais conçus pour être installés à l'intérieur d'une VM pour améliorer les performances de l'OS invité and pour ajouter des fonctionnalités supplémentaires. C'est décrit en détails au chapitre [4, Les suppléments invité](#), page [64](#).

1.3 Aperçu des fonctionnalités

Voici un bref résumé des principales fonctionnalités de VirtualBox :

- **Portabilité.** VirtualBox se lance sur un grand nombre de systèmes d'exploitation hôtes 32 and 64 bits (de nouveau, voir chapitre [1.4, Systèmes d'exploitation hôtes supportés](#), page [16](#) pour les détails).

VirtualBox est ce qu'on appelle un hyperviseur "hébergé" (parfois désigné comme un hyperviseur de "type 2"). Alors qu'un hyperviseur "bare-metal" ou de "type 1" se lancerait directement sur le matériel, VirtualBox exige l'installation d'un système d'exploitation. Il peut alors lancer toutes les applications sur cet hôte.

Dans une très large mesure, VirtualBox est fonctionnellement identique sur toutes les plateformes hôtes and on peut utiliser les mêmes formats de fichiers and d'images. Ceci vous permet de lancer des machines virtuelles créées sur un hôte sur un autre hôte ayant un système d'exploitation hôte différent ; par exemple, vous pouvez créer une machine virtuelle sur Windows puis la lancer sous Linux.

En outre, on peut facilement importer and exporter des machines virtuelles en utilisant le format ouvert de virtualisation (OVF, voir chapitre [1.14, Importer et exporter des machines virtuelles](#), page [33](#)), un standard industriel créé dans ce but. Vous pouvez même importer des OVs qui ont été créés avec un logiciel de virtualisation différent.

- **Aucune virtualisation matérielle requise.** Pour de nombreux scénarios, VirtualBox n'exige pas la construction, dans le processeur, des fonctionnalités récentes du matériel telles que Intel VT-x ou AMD-V. Contrairement à beaucoup d'autres solutions de virtualisation, vous pouvez donc utiliser VirtualBox même sur du vieux matériel où ces fonctionnalités ne sont pas présentes. Les détails techniques sont expliqués au chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201.
- **Suppléments invité : dossiers partagés, fenêtres transparentes (seamless), virtualisation 3D.** Les suppléments invité de VirtualBox sont des paquets logiciels qu'on peut installer à l'intérieur des systèmes invités supportés pour améliorer leurs performances et effectuer une intégration et une communication accrues avec le système hôte. Après avoir installé les suppléments invité, une machine virtuelle supportera l'ajustement automatique des résolutions graphiques, les fenêtres transparentes, la vidéo 3D accélérée et davantage. Les suppléments invité sont décrits en détails au chapitre 4, *Les suppléments invité*, page 64.
Notamment, les suppléments invité offrent les "dossiers partagés", ce qui vous permet d'accéder aux fichiers depuis l'intérieur d'une machine invitée. Les dossiers partagés sont décrits au chapitre 4.3, *Dossiers partagés*, page 75.
- **Bon support matériel.** Entre autres, VirtualBox supporte :
 - **Le multiprocesseur invité (SMP).** VirtualBox peut montrer jusqu'à 32 processeurs virtuels à chaque machine virtuelle, indépendamment du nombre de coeurs présents sur votre hôte physiquement.
 - **Support des périphériques USB.** VirtualBox implémente un contrôleur USB virtuel et vous permet de connecter des périphériques USB de votre choix à vos machines virtuelles sans devoir installer de pilotes spécifiques sur l'hôte. Le support USB n'est pas limité à certaines catégories de périphériques. Pour des détails, voir chapitre 3.10.1, *Paramètres USB*, page 60.
 - **compatibilité matérielle.** VirtualBox virtualise une vaste gamme de périphériques virtuels, parmi lesquels beaucoup sont en général fournis par d'autres plateformes de virtualisation. Ceci inclut les contrôleurs de disques IDE, SCSI et SATA, plusieurs cartes son et réseau virtuelles, les ports série et parallèle virtuels, ainsi qu'un contrôleur d'interruptions entrée/sortie programmable avancé (I/O APIC), ce qu'on trouve dans de nombreux systèmes PC modernes. Cela facilite le clonage d'images de PC depuis des machines réelles et l'importation de machines virtuelles tierces dans VirtualBox.
 - **Support complet de l'ACPI.** L'Advanced Configuration and Power Interface (ACPI) est complètement supportée par VirtualBox. Cela facilite le clonage d'images de PC depuis des machines réelles et l'importation de machines virtuelles tierces dans VirtualBox. Avec son **support d'état d'énergie de l'ACPI** unique, VirtualBox peut même signaler aux systèmes d'exploitation supportant l'ACPI l'état de l'alimentation de l'hôte. Pour les systèmes mobiles sur batterie, le système invité peut ainsi économiser d'énergie et signaler à l'utilisateur le temps restant (par exemple en mode plein écran).
 - **Résolutions sur plusieurs écrans.** Les machines virtuelles VirtualBox supportent les résolutions d'écran sur autant d'écrans que d'écrans physiques, ce qui leur permet de s'étaler sur une grande variété d'écrans reliés au système hôte.
 - **Support iSCSI construit en dur.** Cette fonctionnalité unique vous permet de connecter une machine virtuelle directement à un serveur de stockage iSCSI sans passer par le système hôte. La VM accède à la cible iSCSI directement, sans l'intermédiaire requis pour virtualiser des disques virtuels dans les fichiers conteneurs. Pour des détails, voir le chapitre 5.10, *Serveurs iSCSI*, page 97.

- **Amorçage par le réseau PXE.** Les cartes réseaux virtuelles intégrées de VirtualBox supportent complètement l'amorçage à distance via Preboot Execution Environment (PXE).
 - **Multigénération d'instantanés en branches.** VirtualBox peut sauvegarder des instantanés de votre choix de l'état de la machine virtuelle. Vous pouvez revenir dans le passé and rétablir l'état de la machine virtuelle à n'importe quel instantané, puis démarrer une autre configuration de VM à partir de là, ce qui crée de fait une arborescence d'instantanés complète. Pour les détails, voir le chapitre 1.10, *Instantanés*, page 28. Vous pouvez créer et effacer des instantanés alors que la machine virtuelle est en fonction.
 - **Groupes de VMs.** VirtualBox offre une fonction de groupes permettant à l'utilisateur d'organiser les machines virtuelles en groupe ou individuellement. Outre les groupes classiques, il est également possible pour une VM d'être dans plus d'un groupe et pour des groupes à nested dans une hiérarchie - c'est-à-dire des groupes de groupes. Les opérations faisables sur les groupes sont généralement les mêmes que celles applicables aux VMs individuelles, à savoir le démarrage, la pause, la réinitialisation, la fermeture (état sauvegardé, envoyer une extinction, couper), désactiver l'état sauvegardé, afficher dans le système de fichiers, trier.
 - **Architecture propre ; une modularité sans précédent.** VirtualBox a un aspect extrêmement modulaire avec des interfaces de programmation internes bien définies and une séparation propre du code client and serveur. Cela facilite son contrôle par plusieurs interfaces à la fois : par exemple, vous pouvez démarrer une VM en cliquant simplement sur un bouton de l'interface graphique de VirtualBox, puis contrôler cette machine à partir de la ligne de commande, voire même à distance. Voir le chapitre 1.16, *Interfaces alternatives*, page 35 pour les détails.
- Du fait de son architecture modulaire, VirtualBox peut également présenter toutes ses fonctionnalités and sa flexibilité de configuration dans un **kit de développement logiciel (SDK)**, cohérent qui permet d'intégrer tous les aspects de VirtualBox à d'autres logiciels. Merci de voir le chapitre 11, *Interfaces de programmation de VirtualBox*, page 207 pour les détails.
- **Affichage de machines à distance.** La VirtualBox Remote Desktop Extension (VRDE) permet un accès distant et en haute performance à une machine virtuelle en fonction. Cette extension supporte le protocole de bureau distant (VRDP), construit au départ dans Microsoft Windows, avec des suppléments spéciaux pour un support complet de l'USB sur le client.
- Le VRDE de se base pas sur le serveur RDP construit dans Microsoft Windows ; mais il est inclu directement dans la couche de virtualisation. Il s'en suit qu'il fonctionne avec les systèmes d'exploitation hôte différents de Windows (même en mode texte) and il n'a pas besoin non plus d'un support applicatif dans la machine virtuelle. Le VRDE est décrit en détail au chapitre 7.1, *Affichage distant (VRDP support)*, page 109.
- Sur la base de cette possibilité spéciale, VirtualBox vous offre plus de fonctionnalités uniques :
- **Authentification RDP Extensible.** VirtualBox supporte déjà Winlogon sur Windows and PAM sur Linux pour l'authentification RDP. En outre, il inclut un SDK facile à utiliser qui vous permet de créer des interfaces de votre choix pour d'autres méthodes d'authentification ; voir chapitre 7.1.5, *Authentification RDP*, page 113 pour des détails.
 - **USB via RDP.** Via le support RDP de canaux virtuels, VirtualBox vous permet aussi de connecter des périphériques USB de votre choix en local sur une machine virtuelle en fonction à distance sur un serveur RDP VirtualBox ; voir le chapitre 7.1.4, *USB distant*, page 113 pour les détails.

1.4 Systèmes d'exploitation hôtes supportés

Actuellement, VirtualBox fonctionne sur les systèmes d'exploitation hôtes suivants :

- Hôtes **Windows** :
 - Windows XP, tous les packs service (32 bits)
 - Windows Server 2003 (32 bits)
 - Windows Vista (32 bits and 64 bits¹).
 - Windows Server 2008 (32 bits and 64 bits)
 - Windows 7 (32 bits and 64 bits)
 - Windows 8 (32 bits and 64 bits)
 - Windows Server 2012 (64 bits)
- Hôtes **Mac OS X** :²
 - 10.6 (Snow Leopard, 32 bits and 64 bits)
 - 10.7 (Lion, 32 bits et 64 bits)
 - 10.8 (Mountain Lion, 64 bits)

Il faut du matériel Intel ; merci de voir aussi le chapitre 14, *Limites connues*, page 231.

- Hôtes **Linux** (32 bits et 64 bits³). Cela inclut entre autres :
 - 8.04 (“Hardy Heron”), 8.10 (“Intrepid Ibex”), 9.04 (“Jaunty Jackalope”), 9.10 (“Karmic Koala”), 10.04 (“Lucid Lynx”), 10.10 (“Maverick Meerkat”), 11.04 (“Natty Narwhal”), 11.10 (“Oneiric Ocelot”), 12.04 (“Precise Pangolin”)
 - Debian GNU/Linux 5.0 (“lenny”) et 6.0 (“squeeze”)
 - Oracle Enterprise Linux 4 et 5, Oracle Linux 6
 - Redhat Enterprise Linux 4, 5 et 6
 - Fedora Core 4 à 17
 - Gentoo Linux
 - openSUSE 11.0, 11.1, 11.2, 11.3, 11.4, 12.1, 12.2
 - Mandriva 2010 et 2011

On devrait pouvoir utiliser VirtualBox sur la plupart des systèmes basés sur un noyau Linux 2.6, soit en utilisant l'installateur de VirtualBox, soit en faisant une installation manuelle ; voir le chapitre 2.3, *Installing sur des hôtes Linux*, page 39. Cependant, les distributions Linux formellement testées et supportées sont celles pour lesquelles nous offrons un paquet dédié.

Remarquez qu'à partir de VirtualBox 2.1, les systèmes d'exploitation hôtes Linux basés sur le noyau 2.4 ne sont plus supportées.

- Les hôtes **Solaris** (32 bits et 64 bits) sont supportés avec les restrictions listées au chapitre 14, *Limites connues*, page 231 :
 - Solaris 11 y compris Solaris 11 Express
 - Solaris 10 (u8 et supérieur)

¹Le support pour Windows 64 bits a été ajouté avec VirtualBox 1.5.

²Le support préliminaire de Mac OS X (étape beta) a été ajouté avec VirtualBox 1.4, le support complet avec 1.6. Le support pour Mac OS X 10.4 (Tiger) a été supprimé avec VirtualBox 3.1.

³Le support pour Linux 64 bits a été ajouté avec VirtualBox 1.4.

Remarquez que la liste ci-dessus est informelle. Le support Oracle pour les clients ayant un contrat d'assistance est limité aux systèmes d'exploitation hôtes précités. En outre, les fonctionnalités marquées comme **experimental** ne sont pas supportées. Les retours et les suggestions sur ces fonctionnalités sont bienvenues.

1.5 Installer VirtualBox et les packs d'extension

VirtualBox est fourni dans de nombreux paquets et son installation dépend de votre système d'exploitation hôte. Si vous avez déjà installé ce logiciel, l'installation devrait être facile : pour chaque plateforme hôte, VirtualBox utilise la méthode d'installation la plus courante facile possible. Si vous avez un problème ou des besoins particuliers, reportez vous à la chapitre [2, Détails sur l'installation](#), page 37 pour des détails sur les différentes méthodes d'installation.

À partir de la version 4.0, VirtualBox est divisé en plusieurs composants.

1. Le paquet de base contient tous les composants libres et il est sous licence GNU General Public License V2.
2. Des packs d'extension supplémentaires peuvent être téléchargés, qui rajoutent des fonctionnalités au paquet de base de VirtualBox. Pour l'instant, Oracle fournit un seul pack d'extension, qu'on peut trouver sur <http://www.virtualbox.org> et qui offre les fonctionnalités supplémentaires suivantes :
 - a) Péphériques virtuels USB 2.0 (EHCI) ; voir le chapitre [3.10.1, Paramètres USB](#), page 60.
 - b) Support de VirtualBox Remote Desktop Protocol (VRDP) ; voir le chapitre [7.1, Affichage distant \(VRDP support\)](#), page 109.
 - c) Amorçage Intel PXE ROM avec support pour la carte réseau E1000.
 - d) Support expérimental pour passthrough PCI sur les hôtes Linux ; voir le chapitre [9.6, PCI passthrough](#), page 171.

Les packs d'extension VirtualBox ont une extension de nom de fichiers .vbox-extpack. Pour installer une extension, double-cliquez simplement sur le fichier du paquet et une fenêtre de gestion des opérations réseaux apparaîtra, vous guidant à travers les étapes nécessaires.

Pour voir les packs d'extension actuellement installés, merci de démarrer le gestionnaire VirtualBox (voir la prochaine section). Dans le menu “Fichier”, merci de sélectionner “Préférences”. Dans la fenêtre qui apparaît, allez à la catégorie “Extensions” qui affiche les extensions actuellement installées et vous permet de supprimer un paquet ou d'en ajouter un.

Vous pouvez aussi utiliser VBoxManage en ligne de commande : voir le chapitre [8.36, VBoxManage extpack](#), page 159 pour des détails.

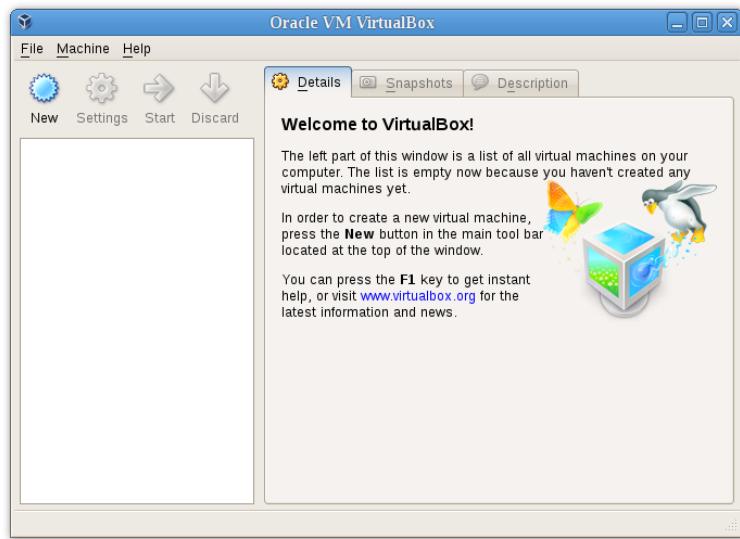
Note: Quand vous serez familier de l'utilisation des assistants, songez à utiliser le mode expert disponible dans certains assistants. Quand il est disponible, on peut le sélectionner en cliquant sur un bouton et cela accélère les processus de l'utilisateur pour utiliser l'assistant.

1.6 Démarrer VirtualBox

Après l'installation, vous pouvez démarrer VirtualBox comme suit :

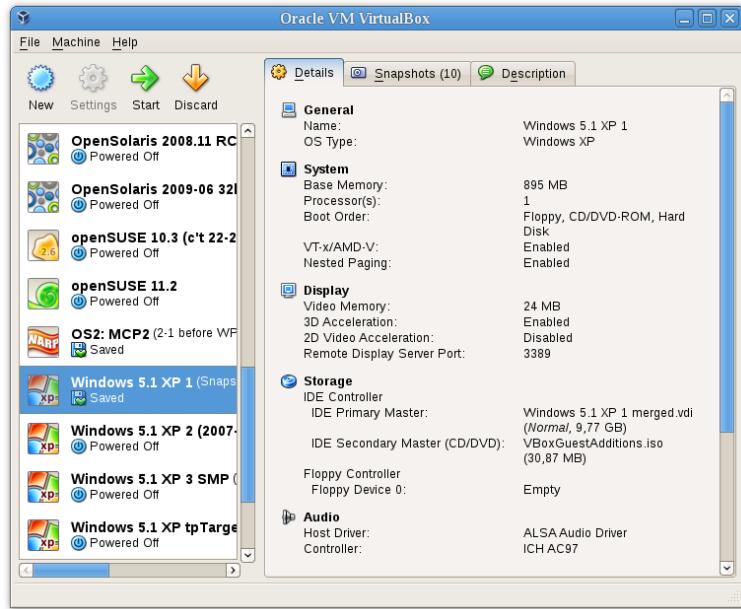
- Sur un hôte Windows, dans le menu standard “Programmes”, cliquez sur l’élément du groupe “VirtualBox”. Sur Vista ou Windows 7, vous pouvez aussi taper “VirtualBox” dans la zone de recherche du menu “Démarrer”.
- Sur un hôte Mac OS X, dans la zone de recherche, cliquez deux fois sur l’élément “VirtualBox” du dossier “Applications”. (Il se peut que vous vouliez mettre cet élément sur votre Dock.)
- Sur un hôte Linux ou Solaris, selon votre environnement de bureau, une icône “VirtualBox” peut avoir été mis soit dans le groupe “Système” soit dans “Outils système” de votre menu “Applications”. Sinon vous pouvez taper VirtualBox dans un terminal.

Quand vous démarrez VirtualBox pour la première fois, une fenêtre comme celle suivante devrait apparaître :



Cette fenêtre s'appelle le “**gestionnaire VirtualBox**”.. À gauche, vous pouvez voir un panneau qui listera, plus tard, toutes vos machines virtuelles. Comme vous n'en avez pas créées, la liste est vide. Une ligne de boutons au-dessus vous permet de créer de nouvelles VMs et de travailler sur celles existantes, lorsque vous en avez. Le panneau à droite affiche les propriétés de la machine virtuelle actuellement sélectionnée s'il y en a une. De nouveau, comme vous n'avez pas encore de machine, le panneau affiche un message de bienvenue.

Pour vous donner une idée de ce à quoi pourrait ressembler VirtualBox plus tard, après que vous avez créé des machines, voici un autre exemple :



1.7 Créer votre première machine virtuelle

Cliquez sur le bouton “Nouveau” en haut de la fenêtre du gestionnaire VirtualBox. Un assistant apparaîtra pour vous guider à travers le paramétrage d'une nouvelle machine virtuelle (VM) :



Sur les écrans suivants, l'assistant vous demandera le minimum d'informations dont il a besoin pour créer une VM, en particulier :

1. Le **nom de la VM** sera plus tard affiché dans la liste des VMs de la fenêtre du gestionnaire VirtualBox, et il sera utilisé pour les fichiers de la VM sur le disque. Si vous pouvez utiliser n'importe quel nom, gardez en tête qu'après avoir créé quelques VMs, vous apprécierez d'avoir donné à vos VMs des noms parlants ; ainsi “Ma VM” serait moins utile que “Windows XP SP2 avec OpenOffice”.
2. Pour le **“Type de système d’exploitation”**, sélectionnez le système d’exploitation que vous voudrez installer plus tard. Les systèmes d’exploitation supportés sont ici regroupés ; si

vous voulez installer une chose très rare et non listé, sélectionnez “Autre”. Selon votre sélection, VirtualBox activera ou désactivera certains paramètres de VM dont peut avoir besoin votre système d’exploitation invité. C’est particulièrement important pour les invités 64 bits (voir le chapitre [3.1.2, Invités 64 bits](#), page [49](#)). Il est donc recommandé de toujours paramétriser la bonne valeur.

3. Sur la page suivante, sélectionnez la **mémoire (RAM)** que VirtualBox devra affecter à chaque fois que la machine virtuelle sera démarrée. La quantité de mémoire donnée ici sera retirée de votre machine hôte laissée au système d’exploitation invité, lequel verra cette mémoire comme RAM installée sur l’ordinateur (virtuel).

Note: Choisissez avec prudence ce paramètre ! La mémoire donnée à la VM ne sera pas disponible pour votre OS hôte pendant que la VM sera en fonction, donc ne spécifiez pas plus que vous ne pouvez donner. Par exemple, si votre machine hôte a 1 Go de RAM et si vous entrez 512 Mo de RAM pour une machine virtuelle en particulier, pendant l’exécution de cette VM, vous n’auriez plus que 512 Mo pour tous les autres logiciels de votre hôte. Si vous lancez deux VMs en même temps, encore plus de mémoire sera affectée à la deuxième VM (qui peut même ne pas démarrer si la mémoire n’est pas disponible). D’un autre côté, vous devriez spécifier autant que votre système d’exploitation invité (et vos applications) ont besoin pour s’exécuter correctement.

Un invité Windows XP exigera au moins quelques centaines de Mo de RAM pour fonctionner correctement, et Windows Vista refusera même de s’installer avec moins de 512 Mo. Bien sûr, si vous voulez lancer des applications gourmandes en ressources graphiques dans votre VM, vous pouvez avoir besoin de plus de RAM.

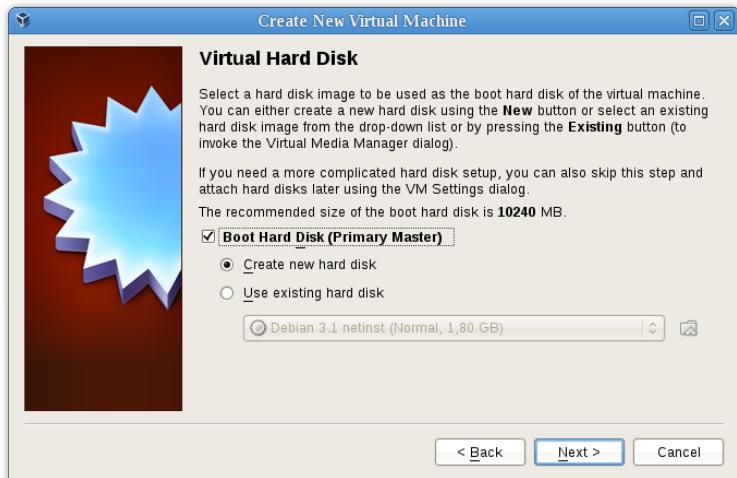
La règle d’or est donc que si vous avez 1 Go de RAM voire plus dans votre ordinateur hôte, il est sûr d’affecter 512 Mo à chaque VM. Mais dans tous les cas, assurez-vous d’avoir au moins 256 à 512 Mo de RAM sur votre système d’exploitation hôte. Sinon, il se peut que vous ne votre OS hôte fasse un usage excessif de l’espace d’échange sur votre disque dur, ce qui peut conduire à un plantage de votre système hôte.

Comme avec les autres paramètres, vous pourrez modifier ce réglage plus tard après avoir créé la VM.

4. Ensuite vous devez spécifier un **disque dur virtuel** pour votre VM.

Il existe de nombreuses façons, potentiellement compliquées, de fournir de l’espace de disque dur à une VM avec VirtualBox (voir le chapitre [5, Le stockage virtuel](#), page [85](#) pour les détails), mais la manière la plus courante est d’utiliser un gros fichier image sur votre “vrai” disque dur, dont VirtualBox présente le contenu à votre VM comme si c’était un disque dur normal. Ce fichier représente un disque dur complet donc vous pouvez même le copier vers un autre hôte et l’utiliser avec une autre installation VirtualBox.

L’assistant affiche la fenêtre suivante :



Vous avez alors les choix suivants :

- Pour créer un nouveau disque dur virtuel vierge, appuyez sur le bouton “Nouveau”.
- Vous pouvez utiliser un fichier image de disque **existant**.

La **boîte à liste déroulante** affichée dans la fenêtre contient toutes les images de disque dont se souvient VirtualBox, probablement parce qu’elles sont actuellement attachées à une machine virtuelle (ou elles l’ont jadis été).

Vous pouvez aussi cliquer sur le petit **bouton dossier** à côté de la boîte à liste déroulante pour ouvrir une boîte de dialogue de fichier standard qui vous permet de choisir un fichier image de disque sur le disque de votre hôte.

x

Vraisemblablement, si vous utilisez VirtualBox pour la première fois, vous voudrez créer une nouvelle image de disque. Appuyez donc sur le bouton “Nouveau”.

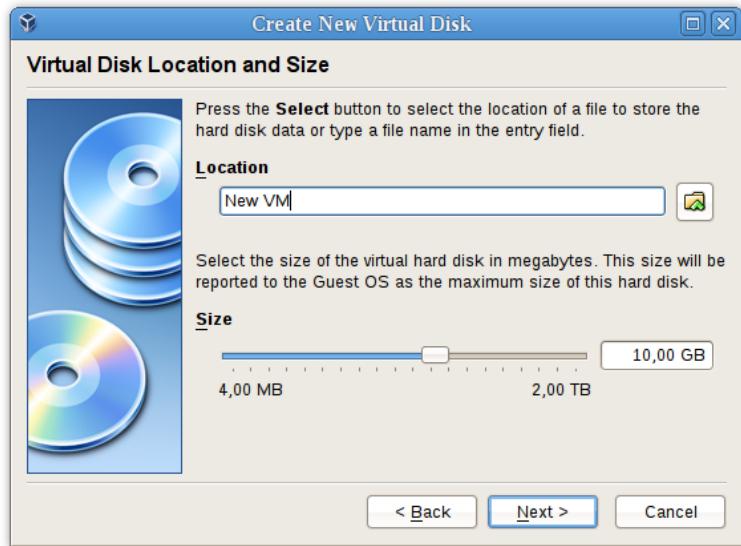
Ceci ouvre une nouvelle fenêtre, celle “**Assistant Crée un nouveau disque virtuel**”, qui vous aide à créer un nouveau fichier image de disque dans le dossier de la nouvelle machine virtuelle.

VirtualBox supporte deux types de fichiers image :

- Un **fichier rempli dynamiquement** ne grossira que lorsque l’invité stockera des données sur son disque dur virtuel. Il sera donc petit au départ sur le disque dur hôte et ne grossira que plus tard jusqu’à la taille indiquée, au fur et à mesure que des données le rempliront.
- Un **fichier à taille fixe** occupera immédiatement l’espace spécifié même si seule une partie de l’espace disque virtuel est en réalité utilisée. S’il occupera beaucoup plus de place, un fichier à taille fixe connaît moins de latence et il va donc légèrement plus vite qu’un fichier rempli dynamiquement.

Pour des détails sur les différences, merci de vous reporter au chapitre [5.2, Fichiers images de disque \(VDI, VMDK, VHD, HDD\)](#), page 88.

Pour empêcher votre disque dur physique de tourner à plein, VirtualBox limite la taille du fichier image. Là encore il doit y avoir assez de place pour le contenu de votre système d’exploitation et les applications que vous voulez installer - pour un invité Windows ou Linux modernes, vous aurez probablement besoin de plusieurs gigaoctets pour une utilisation sérieuse :



Après avoir sélectionné ou créé votre fichier image, appuyez de nouveau sur “**Suivant**” pour passer à la page suivante.

5. Après avoir cliqué sur “**Terminer**”, votre nouvelle machine virtuelle sera créée. Vous la verrez alors dans la liste à gauche de la fenêtre du gestionnaire, avec le nom que vous avez entré au départ.

1.8 Lancer votre machine virtuelle

Pour démarrer une machine virtuelle, vous avez plusieurs choix :

- Cliquez deux fois sur son entrée dans la liste dans a fenêtre du gestionnaire ou
- sélectionnez son entrée dans la liste de la fenêtre du gestionnaire et appuyez sur le bouton “Démarrer” en bas ou
- pour les machines virtuelles créées avec VirtualBox 4.0 ou supérieur, allez au dossier “VirtualBox VMs” dans le dossier personnel de votre système, cherchez le sous-répertoire de la machine que vous voulez démarrer et cliquez deux fois sur le fichier de paramètres de la machine (ayant une extension de fichier .vbox).

Ceci ouvre une nouvelle fenêtre et la machine virtuelle que vous avez sélectionnée va démarrer. Tout ce que vous verriez sur l'écran d'un système normal apparaît dans la fenêtre, comme vous pouvez le voir sur l'image du chapitre [1.2, Un peu de terminologie](#), page [12](#).

En général, vous pouvez utiliser la machine virtuelle presqu'exactement comme vous utiliseriez un vrai ordinateur. Deux points gagnent à être mentionnés quand même.

1.8.1 Démarrage d'une nouvelle VM pour la première fois

Quand on démarre une VM pour la première fois, un autre assistant “**Assistant premier démarrage**” – apparaîtra pour vous aider à sélectionner un **média d'installation**. Comme la VM est créée vierge, agissant exactement comme un vrai ordinateur sans système d'exploitation installé, elle ne fera rien et affichera un message d'erreur selon lequel aucun système d'exploitation n'a été trouvé.

C'est pourquoi l'assistant vous aide à sélectionner un média à partir duquel vous pouvez installer un système d'exploitation.

- Si vous avez un CD ou un DVD physique à partir duquel vous voulez installer votre système d'exploitation invité (par exemple un CD ou un DVD d'installation de Windows), mettez-le dans le lecteur CD ou DVD de votre hôte.

Puis dans la liste déroulante des médias d'installation de l'assistant, sélectionnez “**Lecteur hôte**” avec la bonne lettre de lecteur (ou, en cas d'hôte Linux, le fichier de périphérique). Ceci permettra à votre VM d'accéder au média dans votre lecteur hôte et vous pouvez poursuivre l'installation à partir de là.

- Si vous avez téléchargé un média d'installation sur Internet sous forme d'un fichier image ISO (en général pour une distribution Linux), vous pourriez graver ce fichier sur un CD ou un DVD vierge et poursuivre comme on vient de décrire. Mais avec VirtualBox, vous pouvez passer cette étape et monter le fichier ISO directement. VirtualBox présentera ce fichier comme un lecteur de CD ou de DVD-ROM à la machine virtuelle, comme il le fait avec une image de disque dur virtuel.

Dans ce cas, la liste déroulante de l'assistant contient la liste des médias d'installation qui ont été précédemment utilisés avec VirtualBox.

Si votre média n'est pas dans la liste (surtout si vous utilisez VirtualBox pour la première fois), sélectionnez la petite icône de dossier à côté de la liste déroulante pour afficher une boîte de dialogue de fichiers standard dans laquelle vous pouvez choisir le fichier image sur vos disques hôtes.

Dans les deux cas, après avoir fait vos choix dans l'assistant, vous pourrez installer votre système d'exploitation.

1.8.2 Capturer et rendre le clavier et la souris

À partir de la version 3.2, VirtualBox offre aux nouvelles machines virtuelles une tablette USB virtuelle à travers laquelle les événements de la souris sont transmis au système d'exploitation invité. Il s'en suit que si vous lancez un système d'exploitation invité moderne qui peut gérer de tels périphériques, il se peut que le support de la souris fonctionne sans que la souris ne soit “capturée” comme décrit ci-dessous ; voir chapitre [3.4.1, Onglet “Carte mère”](#), page [52](#) pour plus d'informations.

Sinon, si la machine virtuelle ne voit que la souris et le clavier PS/2 standards, car le système d'exploitation de la machine virtuelle ne “sait” pas qu'il ne fonctionne pas sur un vrai ordinateur, il attend à avoir un contrôle exclusif de votre clavier et de votre souris. Ce n'est cependant pas le cas puisque, sauf si vous lancez votre VM en mode plein écran, votre VM doit partager le clavier et la souris avec d'autres applications et éventuellement d'autres VMs sur l'hôte.

Il s'en suit qu'immédiatement après l'installation d'un système d'exploitation invité, et avant que vous installiez les Suppléments invité (nous expliquerons cela dans une minute), seule une des deux – votre VM ou le reste de votre ordinateur – peut “déténir” le clavier et la souris. Vous verrez un *deuxième* pointeur de souris qui sera toujours enfermé dans la fenêtre de la VM. De base, vous activez la VM en cliquant dedans.

Pour rendre le clavier et la souris à votre système d'exploitation hôte, VirtualBox réserve une touche spéciale de votre clavier pour lui : la “**touche hôte**”. Par défaut, c'est la touche *Contrôle droit* de votre clavier ; sur un hôte Mac, la touche hôte par défaut est la touche de commande gauche. Vous pouvez modifier ce réglage par défaut dans les paramètres globaux de VirtualBox, voir chapitre [1.15, Paramètres globaux](#), page [35](#). Dans tous les cas, le paramétrage actuel de la touche hôte est toujours affiché *en bas à droite de la fenêtre de la VM*, au cas où vous l'auriez oublié :



En détail, tout se traduit comme suit :

- Votre **clavier** appartient à la VM si la fenêtre de la VM du bureau hôte contient le focus du clavier (et si vous avez beaucoup de fenêtres ouvertes dans votre système d'exploitation invité, la fenêtre contenant le focus dans votre VM). Cela veut dire que si vous voulez taper quelque chose dans votre VM, cliquez d'abord sur la barre de titre de la fenêtre de votre VM.

Pour rendre la propriété du clavier, appuyez sur la touche hôte (comme expliqué ci-dessus, en général la touche Contrôle droit).

Remarquez que si la VM possède le clavier, certaines séquences de touches (comme Alt-Tab par exemple) ne seront plus vues par l'hôte, mais iront plutôt à l'invité. Après avoir appuyé sur la touche hôte pour réactiver le clavier hôte, tous les appuis iront de nouveau à l'hôte pour que les séquences comme Alt-Tab ne touchent plus l'invité.

- Votre **souris** n'appartient à la VM qu'après avoir cliqué de nouveau dans la fenêtre de la VM. Le pointeur de souris de l'hôte disparaîtra et votre souris maniera le pointeur de l'invité au lieu de votre pointeur de souris normal.

Remarquez que l'appartenance de la souris ne dépend pas de celle du clavier, même après avoir cliqué sur la barre de titre pour pouvoir taper dans la fenêtre de la VM, votre souris n'appartient pas nécessairement à la VM.

Pour rendre l'appartenance de votre souris à la VM, appuyez sur la touche hôte.

Comme ce comportement peut être gênant, VirtualBox offre un ensemble d'outils et de pilotes de périphériques pour les systèmes invités, appelé les "Suppléments invité VirtualBox", qui rendent beaucoup transparentes les opérations du clavier et de la souris de la VM. Très important, les suppléments vous débarrasserons du deuxième pointeur de souris de "l'invité" et fera fonctionner la souris de votre hôte directement dans l'invité.

On décrira ceci plus tard au chapitre 4, *Les suppléments invité*, page 64.

1.8.3 Taper des caractères spéciaux

Les systèmes d'exploitation prévoient certaines combinaisons de touches pour lancer certaines procédures. Certaines de ces combinaisons de touches peuvent être difficiles à faire dans une machine virtuelle ; car il y a trois candidats à la réception de l'entrée clavier : le système d'exploitation hôte, VirtualBox ou le système d'exploitation invité. Le récepteur de l'appui sur les touches dépend d'un certain nombre de facteurs comprenant la touche elle-même.

- Les systèmes d'exploitation hôte réservent certaines combinaisons de touches pour eux. C'est par exemple impossible de faire **Ctrl+Alt+Supp** si vous voulez redémarrer le système d'exploitation invité de votre machine virtuelle, car cette combinaison de touches est en général rattachée en dur à l'OS hôte (Windows comme Linux l'interceptent), et l'appui sur cette combinaison de touches redémarrera donc votre *hôte*.

De plus, sur les systèmes on Linux et Solaris, qui utilise le système X Window, la combinaison de touches **Ctrl+Alt+Effacement** redémarrent en général le serveur X (pour relancer toute interface graphique en cas de problème). Comme le serveur X intercepte cette combinaison, l'appui sur ces touches relancera en général votre interface graphique *hôte* (et tuera tous les programmes, dont VirtualBox, dans le processus).

Par ailleurs, sur les hôtes Linux qui supportent les terminaux virtuels, la combinaison de touches **Ctrl+Alt+Fx** (où Fx est une touche de fonctions entre F1 et F12) permet en principe de basculer entre des terminaux virtuels. Comme avec Ctrl+Alt+Supp, ces combinaisons sont interceptées par le système d'exploitation hôte et elles basculent donc toujours entre les terminaux de *l'hôte*.

Si vous voulez plutôt envoyer ces combinaisons de touches au système d'exploitation *invité* de la machine virtuelle, vous devrez utiliser une des méthodes suivantes :

- Utiliser les icônes du menu “Machine” de la fenêtre de la machine virtuelle. Vous y trouverez “Insérer Ctrl+Alt+Supp” et “Ctrl+Alt+Effacement” ; cette dernière n'ayant toutefois d'effet qu'avec les invités Linux ou Solaris.
- Appuyer sur des combinaisons de touches spéciales avec la touche hôte (en principe, la touche contrôle droit), alors VirtualBox traduira, pour la machine virtuelle :
 - * **Touche hôte + Supp** envoie Ctrl+Alt+Supp pour redémarrer l'invité);
 - * **Touch hôte + Effacement** pour envoyer Ctrl+Alt+Effacement (pour redémarrer l'interface graphique d'un invité Linux ou Solaris) ;
 - * **Touche hôte + F1** (ou autres touches de fonction) pour simuler Ctrl+Alt+F1 (ou d'autres touches de fonction, pour basculer entre les terminaux virtuels d'un invité Linux).
- Pour d'autres combinaisons de touches telles que **Alt-Tab** (pour basculer entre des fenêtres ouvertes), VirtualBox vous permet de configurer si ces combinaisons concerneront vhôte ou l'invité, si la machine virtuelle a le focus. C'est un paramètre global de toutes les machines virtuelles qui se trouve dans “Fichier” -> “Préférences” -> “Entrée” -> “Capture automatique du clavier”.

1.8.4 Changer de média amovible

Pendant qu'une machine virtuelle fonctionne, vous pouvez changer de média amovible dans le menu “Périphériques” de la fenêtre de la VM. Vous pouvez y sélectionner en détail ce que VirtualBox présente à votre VM en tant que CD, DVD, ou 4isquette.

Les paramètres sont les mêmes que ceux disponibles pour la VM dans la boîte de dialogue “Paramètres” de la fenêtre principale de VirtualBox, mais vu que la boîte de dialogue est désactivée quand la machine est dans l'état “en fonction” ou “sauvegardée”, ce menu supplémentaire vous évite d'éteindre et de redémarrer la VM à chaque fois que vous voulez changer de média.

Dès lors, dans le menu “Périphériques”, VirtualBox vous permet d'attacher le lecteur hôte à l'invité ou de sélectionner une image de DVD ou de disquette en utilisant le gestionnaire d'images de disque, tout comme décrit à la chapitre [1.11, Configuration d'une machine virtuelle](#), page [31](#).

1.8.5 Redimensionner la fenêtre de la machine

Vous pouvez redimensionner la fenêtre de la machine virtuelle quand elle fonctionne. Dans un tel cas, une des choses suivantes arrivera :

1. Si vous avez activé le “**mode échelle**”, l'écran de la machine virtuelle sera adapté à la taille de la fenêtre. Ceci peut être utile si vous avez de nombreuses machines en fonction et si vous voulez les visualiser en fonction en arrière-plan. Sinon, il pourrait être utile d'agrandir

la fenêtre si l'écran d'affichage de la fenêtre est très petit, par exemple car vous y exécutez un vieux système d'exploitation.

Pour activer le mode échelle, appuyez sur **touche hôte + C**, ou sélectionnez “Mode échelle” du menu “Machine” dans la fenêtre de la VM. Pour désactiver le mode échelle, appuyez de nouveau sur la touche hôte + C.

Les valeurs d'affichage de l'écran de l'invité sont préservées lorsque vous redimensionnez la fenêtre. Pour ignorer ces valeurs d'affichage, appuyez sur Majuscule pendant l'opération de redimensionnement.

Merci de voir le chapitre 14, *Limites connues*, page 231 pour des remarques supplémentaires.

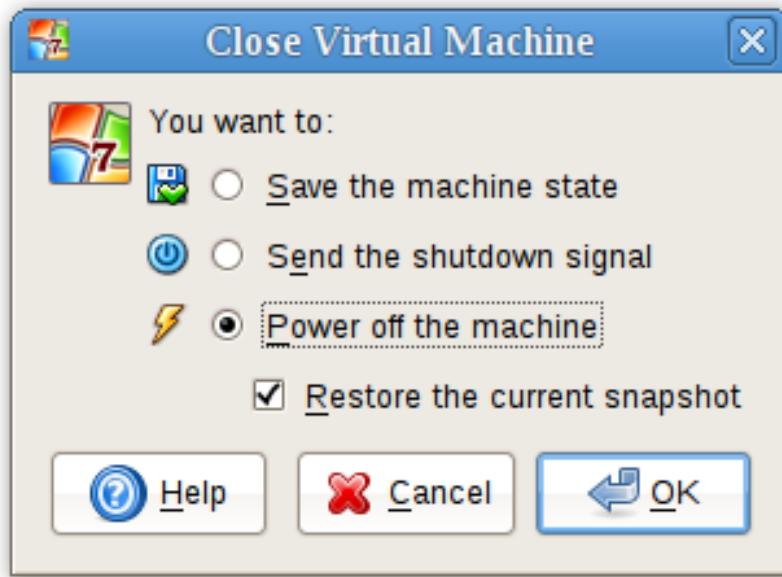
2. Si vous avez installé les Suppléments invité et s'ils supportent le **redimensionnement** automatique, les suppléments invité ajusteront automatiquement la résolution de l'écran du système d'exploitation invité. Par exemple, si vous exécutez un invité Windows 1rc une résolution de 1024x768 pixels et si vous élargissez la fenêtre de la VM de 100 pixels, les suppléments invité passeront la résolution de l'affichage de Windows en 1124x768.

Merci de voir le chapitre 4, *Les suppléments invité*, page 64 pour plus d'informations sur les suppléments invité.

3. Sinon, si la fenêtre est plus haute que l'écran de la VM, l'écran sera centré. Si elle est plus petite, les barres de défilement seront ajoutées à la fenêtre de la machine.

1.8.6 Sauvegarder l'état de la machine

Quand vous cliquez sur le bouton “Fermer” de la fenêtre de votre machine virtuelle (en haut à droite de la fenêtre, exactement comme vous fermeriez n'importe quelle fenêtre de votre système), VirtualBox vous demande si vous voulez “sauvegarder” ou “couper” la VM. (Vous pouvez appuyez sur la touche hôte et “Q” simultanément en guise de raccourci.)



La différence entre ces trois options est cruciale. Elles signifient :

- **Sauvegarder l'état de la machine** : Avec cette option, VirtualBox “gèle” la machine virtuelle en sauvegardant complètement son état sur votre disque local.

Quand vous redémarrez la VM plus tard, vous vous retrouverez avec la VM exactement là où vous l'avez interrompue. Tous vos programmes seront encore ouverts et votre ordinateur recommencera ses tâches. La sauvegarde de l'état d'une machine virtuelle revient, d'une certaine façon, à mettre en veille prolongée un ordinateur portable (par exemple en fermant l'écran).

- **Envoyer le signal Éteindre.** Ceci enverra un message d'extinction ACPI à la machine virtuelle, qui a le même effet que si vous aviez appuyé sur le bouton d'alimentation d'un ordinateur réel. Si la VM exécute un système d'exploitation moderne, cela devrait provoquer un vrai mécanisme d'extinction de la VM.
- **PCouper la machine** : Avec cette option, VirtualBox arrête aussi l'exécution de la machine virtuelle, mais *sans* sauvegarder son état.

Avertissement: Cela revient à débrancher le câble d'alimentation d'un vrai ordinateur sans l'éteindre correctement. Si vous redémarrez la machine après l'avoir coupée, votre système d'exploitation devra redémarrer complètement et il se peut qu'il fasse une vérification de ses disques systèmes (virtuels). Vous ne devriez donc pas le faire car cela peut conduire à une perte de données ou à un état incohérent du disque du système invité.

Par exception, si votre machine virtuelle a des instantanés (voir le prochain chapitre), vous pouvez utiliser cette option pour **restaurer l'instantané actuel** de la machine virtuelle rapidement. Dans ce cas, couper la machine ne dégradera pas son état mais seules les modifications faites depuis que vous avez pris l'instantané seront perdues.

Le bouton “**Désactiver**” de la fenêtre du gestionnaire de VirtualBox Manager désactive un état sauvegardé de la machine virtuelle. Cela a le même effet que de couper la machine, donc les mêmes avertissements s'appliquent.

1.9 Utiliser des groupes de VMs

Les groupes de VMs permettent à l'utilisateur de créer des groupes ad hoc de VMs, et de gérer et d'effectuer des fonctions de façon collective ou individuelle sur ceux-ci. Il existe un certain nombre de fonctions liées aux groupes :

1. Créer un groupe en utilisant l'option de l'interface graphique 1) Remontez une VM au-dessus d'une autre VM.

Créer un groupe en utilisant l'option 2) Sélectionnez plusieurs VMs et sélectionnez “Groupe” sur le menu issu du clic droit, comme suit :



2. Option en ligne de commande 1) Créer un groupe et y affecter une VM :

```
VBoxManage modifyvm "Fred" --groups "/TestGroup"
```

Option en ligne de commande 2) Retirer une VM d'un groupe et effacer le groupe s'il est vide :

```
VBoxManage modifyvm "Fred" --groups ""
```

3. Plusieurs groupes par exemple :

```
VBoxManage modifyvm "Fred" --groups "/TestGroup,/TestGroup2"
```

4. Groupes nested – hiérarchie de groupes comme :

```
VBoxManage modifyvm "Fred" --groups "/TestGroup/TestGroup2"
```

5. Résumé des commandes de groupes : Démarrer, mettre en pause, réinitialiser, fermer (sauvegarder l'état, envoyer un signal d'extinction, couper), désactiver l'état sauvegardé, afficher dans un système de fichiers, trier.

1.10 Instantanés

Avec les instantanés, vous pouvez sauvegarder un état particulier d'une machine virtuelle pour une utilisation ultérieure. Plus tard, vous pourrez rétablir son état même si vous pouvez avoir beaucoup modifié la VM entre temps. Un instantané de la machine virtuelle est ainsi équivalent à passer une machine en état "sauvegardé", comme décrit ci-dessus, mais il peut y en avoir beaucoup, et ces états sauvegardés sont préservés.

Vous pouvez voir les instantanés d'une machine virtuelle en sélectionnant d'abord une machine dans le gestionnaire VirtualBox, puis en cliquant sur le bouton "Instantanés" en haut à droite. Jusqu'à ce que vous preniez un instantané de la machine, la liste des instantanés restera vide sauf pour l'icône "état actuel" qui représente le point "Maintenant" du temps de vie de la machine virtuelle.

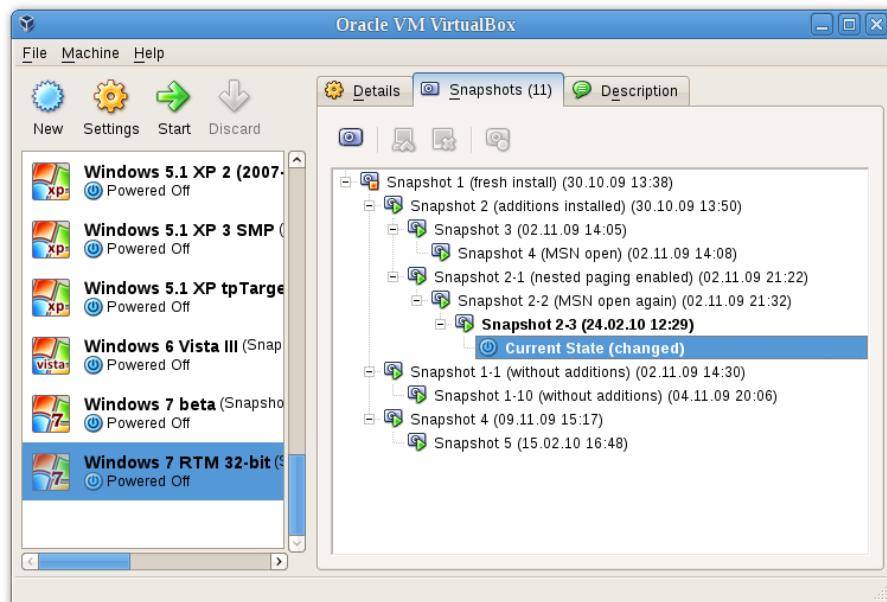
1.10.1 Prendre, restaurer et effacer des instantanés

Il existe trois opérations liées aux instantanés :

1. Vous pouvez **prendre un instantané**. Ceci effectue une copie de l'état actuel de la machine, auquel vous pourrez revenir plus tard n'importe quand..
 - Si votre VM est en fonction, sélectionnez “Prendre un instantané” dans le menu déroulant “Machine” de la fenêtre de la VM.
 - Si votre VM est dans l'état “sauvegardée” ou “coupée”, (comme indiqué à côté de la VM dans la fenêtre principale de VirtualBox), cliquez sur l'onglet “Instantanés” en haut à droite de la fenêtre principale, puis
 - soit sur la petite icône en appareil photo (pour “Prendre un instantané”), soit
 - effectuez un clic droit sur l'icône “état actuel” dans la liste et sélectionnez “Prendre un instantané” dans le menu.

Dans tous les cas, une fenêtre apparaîtra et vous demandera un nom d'instantané. Ce nom a une finalité purement de référence pour vous aider à vous souvenir de l'état de l'instantané. Par exemple, un nom utile serait “Installation neuve à partir de rien, pas de suppléments invité”, ou “Pack service à peine installé”. Vous pouvez aussi ajouter un texte plus long dans le champ “Description” si vous le voulez.

Votre nouvel instantané apparaîtra ensuite dans la liste des instantanés. Juste dessous votre nouvel instantané, vous verrez une icône appelée “état actuel”, signifiant que l'état actuel de votre VM est une variante basée sur l'instantané que vous avez pris précédemment. Si vous prenez plus tard un autre instantané, vous verrez qu'ils seront affichés en séquences et chaque instantané consécutif dérive d'un précédent :



VirtualBox n'impose aucune limite quant au nombre d'instantanés que vous pouvez prendre. La seule limite pratique est l'espace disque sur votre rôle : chaque instantané stocke l'état de 1 machine virtuelle et occupe donc de l'espace disque (voir la prochaine section pour des détails sur ce qui est stocké exactement dans un instantané.)

2. Vous pouvez **restaurer un instantané** en effectuant un clic droit sur un instantané que vous avez pris dans la liste des instantanés. Un restaurant un instantané, vous revenez en

arrière (ou vous reculez) dans le temps : l'état actuel de la machine est perdu et la machine est restaurée dans l'état exact où elle était quand vous avez pris l'instantané.⁴

Note: La restauration d'un instantané concernera également les disques durs connectés à votre VM puisque l'état complet des disques durs virtuels sera rétabli. Ceci signifie aussi que tous les fichiers qui ont été créés depuis l'instantané et toutes les autres modifications de fichiers *seront perdues*. Afin d'empêcher une telle perte de données, tout en utilisant la fonctionnalité des instantanés, il est possible d'ajouter un deuxième disque dur en mode "write-through" en utilisant l'interface VBoxManage et de l'utiliser pour stocker vos données. Vu que les disques durs write-through *ne sont pas* inclus dans les instantanés, ils restent inchangés quand une machine est rétablie. Voir le chapitre [5.4, Modes spéciaux d'écriture d'images](#), page 90 pour des détails.

Pour éviter de perdre l'état actuel quand vous restaurez un instantané, vous pouvez créer un nouvel instantané avant la restauration.

En restaurant un instantané précédent et en prenant plus d'instantanés à partir de là, il est même possible de créer une sorte de réalité alternative et de basculer entre les différentes histoires de la machine virtuelle. Cela peut donner une arborescence complète d'instantanés de machine comme le montre l'impression d'écran ci-dessus.

3. Vous pouvez aussi **effacer un instantané**, ce qui ne touchera pas l'état de la machine virtuelle mais seulement les fichiers du disque qu'utilisait VirtualBox pour stocker les données de l'instantané, libérant ainsi de l'espace disque. Pour effacer un instantané, effectuez un clic droit dessus dans l'arborescence des instantanés et sélectionnez "Effacer". À partir de VirtualBox 3.2, on peut même effacer des instantanés pendant qu'une machine fonctionne.

Note: Si la prise et la restauration d'instantanés sont des opérations très rapides, l'effacement d'un instantané peut mettre temps considérable car de grandes quantité de données peuvent être copiées entre plusieurs fichiers images de disques. Il se peut que les fichiers de disque temporaires aient besoin de beaucoup de place pendant le déroulement de l'opération.

Certaines situations ne peuvent pas être gérées pendant qu'une VM est en fonction et vous aurez un message circonstancié selon lequel vous devez effectuer cet effacement d'instantané quand la VM sera éteinte.

1.10.2 Contenu d'un instantané

Voyez un instantané comme un marquage dans le temps que vous avez posé. De façon plus formelle, un instantané consiste en trois éléments :

- Il contient une copie complète des paramètres de la VM y compris la configuration matérielle, afin que quand vous restaurerez un instantané, les paramètres de la VM soient également restaurés (par exemple, si vous avez modifié la configuration du disque dur ou les paramètres système de la VM, ce changement est annulé quand vous restaurez un instantané.)

La copie de ces paramètres est stockée dans la configuration de la machine, fichier texte XML, occupant ainsi très peu de place.

⁴La terminologie et la fonctionnalité de restauration d'instantanés ont toutes deux changé avec VirtualBox 3.1. Avant cette version, il n'était possible que de revenir au tout dernier instantané pris – pas à ceux antérieurs, et l'opération s'appelait "Désactiver l'état actuel" et non "Restaurer le dernier instantané". La limite a été dépassée avec la version 3.1. Il est maintenant possible de restaurer *n'importe quel* instantané, revenir en arrière et reculer dans le temps.

- L'état complet de tous les disques virtuels attachés à la machine est préservé. Le retour en arrière sur un instantané signifie que toutes les modifications faites sur les disques de la machine – fichier par fichier, bit par bit – seront toutes annulées. Les fichiers créés entre-temps disparaîtront, les fichiers effacés seront restaurés, les modifications de fichiers seront inversées.

(À proprement parler, ceci n'est vrai que pour les disques durs virtuels en mode “normal”. Comme indiqué ci-dessus, vous pouvez configurer des disques pour se comporter autrement avec les instantanés (voir chapitre 5.4, *Modes spéciaux d'écriture d'images*, page 90. De manière encore plus précise et juste techniquement, ce n'est pas le disque dur virtuel lui-même qui est restauré quand on restaure un instantané. En fait, quand on prend un instantané, VirtualBox crée des images de différenciation contenant seulement les modifications depuis que l'instantané a été pris, puis le instantané\$ est restauré, VirtualBox applique cette image de différenciation, revenant ainsi en arrière vers l'état précédent. Ceci est non seulement plus rapide, mais cela utilise moins d'espace disque. Pour les détails, qui peuvent être complexes, merci de voir le chapitre 5.5, *Images de différenciation*, page 92.)

La création d'une telle image de différenciation n'occupe pas beaucoup de place dès le départ sur le disque hôte, puisque l'image de différenciation sera vide au départ (et grandira plus tard de façon dynamique à chaque opération d'écriture sur le disque). Par contre, plus vous utiliserez la machine après avoir créé l'instantané, plus l'image de différenciation grossira.

- Enfin, si vous avez pris un instantané pendant que la machine était en fonction, l'état de la mémoire de la machine est également sauvegardé dans l'instantané (tout comme la mémoire peut être sauvegardée quand vous fermez la fenêtre de la VM). Quand vous restaurerez un tel instantané, l'exécution recommence exactement là où elle en était quand vous avez pris l'instantané.

Le fichier de l'état de la mémoire peut être aussi gros que la taille de la mémoire de la machine virtuelle et il occupera donc lui aussi beaucoup de place sur le disque.

1.11 Configuration d'une machine virtuelle

Quand vous sélectionnez une machine virtuelle dans la liste dans la fenêtre du gestionnaire, vous verrez à droite un résumé des paramètres de cette machine.

Un clic sur le bouton “Paramètres” dans la barre d’outils en haut ouvre une fenêtre détaillée où vous pouvez configurer de nombreuses propriétés de la VM sélectionnée. Mais attention, bien qu'il soit possible de modifier tous les paramètres de la VM après avoir installé un système d'exploitation invité, certaines modifications pourraient empêcher un système d'exploitation invité de fonctionner correctement après l'installation.

Note: Le bouton “Paramètres” est désactivé par défaut quand une VM est soit dans l'état “en fonction”, soit “sauvegardée”. Ceci simplement car la boîte de dialogue des paramètres vous permet de modifier des caractéristiques fondamentales de l'ordinateur virtuel créé pour votre système d'exploitation invité, et ce système d'exploitation peut ne pas bien supporter cela quand, par exemple, la moitié de sa mémoire lui est coupée sous les pieds. Il s'en suit que si le bouton “Paramètres” est désactivé, éteignez d'abord la VM actuelle.

VirtualBox offre pléthore de paramètres qu'on peut changer pour une machine virtuelle. Tous les paramètres modifiables dans la fenêtre “Paramètres” sont décrits en détails au chapitre 3, *Configurer des machines virtuelles*, page 48. Encore plus de paramètres sont disponibles avec l'interface en ligne de commande de VirtualBox ; voir chapitre 8, *VBoxManage*, page 119.

1.12 Supprimer des machines virtuelles

Pour supprimer une machine virtuelle dont vous n'avez plus besoin, effectuez un clic droit sur la liste de VMs du gestionnaire et sélectionnez “Supprimer” dans le menu contextuel qui apparaît.

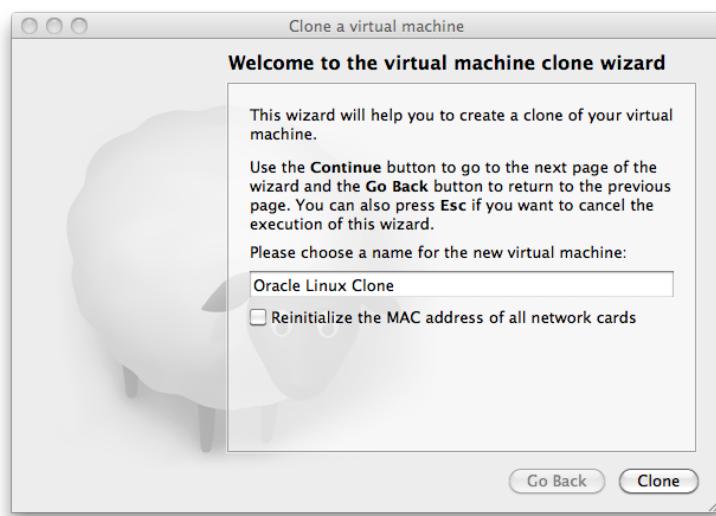
Une fenêtre de confirmation apparaîtra, vous permettant de sélectionner s'il faut supprimer la machine seulement de la liste des machines ou également les fichiers associés.

L'élément “Supprimer” est désactivé quand une machine est en fonction.

1.13 Cloner des machines virtuels

Pour expérimenter une configuration de VM, tester différents niveaux d'OS invité ou sauvegarder une VM, VirtualBox peut créer une copie complète ou liée d'une VM existante.⁵

Un assistant vous guidera dans le processus de clonage :



On peut appeler cet assistant à partir du menu contextuel de la liste des VMs du gestionnaire (en sélectionnant “Cloner”) ou de la vue “Dépôts” de la VM sélectionnée. Choisissez d'abord un @ouveau nom pour le clone. Quand vous sélectionnez **Réinitialiser l'adresse MAC de toutes les cartes réseaux**, toutes les cartes réseaux se voient affecter une nouvelle adresse MAC. Ceci est utile quand la VM source et celle clonée doivent fonctionner sur le même réseau. Si vous ne changez rien, toutes les cartes réseaux auront la même adresse MAC que celle de la VM source. Selon comment vous appelez l'assistant, vous avez différents choix pour l'opération de clonage. Vous devez d'abord décider si le clone devrait être lié à la VM source ou être totalement indépendant :

- **Clone complet** : Dans ce mode, toutes les images de disques dépendantes sont copiées dans le nouveau dossier de la VM. Le clone peut fonctionner complètement sans la VM d'origine.
- **Clone lié** : Dans ce mode, de nouvelles images de différenciation du disque sont créées là où les images de disque parents se trouvent à l'origine. Si vous avez sélectionné l'état actuel de la VM source comme point de clonage, un nouvel instantané sera créé implicitement.

Après avoir sélectionné le mode de clonage, vous devez décider ce que vous voulez exactement cloner. Vous pouvez toujours créer un clone de *l'état actuel* seulement, ou de *tout*. Quand vous sélectionnez *tout*, l'état actuel et, également, tous les instantanés seront clonés. Si vous partez

⁵Le support du clonage a été introduit avec VirtualBox 4.1.

d'un instantané ayant des fils en plus, vous pouvez également cloner *l'état actuel et tous ses fils*. Ceci crée un clone à partir de cet instantané et inclut tous les instantanés fils.

La durée de l'opération de clonage dépend de la taille et du nombre d'images de disques attachées. Garder aussi en tête que tout instantané a des images de différenciation de disques rattachées, qu'il faut aussi cloner.

L'icône "Cloner" est désactivée quand une machine est en fonction.

Pour savoir comment cloner une VM en ligne de commande, merci de voir chapitre 8.9, [VBox-Manage clonevm](#), page 137.

1.14 Importer et exporter des machines virtuelles

VirtualBox peut importer et exporter des machines virtuelles au format du standard industriel Open Virtualization Format (OVF).⁶

OVF est un standard de plateforme croisée supporté par de nombreux produits de virtualisation, permettant de créer des machines virtuelles prêtes à l'emploi importables ensuite dans un virtualiseur tel que VirtualBox. VirtualBox facilite l'expertation et l'importation d'OVF, en y donnant un accès et un support depuis la fenêtre du gestionnaire et son interface en ligne de commande. Cela permet d'empaqueter ce qu'on appelle des **applicatifs virtuels** : des images de disque et des paramètres de configuration qu'on peut distribuer facilement. De cette façon, chacun peut donner des logiciels dans un paquet prêt à utiliser complet (incluant système d'exploitation et applications) qui n'a pas besoin de configuration ou d'installation autre que l'importation dans VirtualBox.

Note: Le standard OVF est complexe et son support dans VirtualBox est un processus évolutif. En particulier, nous ne garantissons pas que VirtualBox supporte tous les applicatifs créés par d'autres logiciels de virtualisation. Pour une liste des limites connues, merci de voir le chapitre 14, [Limites connues](#), page 231.

Les applicatifs au format OVF peuvent apparaître sous deux variantes :

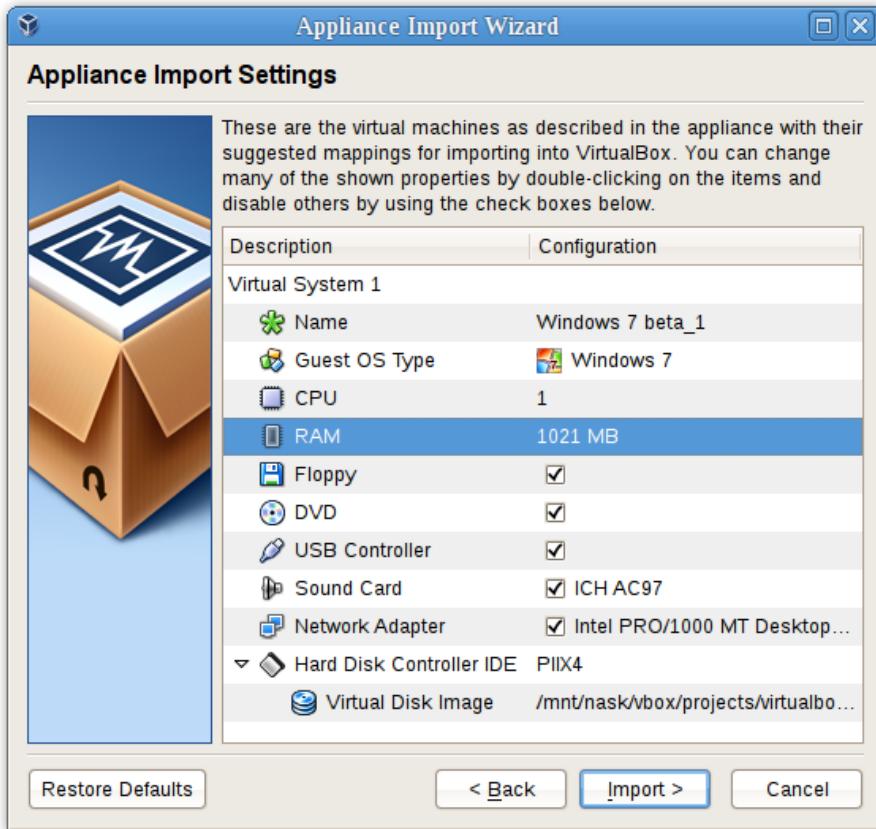
1. Ils peuvent être fournis en plusieurs fichiers, comme une ou plusieurs images de disque, en général au format universel VMDK (voir chapitre 5.2, [Fichiers images de disque \(VDI, VMDK, VHD, HDD\)](#), page 88) et un fichier de description en texte, au format XML avec une extension .ovf. Ces fichiers doivent résider dans le même répertoire pour que VirtualBox puisse les importer.
2. Sinon, les fichiers ci-dessus peuvent être empaquetés dans un seul fichier archive, ayant en général une extension .ova. (De tels fichiers archives utilisent une variante du format d'archivage TAR et peuvent donc être déballés hors de VirtualBox avec n'importe quel outil capable de déballer des fichiers TAR standards.)

Pour **importer** un applicatif dans l'un des formats ci-dessus, cliquez simplement deux fois sur le fichier OVF/OVA.⁷ Sinon, sélectionnez "Fichier" -> "Importer un applicatif" à partir de la fenêtre du gestionnaire. Dans la boîte de dialogue de fichier qui apparaît, allez sur le fichier ayant soit l'extension .ovf, soit .ova.

Si VirtualBox peut gérer le fichier, une boîte de dialogue ressemblant à celle suivante apparaîtra :

⁶Le support d'OVF a été introduit à l'origine avec VirtualBox 2.2 et a connu des améliorations essentielles au fur et à mesure des versions consécutives.

⁷À partir de la version 4.0, VirtualBox crée des associations de types de fichiers pour des fichiers OVF et OVA sur votre système d'exploitation hôte.



Elle présente les machines virtuelles décrites dans le fichier OVF et elle vous permet de modifier les paramètres de la machine virtuelle en cliquant deux fois sur les éléments de description. Une fois que vous avez cliqué sur “**Importer**”, VirtualBox copiera les images de disque et créera les machines virtuelles locales ayant les paramètres décrits dans la boîte de dialogue. Elles apparaîtront alors dans la liste de machines virtuelles du gestionnaire.

Remarquez que vu que les images de disque sont souvent grosses, et que les images VMDK fournis avec les applicatifs virtuels sont en général incluses dans un format compressé spécial inadapté à l’usage direct par les machines virtuelles, il faudra d’abord déballer les images et les copier, ce qui peut prendre quelques minutes.

Pour savoir comment importer une image en ligne de commande, merci de voir chapitre 8.10, [VBoxManage import](#), page 138.

Inversement, pour **exporter** des machines virtuelles déjà présentes dans VirtualBox, sélectionnez “Fichier” -> “Exporter un applicatif”. Une boîte de dialogue apparaîtra vous permettant de mettre plusieurs machines virtuelles dans un applicatif OVF. Puis, sélectionnez la destination où stocker les fichiers cibles, le processus de conversion commence. Cela peut mettre encore du temps.

Pour savoir comment exporter une image en ligne de commande, merci de voir chapitre 8.11, [VBoxManage export](#), page 139.

Note: OVF ne peut pas décrire d’instantanés pris pour une machine virtuelle. Il s’en suit que lorsque vous exportez une machine virtuelle avec des instantanés, seul l’état actuel de la machine sera exporté et les images de disque de l’export seront dans l’état “aplati”, identique à l’état actuel de la machine virtuelle.

1.15 Paramètres globaux

Vous pouvez atteindre la boîte de dialogue des paramètres globaux par le menu **Fichier**, en sélectionnant l'élément **Préférences...**. Elle offre une sélection de paramètres qui s'appliquent à toutes les machines virtuelles de l'utilisateur actuel ou, s'il y a des **Extensions**, à tout le système :

1. **Général** Permet à l'utilisateur de spécifier le dossier/répertoire par défaut des fichiers de la VM et la bibliothèque d'Authentification à RDP.
2. **Entrée** Permet à l'utilisateur de spécifier la touche Hôte. Elle indique la touche qui active si le curseur est sur le focus de la VM ou dans les fenêtres du système d'exploitation hôte (voir chapitre 1.8.2, *Capturer et rendre le clavier et la souris*, page 23) et qu'on utilise aussi pour activer certaines actions sur la VM (voir chapitre 1.8.3, *Taper des caractères spéciaux*, page 24)
3. **Mises à jour** Permet à l'utilisateur de spécifier divers paramètres des mises à jour automatiques.
4. **Langue** Permet à l'utilisateur de spécifier la langue de l'interface graphique.
5. **Affichage** Permet à l'utilisateur de préciser la résolution de l'écran, sa hauteur et sa largeur.
6. **Réseau** Permet à l'utilisateur de configurer les détails des réseaux Host Only.
7. **Extensions** Permet à l'utilisateur de lister et de gérer les paquets d'extension installés.
8. **Proxy** Permet à l'utilisateur de configurer un serveur proxy HTTP.

1.16 Interfaces alternatives

Comme indiqué brièvement à la chapitre 1.3, *Aperçu des fonctionnalités*, page 13, VirtualBox possède un concept interne très flexible qui permet d'utiliser plusieurs interfaces pour contrôler les mêmes machines virtuelles. Pour illustrer, vous pouvez par exemple démarrer une machine virtuelle avec la fenêtre du gestionnaire de VirtualBox puis l'arrêter en ligne de commande. Avec le support de bureau distant (RDP) de VirtualBox, vous pouvez même lancer des machines virtuelles à distance sur un serveur sans écran et rediriger la sortie graphique sur le réseau.

En détails, le paquet standard de VirtualBox inclut les interfaces suivantes :

1. **VirtualBox** est le gestionnaire VirtualBox. Cette interface graphique utilise l'environnement Qt ; la plupart de ce manuel de l'utilisateur est consacré à sa description. S'il est le plus facile à utiliser, certaines fonctionnalités plus avancées de VirtualBox n'y figurent pas pour garder sa simplicité.
2. **VBoxManage** est notre interface en ligne de commande pour avoir un contrôle automatisé et très détaillé de tous les aspects de VirtualBox. Il est décrit au chapitre 8, *VBoxManage*, page 119.
3. **VBoxSDL** est une interface graphique alternative simple, possédant des fonctionnalités volontairement limitées, conçue pour n'afficher que des machines virtuelles contrôlées en détail avec **VBoxManage**. C'est intéressant pour des environnements professionnels où l'affichage de tous les gadgets les boutons de l'interface graphique complète n'est pas faisable. **VBoxSDL** est décrit au chapitre 9.1, *VBoxSDL, l'afficheur simplifié de VM*, page 161.
4. Enfin, **VBoxHeadless** est encore une autre interface qui produit une sortie invisible sur l'hôte, mais qui agit exactement comme un serveur RDP si l'extension de bureau à distance de VirtualBox (VRDE) est installée. Contrairement aux autres interfaces graphiques, l'interface sans affichage n'a pas besoin de support graphique. Cela sert, par exemple, si

1 Premiers pas

vous voulez héberger vos machines virtuelles sur un serveur Linux sans affichage et sans système X Window installé. Pour les détails, voir chapitre [7.1.2, *VBoxHeadless, le serveur de bureau distant*](#), page [110](#).

Si les interfaces ci-dessus ne satisfont pas encore vos besoins particuliers, il est possible de créer d'autres interfaces avec le moteur de virtualisation complexe au cœur de VirtualBox, car le cœur de VirtualBox présente ouvertement toutes ses fonctionnalités dans une API propre ; merci de vous reporter au chapitre [11, *Interfaces de programmation de VirtualBox*](#), page [207](#).

2 Détails sur l'installation

Comme l'installation de VirtualBox varie selon votre système d'exploitation hôte, nous fournissons des instructions d'installation dans quatre chapitres distincts, respectivement pour Windows, Mac OS X, Linux et Solaris.

2.1 Installation sur des hôtes Windows

2.1.1 Prérequis

Pour les différentes versions de Windows que nous supportons en tant que systèmes d'exploitation hôtes, merci de vous reporter au chapitre 1.4, *Systèmes d'exploitation hôtes supportés*, page 16.

En outre, l'installateur Windows 1.1 ou supérieur doit être présent sur votre système. Cela devrait être le cas si vous avez installé les mises à jour récentes de Windows.

2.1.2 Effectuer l'installation

L'installation de VirtualBox peut se lancer

- soit en cliquant deux fois sur son fichier exécutable (contenant les architectures 32 et 64 bits)
- soit en entrant

```
VirtualBox.exe -extract
```

sur la ligne de commande. Cela va extraire les deux installateurs dans un répertoire temporaire où vous trouverez ensuite les fichiers .MSI habituels. Puis vous pouvez faire un

```
msiexec /i VirtualBox-<version>-MultiArch_-<x86|amd64>.msi
```

pour lancer l'installation.

Dans tous les cas, ceci affichera la boîte de dialrgue de bienvenue dans l'installation et vous permet de choisir où installer VirtualBox et les composants à installer. Outre l'application VirtualBox, les composants suivants sont disponibles :

Support USB Ce paquet contient des pilotes spéciaux pour votre hôte Windows dont a besoin VirtualBox pour supporter pleinement les périphériques USB dans vos machines virtuelles.

Réseau Ce paquet contient les pilotes réseaux supplémentaires pour votre hôte Windows, dont a besoin VirtualBox pour supporter le réseau bridgç (pour que les cartes réseaux de votre VM soient accessibles depuis d'autres machines de votre réseau physique).

Support Python Ce paquet contient le support de scriptage Python pour l'API de VirtualBox (voir chapitre 11, *Interfaces de programmation de VirtualBox*, page 207). Pour qu'ils fonctionnent, une installation de Python sur Windows opérationnelle est requise sur le système.¹

¹Voir, par exemple, <http://www.python.org/download/windows/>.

2 Détails sur l'installation

Selon votre configuration de Windows, il se peut que vous voyez des avertissements de “pilotes non signés” ou équivalent. Merci de sélectionner “Continuer” sur ces avertissements, sans quoi VirtualBox pourrait ne pas fonctionner correctement après l'installation.

L'installateur va créer un groupe “VirtualBox” dans le menu “Démarrer” de Windows qui vous permet de lancer l'application et d'accéder à sa documentation.

Avec des paramètres standards, VirtualBox sera installé pour tous les utilisateurs du système local. Si ce n'est pas ce que vous voulez, vous devez appeler l'installateur en l'extrayant d'abord via

```
VirtualBox.exe -extract
```

puis en faisant comme suit :

```
VirtualBox.exe -msiparams ALLUSERS=2
```

ou

```
msiexec /i VirtualBox-<version>-MultiArch_<x86|amd64>.msi ALLUSERS=2
```

sur les fichiers .MSI extraits. Ceci n'installera VirtualBox que pour l'utilisateur actuel.

Si vous ne voulez pas installer toutes les fonctionnalités de VirtualBox, vous pouvez régler le paramètre ADDLOCAL en option pour nommer explicitement les fonctionnalités à installer. Les fonctionnalités suivantes sont disponibles :

VBoxApplication Binaires principaux de VirtualBox.

Note: Il ne faut pas que cette fonctionnalité soit absente car elle contient l'ensemble de fichiers minimum pour que l'installation de VirtualBox fonctionne.

VBoxUSB Support USB.

VBoxNetwork Tout le support réseau ; y compris les fonctionnalités VBoxNetworkFlt et VBoxNetworkAdp (voir ci-dessous).

VBoxNetworkFlt Support du réseau bridgé.

VBoxNetworkAdp Support du réseau Host-only.

VBoxPython Support Python.

Par exemple, pour n'installer que le support USB avec les binaires principaux, faites un :

```
VirtualBox.exe -msiparams ADDLOCAL=VBoxApplication,VBoxUSB
```

ou un

```
msiexec /i VirtualBox-<version>-MultiArch_<x86|amd64>.msi ADDLOCAL=VBoxApplication,VBoxUSB
```

2.1.3 Désinstallation

Comme VirtualBox utilise l'installateur standard de Microsoft Windows, vous pouvez désinstaller VirtualBox en sécurité n'importe quand en choisissant l'entrée du programme dans le menu “Ajouter/supprimer des programmes” du panneau de configuration de Windows.

2.1.4 Installation sans efforts

Vous pouvez effectuer des installations sans efforts en utilisant le support MSI standard.

2.2 Installation sur des hôtes Mac OS X

2.2.1 Effectuer l'installation

Pour les hôtes Mac OS X, VirtualBox est dans un fichier image de disque (dmg). Effectuez les étapes suivantes :

1. Cliquez deux fois sur ce fichier pour monter son contenu.
2. Une fenêtre s'ouvrira, vous disant de cliquer deux fois sur le fichier de l'installateur `VirtualBox.mpkg` affiché dans cette fenêtre.
3. Ceci démarrera l'installateur, qui vous permettra de sélectionner où installer VirtualBox.

Après l'installation, vous pouvez voir une icône VirtualBox dans le dossier "Applications" du Finder.

2.2.2 Désinstallation

Pour désinstaller VirtualBox, réouvez le fichier image de disque (dmg) et cliquez deux fois sur l'icône Désinstaller dedans.

2.2.3 Installation sans efforts

Pour effectuer une installation non interactive de VirtualBox, vous pouvez la version en ligne de commande de l'installateur.

Montez le fichier image du disque (dmg) comme décrit dans l'installation normale. Ouvrez ensuite une session de terminal et exécutez :

```
sudo installer -pkg /Volumes/VirtualBox/VirtualBox.mpkg \
    -target /Volumes/Macintosh\ HD
```

2.3 Installing sur des hôtes Linux

2.3.1 Prérequis

Pour les diverses versions de Linux que nous supportons comme systèmes d'exploitation hôtes, merci de vous reporter au chapitre [1.4, Systèmes d'exploitation hôtes supportés](#), page 16.

Vous devrez installer les paquets suivants sur votre système Linux avant de commencer l'installation (certains systèmes feront cela pour vous automatiquement quand vous installerez VirtualBox) :

- Qt 4.4.0 ou supérieur ;
- SDL 1.2.7 ou supérieur (cette bibliothèque graphique s'appelle en général `libsdl` ou équivalent).

Note: Pour être précis, ces paquets ne sont requis que si vous voulez lancer les interfaces graphiques de VirtualBox. En particulier, VirtualBox, le gestionnaire graphique de VirtualBox, a besoin de Qt et de SDL ; VBoxSDL, notre interface graphique simplifiée, exige seulement SDL. Inversement, si vous ne voulez que VBoxHeadless, ni Qt ni SDL ne sont requis.

2.3.2 Le module noyau de VirtualBox

VirtualBox utilise un module spécial du noyau appelé `vboxdrv` pour effectuer des affectations de mémoire physique et contrôler le processeur pour exécuter les systèmes invités. Sans ce module du noyau, vous pouvez utiliser le gestionnaire de VirtualBox pour configurer des machines virtuelles mais elles ne démarreront pas. En outre, existe des modules noyau `vboxnetflt` et `vboxnetadp` requis pour les fonctionnalités réseau plus avancées de VirtualBox.

Le module noyau de VirtualBox est automatiquement installé sur votre système quand vous installez VirtualBox. Pour le maintenir avec les futures mises à jour du noyau, pour les distributions Linux qui le permettent – la plupart de celles actuelles – nous recommandons d'installer Dynamic Kernel Module Support (DKMS)². Cet environnement vous aide à construire et à mettre à jour les modules du noyau.

Si DKMS n'est pas déjà installé, exécutez une des procédures suivantes :

- Sur un système Ubuntu :

```
sudo apt-get install dkms
```

- Sur un système Fedora :

```
yum install dkms
```

- Sur un système Mandriva ou Mageia :

```
urpmi dkms
```

Si DKMS est disponible et installé, le module du noyau VirtualBox devrait toujours fonctionner automatiquement et il sera automatiquement reconstruit si vous mettez à jour le noyau de votre hôte.

Sinon, il n'y a que deux situations où vous devrez vous préoccuper du module noyau :

1. L'installation échoue dès le départ. Cela signifie probablement que votre système Linux n'est pas prêt à construire des modules noyau externes.

La plupart des distributions Linux peuvent être paramétrées simplement en installant les bons paquets - en principe il s'agit du compilateur GNU (GCC), GNU Make (make) et des paquets contenant les fichiers d'en-tête de votre noyau - et en s'assurant que toutes les mises à jour du système sont installées et que le système exécute le noyau le plus récent proposé par la distribution. *Les numéros de version des paquets des fichiers d'en-tête doivent être les mêmes que celui du noyau que vous utilisez.*

- Avec les versions Debian et Ubuntu, vous devez installer la bonne version de `linux-headers` et s'il existe, le paquet `linux-kbuild`. Les versions actuelles d'Ubuntu devraient avoir installé les bons paquets par défaut.
 - Dans les versions de Debian et d'Ubuntu encore plus anciennes, vous devez installer la bonne version du paquet `kernel-headers`.
 - Sur les systèmes Fedora et Redhat, le paquet est `kernel-devel`.
 - Sur SUSE et openSUSE Linux, vous devez installer les bonnes versions des paquets `kernel-source` et `kernel-syms`.
 - Si vous avez construit votre propre noyau, vous devrez vous assurer d'avoir installé toutes les en-têtes requises et d'autres fichiers au bon endroit pour construire les modules externes. Les détails sur la manière de faire cela dépendent de la façon dont vous avez construit votre noyau et si vous n'êtes pas sûr, vous devriez consulter la documentation que vous avez suivie pour construire.
2. Le noyau de votre hôte Linux a été mis à jour et DKMS n'est pas installé. Dans ce cas, le module du noyau devra être réinstallé en exécutant (en tant qu'administrateur) :

```
/etc/init.d/vboxdrv setup
```

²Voir http://en.wikipedia.org/wiki/Dynamic_Kernel_Module_Support pour une introduction.

2.3.3 Effectuer l'installation

VirtualBox est disponible de base dans un certain nombre de formats de paquet pour diverses distributions Linux classiques (voir chapitre [1.4, Systèmes d'exploitation hôtes supportés](#), page [16](#) pour les détails). En outre, un installateur générique alternatif (.run) devrait fonctionner sur la plupart des distributions Linux.

2.3.3.1 Installer VirtualBox à partir d'un paquet Debian/Ubuntu

Tout d'abord, téléchargez le paquet adapté à votre distribution. Les exemples suivants supposent que vous installez sur un système Ubuntu Karmic 32 bits. Utilisez `dpkg` pour installer le paquet Debian :

```
sudo dpkg -i VirtualBox-3.2_4.2.51_Ubuntu_karmic_i386.deb
```

On vous demandera d'accepter la licence d'utilisation Personnelle et d'évaluation de VirtualBox. Sauf si vous répondez "oui" l'installation sera annulée.

L'installateur cherchera aussi un module noyau VirtualBox adapté à votre noyau. Le paquet comprend des modules précompilés pour la plupart des configurations de noyau classiques. Si aucun module noyau adapté n'est trouvé, le script d'installation essaie de construire un module lui-même. Si le processus de construction ne réussit pas, on vous affichera un avertissement et le paquet ne sera pas configuré. Merci de jeter un œil sur `/var/log/vbox-install.log` pour voir pourquoi la compilation a échoué. Il se peut que vous deviez installer les en-têtes de Linux appropriées (voir chapitre [2.3.2, Le module noyau de VirtualBox](#), page [40](#)). Après avoir corrigé les problèmes, faites

```
sudo /etc/init.d/vboxdrv setup
```

Ceci démarrera un nouvel essai de construction du module.

Si un module de noyau convenable a été trouvé dans le paquet, ou si le module a été construit avec succès, le script d'installation essaiera de charger ce module. Si ceci échoue, merci de voir chapitre [12.7.1, Le module du noyau Linux refuse de se charger](#), page [221](#) pour plus d'informations.

Une fois que VirtualBox s'est installé et configuré avec succès, vous pouvez le démarrer en sélectionnant "VirtualBox" dans votre menu démarrer ou en ligne de commande (voir chapitre [2.3.5, Démarrer VirtualBox sur Linux](#), page [45](#)).

2.3.3.2 Utiliser l'installateur alternatif (VirtualBox.run)

L'installateur alternatif effectue les étapes suivantes :

- Il déballe les fichiers de l'application dans le répertoire cible,
`/opt/VirtualBox/`
 qu'on ne peut pas modifier.
- Il construit les modules noyau de VirtualBox (`vboxdrv`, `vboxnetflt` et `vboxnetadp`) et il les installe.
- Il crée `/etc/init.d/vboxdrv`, un script d'installation pour démarrer le module noyau de VirtualBox.
- Il crée un nouveau groupe système appelé `vboxusers`.
- Il crée des liens symboliques `/usr/bin` vers un script shell (`/opt/VirtualBox/VBox`) qui fait des vérifications de propriété et qui répartit les exécutables finaux, `VirtualBox`, `VBoxSDL`, `VBoxVRDP`, `VBoxHeadless` et `VBoxManage`

2 Détails sur l'installation

- Il crée `/etc/udev/rules.d/10-vboxdrv.rules`, un fichier de description pour udev, s'il est présent, pour rendre les périphériques USB accessibles par tous les utilisateurs du groupe `vboxusers`.
- Il écrit le répertoire d'installation `/etc/vbox/vbox.cfg`.

L'installateur doit être exécuté en tant qu'administrateur, soit avec `install` soit avec `uninstall`, en premier paramètre.

```
sudo ./VirtualBox.run install
```

Ou si vous n'avez pas de commande “`sudo`” disponible, lancez ceci plutôt en tant qu'administrateur :

```
./VirtualBox.run install
```

Après cela, vous devez mettre tous les utilisateurs qui devraient pouvoir accéder à des périphériques USB depuis des invités VirtualBox dans le groupe `vboxusers`, soit via les outils de gestion graphiques, soit en lançant la commande suivante en tant qu'administrateur :

```
sudo usermod -a -G vboxusers username
```

Note: La commande `usermod` de certaines distributions Linux anciennes ne supporte pas l'option `-a` (qui ajoute l'utilisateur au groupe donné sans le rendre membre d'autres groupes). Dans ce cas, cherchez les membres actuels du groupe avec la commande `groups` et ajoutez tous ces groupes dans une liste séparée par des virgules à la ligne de commande dans l'option `-G`, comme ceci : `usermod -G group1,group2,vboxusers username`.

2.3.3.3 Effectuer une installation manuelle

Si, pour une raison quelconque, vous ne pouvez pas utiliser l'installateur en script shell décrit précédemment, vous pouvez aussi effectuer une installation manuelle. Appelez l'installateur comme ceci :

```
./VirtualBox.run --keep --noexec
```

Ceci déballera tous les fichiers nécessaires à l'installation dans le répertoire `install` sous le répertoire actuel. Les fichiers de l'application VirtualBox sont contenus dans `VirtualBox.tar.bz2` que vous pouvez déballer dans n'importe quel répertoire de votre système. Par exemple :

```
sudo mkdir /opt/VirtualBox
sudo tar jxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
```

ou en tant qu'administrateur :

```
mkdir /opt/VirtualBox
tar jxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
```

Les sources du module noyau de VirtualBox sont fournies dans le répertoire `src`. Pour construire le module, allez dans le répertoire et lancez

```
make
```

Si tout se construit correctement, lancez la commande suivante pour installer le module dans le bon répertoire de modules :

2 Détails sur l'installation

```
sudo make install
```

Si vous n'avez pas sudo, passez en compte administrateur et lancez

```
make install
```

Le module de noyau VirtualBox a besoin d'un nœud de périphérique pour fonctionner. La commande make ci-dessus vous dira comment créer le nœud de périphérique selon votre système Linux. La procédure est légèrement différente pour une installation Linux classique avec le répertoire /dev, un système avec devfs, maintenant obsolète, et un système Linux moderne avec udev.

Sur certaines distributions Linux, vous pourriez rencontrer des difficultés pour construire le module. Vous devrez analyser les messages d'erreur du système de construction pour trouver la cause des problèmes. En général, assurez-vous que les sources du noyau Linux actuel sont utilisées pour le processus de construction.

Remarquez que le nœud de périphérique /dev/vboxdrv du module noyau doit appartenir à root:root et n'être autorisé en lecture/écriture que pour l'utilisateur.

Ensuite, vous devrez installer le script d'initialisation du système pour le module du noyau :

```
cp /opt/VirtualBox/vboxdrv.sh /etc/init.d/vboxdrv
```

(en supposant que vous avez installé VirtualBox dans le répertoire /opt/VirtualBox) et activé le script d'initialisation en utilisant la méthode adaptée à votre distribution. Vous devriez créer un fichier de configuration pour VirtualBox :

```
mkdir /etc/vbox
echo INSTALL_DIR=/opt/VirtualBox > /etc/vbox/vbox.cfg
```

et, par commodité, créer les liens symboliques suivants :

```
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VirtualBox
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxManage
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxHeadless
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxSDL
```

2.3.3.4 Mettre à jour et désinstaller VirtualBox

Avant de mettre à jour ou de désinstaller VirtualBox, vous devez fermer toutes les machines virtuelles actuellement en fonction et quitter les applications VirtualBox ou VBoxSVC. Pour mettre à jour VirtualBox, lancez simplement l'installateur de la version à jour. Pour désinstaller VirtualBox, appelez l'installateur comme ceci :

```
sudo ./VirtualBox.run uninstall
```

ou, en tant qu'administrateur

```
./VirtualBox.run uninstall
```

. À partir de la version 2.2.2, vous pouvez désinstaller le paquet .run en appelant

```
/opt/VirtualBox/uninstall.sh
```

Pour désinstaller VirtualBox à la main, inversez simplement les étapes de l'installation manuelle.

2.3.3.5 Installation automatique des paquets Debian

Les paquets Debian solliciteront une intervention de l'utilisateur lors de la première installation. On utilise le système debconf pour effectuer cette tâche. Pour empêcher toute intervention de l'utilisateur pendant l'installation, vous pouvez définir des valeurs par défaut. Un fichier vboxconf peut contenir les paramètres de debconf suivants :

```
virtualbox virtualbox/module-compilation-allowed boolean true
virtualbox virtualbox/delete-old-modules boolean true
```

La première ligne

permet la compilation du module noyau vboxdrv si aucun module n'a été trouvé pour le noyau actuel. La deuxième ligne permet au paquet d'effacer les anciens modules vboxdrv compilés par des installations précédentes.

Ces paramètres par défaut peuvent être appliqués avec

```
debconf-set-selections vboxconf
```

avant l'installation du paquet Debian VirtualBox.

De plus, il existe des options de configuration classiques que vous pouvez définir avant l'installation, décrites au chapitre [2.3.3.7, Options d'installation automatique](#), page [44](#).

2.3.3.6 Installation automatique des paquets .rpm

Le format .rpm n'offre pas de système de configuration comparable au système debconf. S'voir chapitre [2.3.3.7, Options d'installation automatique](#), page [44](#) pour savoir comment définir certaines options d'installation classiques fournies par VirtualBox.

2.3.3.7 Options d'installation automatique

Pour configurer le processus d'installation de nos paquets .deb et .rpm, vous pouvez créer un fichier de réponses appelé `/etc/default/virtualbox`. La génération automatique de la règle udev peut être empêchée par le paramètre suivant :

```
INSTALL_NO_UDEV=1
```

La création du groupe `vboxusers` peut être empêchée avec

```
INSTALL_NO_GROUP=1
```

Si la ligne

```
INSTALL_NO_VBOXDRV=1
```

est spécifiée, l'installeur du paquet n'essaiera pas de construire le module noyau `vboxdrv` si aucun module adapté au noyau actuel n'a été trouvé.

2.3.4 Le groupe `vboxusers`

Les installeurs pour Linux créent le groupe d'utilisateurs système `vboxusers` pendant l'installation. Tous les utilisateurs du système qui vont utiliser les périphériques USB depuis des invités VirtualBox doivent être membres de ce groupe. Un utilisateur peut devenir membre du groupe `vboxusers` via l'outil graphique de gestion des utilisateurs/groupes ou en ligne de commande avec

```
sudo usermod -a -G vboxusers nomutilisateur
```

2.3.5 Démarrer VirtualBox sur Linux

La façon la plus simple de démarrer un programme VirtualBox est de lancer le programme de votre choix (VirtualBox, VBoxManage, VBoxSDL ou VBoxHeadless) à partir d'un terminal. Ce sont des liens symboliques vers `VBox.sh` qui démarrent pour vous le programme nécessaire.

Les instructions détaillées suivantes ne devraient vous intéresser que si vous souhaitez exécuter VirtualBox sans l'installer préalablement. Vous devriez commencer par compiler le module noyau `vboxdrv` (voir ci-dessus) et l'insérer dans le noyau Linux. VirtualBox consiste en un démon de service (`VBoxSVC`) et plusieurs applications. Le démon est démarré automatiquement si nécessaire. Toutes les applications VirtualBox communiqueront avec le démon par les sockets du démon local d'Unix. Il peut y avoir plusieurs instances de démon sous différents comptes utilisateurs et les applications peuvent ne communiquer qu'avec le démon en fonction sous le compte utilisateur en tant qu'application. La socket de domaine locale réside dans un sous-répertoire du répertoire des fichiers temporaires de votre système, appelé `.vbox-<username>-ipc`. En cas de problèmes de communication ou de problèmes au démarrage du serveur, vous pouvez essayer de supprimer ce répertoire.

Toutes les applications de VirtualBox (VirtualBox, VBoxSDL, VBoxManage et VBoxHeadless) exigent que le répertoire de VirtualBox soit dans le chemin de la bibliothèque :

```
LD_LIBRARY_PATH=. ./VBoxManage showvminfo "Windows XP"
```

2.4 Installation sur les hôtes Solaris

Pour les versions spécifiques de Solaris que nous supportons comme systèmes d'exploitation hôtes, merci de vous reporter au chapitre [1.4, Systèmes d'exploitation hôtes supportés](#), page [16](#).

Si vous avez déjà une instance installée de VirtualBox sur votre hôte Solaris, merci de la désinstaller avant d'installer une nouvelle instance. Reportez-vous à chapitre [2.4.4, Désinstallation](#), page [46](#) pour les instructions de désinstallation.

2.4.1 Effectuer l'installation

VirtualBox est disponible en paquet Solaris standard. Téléchargez le paquet SunOS VirtualBox qui comprend les versions 32 et 64 bits de VirtualBox. *L'installation doit se faire en tant qu'administrateur et depuis l'espace global car l'installateur de VirtualBox charge des pilotes noyau qui ne peuvent pas l'être depuis des espaces non globaux.* Pour vérifier là où vous êtes actuellement, lancez la commande `zonename`. Exécutez les commandes suivantes :

```
gunzip -cd VirtualBox-4.2.51-SunOS.tar.gz | tar xvf -
```

À partir de VirtualBox 3.1, le paquet noyau de VirtualBox n'est plus distinct et il a été intégré au paquet principal. Installez le paquet VirtualBox en utilisant :

```
pkgadd -d VirtualBox-4.2.51-SunOS.pkg
```

Note: Si vous utilisez les Espaces Solaris, pour n'installer VirtualBox que dans celui actuel et pas dans une autre, utilisez `pkgadd -G`. Pour plus d'informations, reportez-vous au manuel de `pkgadd` ; voir aussi chapitre [2.4.6, Configurer un espace pour exécuter VirtualBox](#), page [47](#).

L'installateur vous demandera alors d'entrer le paquet que vous voulez installer. Choisissez "1" ou "tous" et poursuivez. Ensuite, l'installateur vous demandera si vous voulez autoriser l'exécution du script `postinstall`. Choisissez "y" et poursuivez, car il est fondamental d'exécuter ce script

qui installe le module noyau de VirtualBox. Suite à cette confirmation, l'installateur installra VirtualBox et exécutera le script de paramétrage postinstall.

Une fois que le script postinstall a été exécuté, votre installation est maintenant terminée. Vous pouvez maintenant effacer en toute sécurité de votre système le paquet décompressé et les fichiers autoresponse. VirtualBox devrait être installé dans /opt/VirtualBox.

2.4.2 Le groupe vboxuser

À partir de VirtualBox 4.1, l'installateur crée un groupe d'utilisateurs système `vboxuser` pendant l'installation sur les hôtes Solaris qui supportent les fonctionnalités USB exigées par VirtualBox. Tous les utilisateurs système qui vont utiliser des périphériques USB à partir des invités dans VirtualBox doivent faire partie de ce groupe. Un utilisateur peut devenir membre de ce groupe avec les outils de gestion graphiques des utilisateurs/groupes ou, en ligne de commande, en exécutant en tant qu'administrateur :

```
usermod -G vboxuser username
```

Remarquez que l'ajout d'un utilisateur actif à ce groupe exigera que l'utilisateur se déconnecte et se reconnecte. Vous devriez faire cela à la main après avoir installé avec succès le paquet.

2.4.3 Démarrer VirtualBox sur Solaris

La façon la plus simple de démarrer un programme VirtualBox est de lancer le programme de votre choix (VirtualBox, VBoxManage, VBoxSDL ou VBoxHeadless) à partir d'un terminal. Ce sont des liens symboliques vers `VBox.sh` qui démarre pour vous le programme requis.

Vous pouvez aussi appeler directement les programmes souhaités depuis /opt/VirtualBox. L'utilisation des liens fournis est plus facile car vous n'êtes pas obligé de taper le chemin complet.

Vous pouvez configurer certains éléments de l'interface graphique VirtualBox en Qt tels que la police et les couleurs en exécutant `VBoxQt config` à partir du terminal.

2.4.4 Désinstallation

La désinstallation de VirtualBox sur Solaris nécessite les droits d'administrateur. Pour effectuer la désinstallation, démarrez une session administrateur en terminal et exécutez :

```
pkgrm SUNWvbox
```

Après confirmation, ceci supprimera VirtualBox de votre système.

Si vous désinstallez VirtualBox version 3.0 ou inférieur, vous devez supprimer le paquet d'interface noyau VirtualBox, exécutez :

```
pkgrm SUNWvboxkern
```

2.4.5 Installation sans efforts

Pour effectuer une installation non interactive de VirtualBox, nous avons créé un fichier de réponses appelé `autoresponse` qu'nsilisera l'installateur pour répondre à ses questions plutôt que de vous les poser.

Extrayez le paquet `tar.gz` comme décrit dans l'installation normale. Puis ouvrez une session administrateur en terminal et exéutez :

```
pkgadd -d VirtualBox-4.2.51-SunOS-x86 -n -a autoresponse SUNWvbox
```

Pour effectuer une désinstallation non interactive, ouvrez une session administrateur en terminal et exéutez :

```
pkgrm -n -a /opt/VirtualBox/autoresponse SUNWvbox
```

2.4.6 Configurer un espace pour exécuter VirtualBox

À partir de VirtualBox 1.6, il est possible d'exécuter VirtualBox depuis des espaces Solaris. Pour une introduction aux espaces Solaris, merci de vous reporter à http://www.sun.com/bigadmin/features/articles/solaris_zones.jsp.

En supposant que VirtualBox a déjà été installé dans votre espace, vous devez donner au noeud de périphérique de VirtualBox un accès à la zone. Ceci se fait en effectuant les étapes suivantes. Démarrer une session administrateur en terminal et exécutez :

```
zonecfg -z vboxzone
```

Dans l'invite zonecfg, ajoutez la ressource device et les propriétés match à l'espace. Voici comment on peut faire :

```
zonecfg:vboxzone>add device
zonecfg:vboxzone:device>set match=/dev/vboxdrv
zonecfg:vboxzone:device>end
zonecfg:vboxzone>verify
zonecfg:vboxzone>exit
```

Si vous exécutez VirtualBox 2.2.0 ou supérieur sur des hôtes Solaris 11 ou Nevada, vous devriez aussi ajouter un périphérique pour /dev/vboxusbmon, identique à ce qui est indiqué ci-dessus. Cela ne s'applique pas aux hôtes Solaris 10 à cause de l'absence du support USB.

Remplacez "vboxzone" par le nom de l'espace dans lequel vous souhaitez exécuter VirtualBox. Ensuite, redémarrez l'espace en utilisant zoneadm et vous devriez pouvoir lancer VirtualBox depuis l'espace configuré.

3 Configurer des machines virtuelles

Alors que le chapitre 1, *Premiers pas*, page 11 vous donnait une introduction rapide à VirtualBox et sur la façon de faire marcher votre première machine virtuelle, le chapitre qui suit décrit en détail la manière de configurer des machines virtuelles.

Vous disposez d'une latitude considérable dans la décision sur le matériel qui sera donné à l'invité. Le matériel virtuel peut être utilisé pour communiquer avec le système hôte ou avec d'autres invités. Par exemple, si vous fournissez à VirtualBox l'image d'un CD-ROM dans un fichier ISO, VirtualBox peut présenter cette image à un système invité comme s'il s'agissait d'un CD-ROM physique. De la même façon, vous pouvez donner à un système invité un accès au réseau par sa carte réseau virtuelle, si tel est votre choix, donner au système hôte, à d'autres invités ou à des ordinateurs un accès Internet au système hôte.

3.1 Systèmes d'exploitation invités supportés

Comme VirtualBox est conçu pour offrir un environnement de virtualisation générique pour les systèmes x86, il peut exécuter des systèmes d'exploitation de tout type, même ceux non cités ici. Cependant, le principal est pour nous d'optimiser VirtualBox pour les systèmes invités suivants :

Windows NT 4.0 Toutes les versions, les éditions et les packs service sont complètement supportés ; cependant, il y a des problèmes avec les anciens packs service. Nous recommandons d'installer le pack service 6a. Les suppléments invité sont disponibles avec des fonctionnalités limitées.

Windows 2000 / XP / Server 2003 / Vista / Server 2008 / Windows 7 / Windows 8 / Server 2012

Toutes les versions, les éditions et les packs service sont complètement supportés (y compris les versions 64 bits, sous réserve des conditions préalables listées ci-dessous). Les suppléments invité sont disponibles.

DOS / Windows 3.x / 95 / 98 / ME Un test restreint a été effectué. Utilisation non recommandée au-delà des mécanismes d'installation primitive. Pas de suppléments invité disponibles.

Linux 2.4 Support limité.

Linux 2.6 Toutes les versions/éditions sont complètement supportées (32 bits et 64 bits). Les suppléments invité sont disponibles.

Nous recommandons fortement d'utiliser un noyau Linux version 2.6.13 ou supérieur pour une meilleure performance.

Note: Certaines versions du noyau Linux ont des bogues les empêchant de s'exécuter dans un environnement virtuel ; merci de voir chapitre 12.4.3, *Versions buggées du noyau Linux Linux 2.6*, page 218 pour des détails.

Solaris 10 (u6 et supérieur), Solaris 11 (y compris Solaris 11 Express) Complètement supporté (32 bits et 64 bits). Les suppléments invité sont disponibles.

FreeBSD Exige l'activation de la virtualisation matérielle. Support limité. Les suppléments invité ne sont pas encore disponibles.

OpenBSD Exige l'activation de la virtualisation matérielle. Les versions 3.7 et supérieur sont supportées. Les suppléments invité ne sont pas encore disponibles.

OS/2 Warp 4.5 Exige l'activation de la virtualisation matérielle. Nous ne supportons officiellement que MCP2 ; les autres versions d'OS/2 peuvent ou pas fonctionner. Les suppléments invité sont disponibles avec des fonctionnalités limitées.¹

Mac OS X VirtualBox 3.2 a ajouté un support expérimental des invités Mac OS X, mais il est fourni avec des restrictions. Merci de voir la section suivante et chapitre 14, *Limites connues*, page 231.

3.1.1 Inviés Mac OS X

À partir de la version 3.2, VirtualBox a un support expérimental des invités Mac OS X. Il vous permet d'installer et d'exécuter des versions non modifiées de Mac OS X sur du matériel hôte supporté.

Si les solutions concurrentes offrent des modifications des DVDs d'installation de Mac OS X (chargeur d'amorçage différent, fichiers remplacés), VirtualBox est le premier produit à offrir une architecture PC moderne qu'attend OS X sans besoin de "bidouilles".

Vous devriez garder en tête un certain nombre de **problèmes importants** avant d'essayer d'installer un invité Mac OS X :

1. Mac OS X est un logiciel propriétaire sous licence et il contient **des restrictions juridiques et techniques** limitant son utilisation à certains matériels et scénarios d'utilisations. Il est important que vous compreniez et que vous respectiez ces restrictions.

En particulier, pour la plupart des versions de Mac OS X, Apple interdit de les installer sur du matériel non Apple.

Ces restrictions juridiques sont également renforcées au niveau technique. Mac OS X vérifie s'il s'exécute sur du matériel Apple et la plupart des DVDs fournis avec le matériel Apple vérifient même le modèle exact. Ces restrictions *ne sont pas* contournées par VirtualBox et s'appliquent toujours.

2. Seuls des **processeurs** connus et testés par Apple sont supportés. Il s'en suit que si votre processeur Intel est plus récent que la construction de Mac OS X, ou si vous avez un processeur non Intel, il plantera très probablement au moment du démarrage avec une exception "Unsupported CPU". Il vaut mieux généralement utiliser le DVD de Mac OS X fourni avec votre matériel Apple.
3. L'installateur de Mac OS X s'attend à ce que le disque dur soit **partitionné**, donc quand il n'offre pas de sélection, vous devez lancer l'Outil de disque du menu "Outils" et partitionner le disque dur. Puis, fermez l'outil de disque et poursuivez l'installation.
4. En outre, comme le support de Mac OS X de VirtualBox est actuellement encore expérimental, merci de vous reporter aussi à chapitre 14, *Limites connues*, page 231.

3.1.2 Invités 64 bits

VirtualBox supporte les systèmes d'exploitation invités 64 bits, même sur des systèmes d'exploitation hôtes 32 bits,² pourvu qu'existent les conditions suivantes :

¹Voir chapitre 14, *Limites connues*, page 231.

²Le support des invités 64 bits a été ajouté avec VirtualBox 2.0 ; le support des invités 64 bits sur des hôtes 32 bits a été ajouté avec VirtualBox 2.1.

1. Vous avez besoin d'un processeur 64 bits avec le support de la virtualisation matérielle (voir chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201).
2. Vous devez activer la virtualisation matérielle pour la VM particulière où vous voulez le support du 64 bits ; la virtualisation logicielle n'est pas supportée pour les VMs 64 bits.
3. Si vous voulez utiliser un invité 64 bits sur un système d'exploitation hôte 32 bits, vous devez aussi sélectionner un système d'exploitation 64 bits pour la VM en particulier. Comme le support du 64 bits sur des hôtes 32 bits implique des dépassements supplémentaires, VirtualBox n'active ce support qu'à la demande expresse.

Sur des hôtes 64 bits (fournis en général avec le support de la virtualisation matérielle), les systèmes d'exploitation invités 64 bits sont toujours supportés, indépendamment des paramètres, donc vous pouvez simplement installer un système d'exploitation 64 bits dans l'invité.

Avertissement: Sur tous les hôtes, vous devriez activer le **I/O APIC** pour les machines virtuelles que vous souhaitez utiliser en mode 64 bits. C'est surtout vrai pour les VMs Windows 64 bits. Voir chapitre 3.3.2, *Onglet “Avancé”*, page 51. En outre, pour les invités Windows 64 bits, vous devriez vous assurer que la VM utilise le **périphérique réseau Intel**, car il n'y a pas de support pour le pilote 64 bits pour la carte AMD PCNet ; voir chapitre 6.1, *Matériel réseau virtuel*, page 98.

Si vous utilisez l'assistant “Créer une VM” de l'interface graphique de VirtualBox, (voir chapitre 1.7, *Créer votre première machine virtuelle*, page 19), VirtualBox utilisera automatiquement les bons paramètres pour chaque type de système d'exploitation 64 bits choisi.

3.2 Matériel émulé

VirtualBox virtualise presque tout le matériel de l'hôte. Selon la configuration de la VM, l'invité utilisera le matériel virtuel suivant :

- **Périphériques d'entrée.** Par défaut, VirtualBox émule un clavier et une souris PS/2 standards. Ces périphériques sont supportés par presque tous les systèmes d'exploitation passés et actuels.
En outre, VirtualBox peut fournir des périphériques d'entrée virtuels pour éviter de devoir capturer la souris et un clavier, comme décrit au chapitre 1.8.2, *Capturer et rendre le clavier et la souris*, page 23.
- **Graphisme.** Le périphérique graphique de VirtualBox (parfois cité comme périphérique VGA) n'est pas, contrairement à presque tous les autres périphériques émulés, basé sur un équivalent physique. C'est un périphérique de synthèse simple qui fournit une compatibilité avec les VGA et plusieurs registres étendus standards par la VESA BIOS Extensions (VBE).
- **Stockage.** VirtualBox émule actuellement l'interface ATA standard qu'on trouve dans les puces PIIX3/PIIX4 Intel, l'interface SATA (AHCI) et deux adaptateurs SCSI (LSI Logic et BusLogic) ; voir chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85 pour des détails. Bien qu'en fournir un d'entre eux suffirait pour VirtualBox en lui-même, cette multitude d'adaptateurs de stockage est requise pour des raisons de compatibilité avec d'autres hyperviseurs. Windows est particulièrement pointilleux sur ces périphériques d'amorçage, et la migration des VMs entre hyperviseurs est très difficile voire impossible si les contrôleurs de stockage sont différents.
- **Réseau.** Voir chapitre 6.1, *Matériel réseau virtuel*, page 98.

- **USB.** VirtualBox émule deux contrôleurs d'hôte USB, EHCI et OHCI. Deux contrôleurs hôtes sont nécessaires car l'OHCI ne gère que les périphériques USB à faible ou grande vitesse (USB 1.x et 2.0), alors que l'EHCI ne gère que les périphériques à très haute vitesse (USB 2.0 seulement). Les contrôleurs USB émulés ne communiquent pas directement avec les périphériques sur l'hôte, mais plutôt avec une couche USB virtuelle qui rend abstrait le protocole USB et qui permet d'utiliser des périphériques USB distants.
- **Audio.** Voir chapitre [3.7, Paramètres de son](#), page [58](#).

3.3 Paramètres généraux

Dans la fenêtre des paramètres, dans “Général”, vous pouvez configurer la plupart des aspects fondamentaux de la machine virtuelle, tels que la mémoire ou du matériel essentiel. Il y a trois onglets, “Base”, “Avancé” et “Description”.

3.3.1 Onglet “Base”

Sous l'onglet “Base” de la catégorie Général des paramètres, vous pouvez voir ces paramètres :

Nom Le nom sous lequel apparaît la VM dans la liste des VMs de la fenêtre principale. Sous ce nom, VirtualBox enregistre aussi les fichiers de configuration de la VM. En changeant le nom, VirtualBox renomme aussi ces fichiers. Il s'en suit que vous ne pouvez utiliser que des caractères autorisés par les noms de fichier de votre système d'exploitation hôte.

Remarquez qu'en interne, VirtualBox utilise des identifiants unique (UUIDs) pour identifier les machines virtuelles. Vous pouvez les afficher avec VBoxManage.

Système d'exploitation / version Le type du système d'exploitation invité qui est (ou sera) installé dans la VM. C'est le même paramètre que celui spécifié dans l'assistant “Nouvelle machine virtuelle”, comme décrit au chapitre [1.7, Créez votre première machine virtuelle](#), page [19](#).

Si les paramètres par défaut d'une VM nouvellement créée sont fonction du type de système d'exploitation sélectionné, la modification ultérieure du type n'a aucun effet sur les paramètres de la VM ; cette valeur est donc purement informative et décorative.

3.3.2 Onglet “Avancé”

Dossier d'instantané Par défaut, VirtualBox enregistre les données de l'instantané avec vos autres données de configuration de VirtualBox ; voir chapitre [10.1, Où VirtualBox stocke ses fichiers](#), page [196](#). Avec ce paramètre, vous pouvez spécifier un autre dossier pour chaque VM.

Presse-papier partagé Vous pouvez sélectionner ici si le presse-papier du système d'exploitation invité devrait être partagé avec celui de votre hôte. Si vous sélectionnez “Bidirectionnel”, VirtualBox s'assurera toujours que les deux presse-papier contiennent les mêmes données. Si vous sélectionnez “Hôte vers invité” ou “Invité vers hôte”, VirtualBox copiera les données du presse-papier dans une direction.

Le partage du presse-papier nécessite que les suppléments invité de VirtualBox soient installés. Il s'en suit que ce paramètre n'a aucun effet sans cela ; voir chapitre [4, Les suppléments invité](#), page [64](#) pour les détails.

Le presse-papier partagé est désactivé par défaut. Voir chapitre [13.3.2.3, Presse-papier](#), page [228](#) pour une explication. On peut changer ce paramètre n'importe quand en utilisant le menu “Presse-papier” du menu “Périphériques” de la machine virtuelle.

Média amovible : se rappeler des changements au moment de l'exécution Si ceci est coché, VirtualBox enregistrera l'état des médias montés entre deux exécutions d'une machine virtuelle.

Mini barre d'outils En mode plein écran ou transparent, VirtualBox peut afficher une petite barre d'outils contenant certains éléments disponibles normalement dans la barre de menus de la machine virtuelle. Cette barre d'outils se réduit à une ligne grise, sauf si vous déplacez la souris dessus. Avec la barre d'outils, vous pouvez basculer entre le mode plein écran et transparent, contrôler l'exécution de la machine ou activer certains périphériques. Si vous ne voulez pas voir la barre d'outils, désactivez ce paramètre.

3.3.3 Onglet “Description”

Ici, vous pouvez taper une description pour votre machine virtuelle si vous le voulez. Cela n'a aucun effet sur le fonctionnement de la machine, mais pourriez trouver cet espace utile pour y noter des choses comme la configuration d'une machine virtuelle et le logiciel installé dedans.

3.4 Paramètres système

La catégorie “Système” regroupe divers paramètres liés au matériel de base présenté à la machine virtuelle.

Note: Le mécanisme d'activation de Microsoft Windows étant sensible aux modifications matérielles, si vous modifiez les paramètres matériels pour un invité Windows, certains de ces changements peuvent provoquer la demande d'une nouvelle activation de Microsoft.

3.4.1 Onglet “Carte mère”

Dans l'onglet “Carte mère”, vous pouvez influencer le matériel virtuel qui serait normalement sur la carte mère d'un vrai ordinateur.

Mémoire de base Ceci définit la quantité de RAM affectée et donnée à la VM quand elle est en fonction. La quantité de mémoire spécifiée sera récupérée sur le système d'exploitation hôte, donc il faut qu'elle soit disponible ou rendue disponible comme mémoire libre sur l'hôte au moment du démarrage de la VM et elle ne sera pas disponible pour l'hôte tant que la VM sera en fonction. C'est le même paramètre que celui spécifié dans l'assistant “Nouvelle machine virtuelle”, comme décrit dans les grandes lignes du chapitre 1.7, [Créer votre première machine virtuelle](#), page 19 above.

En général, il est possible de modifier la taille de la mémoire après avoir installé le système d'exploitation invité (pourvu que vous ne réduisiez pas la mémoire à une quantité telle qu'un système d'exploitation ne démarrerait plus).

Ordre d'amorçage Ce paramètre détermine l'ordre dans lequel le système d'exploitation invité essaiera de s'amorcer sur les différents périphériques de démarrage virtuels. Analogique à un paramètre du BIOS d'un vrai PC, VirtualBox peut dire à un OS invité de démarrer sur une disquette, un lecteur CD/DVD virtuel, le disque dur virtuel (chacun d'eux étant défini par les autres paramètres de la VM), le réseau ou aucun.

Si vous sélectionnez “Réseau”, la VM essaiera de démarrer sur le mécanisme PXE du réseau virtuel. Il faut le configurer en détail en ligne de commande Merci de voir chapitre 8.8, [VBoxManage modifyvm](#), page 130.

Chipset Vous pouvez sélectionner ici le chipset qui sera présenté à la machine virtuelle. Avant VirtualBox 4.0, PIIX3 était ici la seule option disponible. Pour les systèmes d'exploitation invités modernes tels que Mac OS X, ce vieux chipset n'est plus bien supporté. Du coup, VirtualBox 4.0 a introduit une émulation du chipset ICH9, plus moderne, qui supporte le PCI express, trois bus PCI, des ponts PCI-à-PCI et des Message Signalled Interrupts (MSI). Cela permet aux systèmes d'exploitation modernes de gérer davantage périphériques PCI sans besoin d'un partage d'IRQ. Remarquez que le support ICH9 est expérimental et non recommandé pour les systèmes d'exploitation invités qui n'en ont pas besoin.

Activer I/O APIC Advanced Programmable Interrupt Controllers (APICs) (contrôleurs d'interruptions programmables avancés) sont des fonctionnalités du matériel x86 récent ayant remplacé ces dernières années le Programmable Interrupt Controllers (PICs). Avec un I/O APIC, les systèmes d'exploitation peuvent utiliser plus de 16 requêtes d'interruption (IRQs), évitant donc le partage d'IRQ pour une meilleure fiabilité.

Note: L'activation de I/O APIC est *requis* pour les systèmes d'exploitation invités 64 bits, surtout Windows Vista ; il est aussi nécessaire si vous voulez utiliser plus d'un processeur virtuel dans une machine virtuelle.

Cependant, le support des I/O APICs logiciel n'est pas fiable avec les systèmes d'exploitation autres que Windows. De plus, l'utilisation de I/O APIC augmente légèrement la charge de la virtualisation et donc, cela ralentit un peu l'OS invité.

Avertissement: Tous les systèmes d'exploitation Windows à partir de Windows 2000 installent des noyaux différents en fonction de la disponibilité de l'I/O APIC. Comme avec l'ACPI, l'I/O APIC *ne doit pas être désactivé après l'installation* d'un OS invité Windows. Son activation après l'installation n'aura par contre aucun effet.

Activer l'EFI Ceci active la Extensible Firmware Interface (EFI), qui remplace le BIOS de base et peut servir dans certains cas d'utilisation avancée. Merci de vous reporter à chapitre [3.12, Autre firmware \(EFI\)](#), page [62](#) pour les détails.

Horloge matérielle en temps UTC Si ceci est coché, VirtualBox indiquera à l'invité l'heure du système au format UTC plutôt qu'en temps local (hôte). Cela change la façon dont agit l'horloge en temps réel virtuelle (RTC) et peut être utile pour des systèmes d'exploitation invité de type Unix qui attendent généralement une horloge matérielle en UTC.

Activer le pointage absolu des périphériques Si vous l'activez, VirtualBox signale à la machine virtuelle qu'une tablette USB est présente et il communique les événements de la souris à la machine virtuelle par ce périphérique. Si vous le désactivez, les événements de la souris sont communiqués par le périphérique virtuel traditionnel de souris PS/2.

L'utilisation de la tablette USB virtuelle présente l'avantage que les mouvements sont transmis avec une coordination totale (et non comme des modifications de position relatives), ce qui permet à VirtualBox de traduire les événements de la souris dans la fenêtre de la VM en événements de tablette sans devoir "capturer" la souris pour l'invité comme décrit au chapitre [1.8.2, Capturer et rendre le clavier et la souris](#), page [23](#). Ceci rend l'utilisation de la VM moins pénible même si les suppléments invité ne sont pas installés.³

³The virtual USB tablet was added with VirtualBox 3.2. Selon le système d'exploitation invité sélectionné, ceci est maintenant activé par défaut pour les nouvelles machines virtuelles.

De plus, vous pouvez désactiver **Advanced Configuration and Power Interface (ACPI)** que VirtualBox présente par défaut au système d'exploitation invité. L'ACPI est le standard industriel actuel permettant au système d'exploitation de reconnaître le matériel, de configurer les cartes mères et d'autres périphériques, et de gérer l'énergie. Comme tous les PCs modernes contiennent cette fonctionnalité et Windows et Linux le supportent depuis des années, il est activé par défaut dans VirtualBox. On ne peut le désactiver qu'en ligne de commande ; voir chapitre 8.8, [VBoxManage modifyvm](#), page 130.

Avertissement: Tous les systèmes d'exploitation Windows à partir de Windows 2000 installent des noyaux différents selon que l'ACPI est activé, donc l'ACPI *ne doit pas être désactivé* après l'installation d'un OS invité Windows. Son activation après l'installation n'aura par contre aucun effet.

3.4.2 Onglet “Processeur”

Dans l'onglet “Processeur”, vous pouvez définir le nombre de **coeurs de processeur** virtuels que devrait voir le système d'exploitation invité. À partir de la version 3.0, VirtualBox supporte le symmetrical multiprocessing (SMP) et peut présenter jusqu'à 32 coeurs de processeur virtuels à chaque machine virtuelle.

Vous ne devriez quand même pas configurer une machines virtuelles pour utiliser plus de coeurs de processeur que vous n'en avez physiquement.

Dans cet onglet, vous pouvez aussi définir le exécution du processeur. Ce paramètre limite la quantité de temps qu'en processeur hôte peut donner pour émuler un processeur virtuel. Le réglage par défaut est de 100% ce qui veut dire qu'il n'y a aucune limite. Un réglage à 50% implique qu'un seul processeur peut utiliser jusqu'à 50% d'un seul processeur hôte. Remarquez que la limitation du temps d'exécution des processeurs virtuels peut provoquer des problèmes de timing dans l'invité.

Par ailleurs, le paramètre “**Activer PAE/NX**” détermine si les possibilités PAE et NX du processeur hôte seront proposées à la machine virtuelle. PAE signifie “Physical Address Extension”. Normalement, s'il est activé et supporté par le système d'exploitation, même un processeur x86 32 bits peut accéder à plus de 4 Go de RAM. Cela est rendu possible par l'ajout de 4 bits aux adresses mémoire, de sorte qu'avec 36 bits, on peut gérer jusqu'à 64 Go. Certains systèmes d'exploitation (tels qu'Ubuntu Server) exigent le support PAE du processeur et ne peuvent pas fonctionner dans une machine virtuelle sans cela.

Sur des machines virtuelles qui exécutent des systèmes d'exploitation serveurs modernes, VirtualBox supporte aussi le branchement à chaud du processeur. Pour des détails là-dessus, merci de vous reporter à chapitre 9.5, [Montage de processeur à chaud](#), page 170.

3.4.3 Onglet “Accélération”

Dans cet onglet, vous pouvez déterminer si VirtualBox devrait utiliser les extensions de virtualisation matérielle que peut supporter votre hôte et comment. C'est le cas avec la plupart des processeurs fabriqués après 2006.

Vous pouvez sélectionner individuellement pour chaque machine virtuelle si VirtualBox devrait utiliser la virtualisation logicielle ou matérielle.⁴

Dans la plupart des cas, les paramètres défaut conviendront ; VirtualBox aura choisi des options par défaut en fonction du système d'exploitation que vous avez sélectionné quand vous

⁴Avant VirtualBox version 2.2, la virtualisation logicielle était par défaut ; à partir de la version 2.2, VirtualBox activera la virtualisation matérielle par défaut pour les nouvelles machines virtuelles créées. (Les machines virtuelles existantes ne sont pas automatiquement modifiées pour des raisons de compatibilité, et vous pouvez bien sûr modifier le réglage par défaut pour chaque machine virtuelle.)

avez créé la machine virtuelle. Toutefois, dans certaines situations, il se peut que vous vouliez modifier ces réglages par défaut préconfigurés.

Il se peut que les utilisateurs avancés s'intéressent aux détails techniques sur la virtualisation logicielle vs matérielle ; merci de voir chapitre [10.3, Virtualisation matérielle vs. logicielle](#), page [201](#).

Si votre processeur hôte supporte les fonctions **pagination nested** (AMD-V) ou **EPT** (Intel VT-x), vous pouvez compter sur une augmentation significative des performances en activant la pagination nested en plus de la virtualisation matérielle. Pour des détails techniques, voir chapitre [10.6, Vagination nestée et VPIIDs](#), page [205](#).

3.5 Paramètres d'affichage

Taille de la mémoire graphique Ceci définit la taille de la mémoire fournie par la carte graphique virtuelle et disponible pour l'invité, en Mo. Comme avec la mémoire globale, la quantité spécifiée sera affectée à partir de la mémoire résidente de l'hôte. À partir de la quantité de mémoire graphique, certaines résolutions et profondeur des couleurs peuvent être disponibles.

L'interface graphique affichera un avertissement si la quantité de mémoire graphique est trop petite pour pouvoir passer la VM en mode plein écran. La valeur minimum dépend du nombre d'écrans virtuels, de la résolution de l'écran et de la profondeur des couleurs sur l'affichage hôte, ainsi que de l'activation de *l'accélération 3D* et de *l'accélération graphique 2D*. Une ébauche d'estimation est $profondeur des couleurs / 8) \times \text{nombre de pixels verticalement} \times \text{nombre de pixels horizontalement} \times \text{nombre d'écrans} = \text{nombre de bytes}$. Comme indiqué ci-dessus, de la mémoire supplémentaire pourrait être nécessaire pour définir les accélérations d'affichage activées.

Nombre d'écrans Avec ce paramètre, VirtualBox peut fournir plusieurs écrans virtuels à une machine virtuelle. Si un système d'exploitation invité (tel que Windows) supporte le rattachement à plusieurs écrans, VirtualBox peut affirmer que plusieurs écrans virtuels sont présents.⁵ Jusqu'à 8 écrans virtuels peuvent ainsi être supportés.

La sortie de plusieurs écrans virtuels peut être affichée sur l'hôte dans plusieurs fenêtres de VM placées côte à côte.

Néanmoins, en mode plein écran ou transparent, ils utiliseront les écrans physiques disponibles connectés à l'hôte. Il s'en suit que pour que les modes plein écran et transparent fonctionnent avec plusieurs écrans, vous aurez besoin d'au moins autant d'écrans physiques que d'écrans virtuels configurés, sans quoi VirtualBox renverra une erreur. Vous pouvez configurer la relation entre les écrans hôte et invité en utilisant le menu Vue, en appuyant sur Touche hôte + Origine pendant que vous êtes en mode plein écran ou transparent.

Merci de voir aussi chapitre [14, Limites connues](#), page [231](#).

Activer l'accélération 3D Si vous avez installé les Suppléments invité sur une machine virtuelle, vous pouvez sélectionner ici si l'invité devrait supporter la vidéo accélérée 3D. Merci de vous reporter à chapitre [4.4.1, Accélération 3D matérielle \(OpenGL and Direct3D 8/9\)](#), page [77](#) pour des détails.

Activer l'accélération graphique 2D Si vous avez installé les Suppléments invité sur une machine virtuelle contenant Windows, vous pouvez sélectionner ici si l'invité doit supporter la vidéo accélérée 2D. Merci de vous reporter à chapitre [4.4.2, L'accélération matérielle 2D pour les invités Windows](#), page [79](#) pour les détails.

⁵Le support multi-écrans a été ajouté avec VirtualBox 3.2.

Affichage distant Dans l'onglet “Affichage distant”, si vous avez installé l'extension VirtualBox Remote Display Extension (VRDE), vous pouvez activer le serveur VRDP construit dans VirtualBox. Cela vous permet de vous connecter à la console de la machine virtuelle à distance avec n'importe quel client RDP standard tel que `mstsc.exe` fourni avec Microsoft Windows. Sur les systèmes Linux et Solaris, vous pouvez utiliser le programme standard open-source `rdesktop`. Ces fonctionnalités sont décrites en détail au chapitre [7.1, Affichage distant \(VRDP support\)](#), page [109](#).

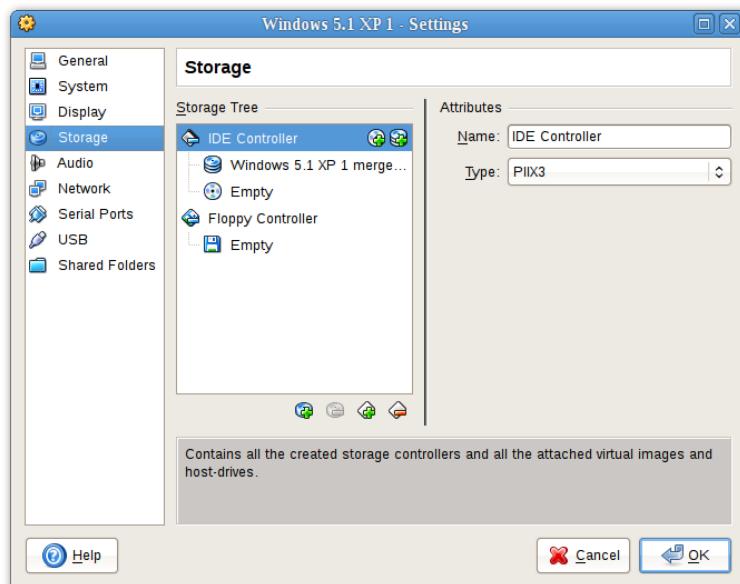
3.6 Paramètres du stockage

La catégorie “Stockage” des paramètres de la VM vous permet de connecter des images et des lecteurs de disque dur, de CD/DVD et de disquette virtuelles à votre machine virtuelle.

Dans un vrai PC, ce qu'on appelle les “contrôleurs de stockage” connectent des lecteurs de disque physiques au reste de l'ordinateur. De la même façon, VirtualBox présente des contrôleurs de stockage virtuels à une machine virtuelle. Sous chaque contrôleur, vous voyez les périphériques virtuels connectés (disques durs, lecteur CD/DVD ou disquette).

Note: Cette section ne peut vous donner qu'une brève introduction aux paramètres de stockage de VirtualBox. Comme VirtualBox vous donne un choix énorme d'options dans cette zone, nous avons consacré tout un chapitre de ce manuel de l'utilisateur à l'explication de tous les détails : merci de voir chapitre [5, Le stockage virtuel](#), page [85](#).

Si vous avez utilisé l'assistant “Créer une VM” pour créer une machine, vous verrez normalement quelque chose comme suit :



Selon le type de système d'exploitation invité que vous avez sélectionné quand vous avez créé la VM, la présentation classique des périphériques de stockage dans une nouvelle VM est ainsi :

- Vous verrez un **contrôleur IDE**, auquel on a connecté un lecteur CD/DVD virtuel (sur le “deuxième port maître”) du contrôleur IDE).
- Vous verrez aussi un **contrôleur SATA controller**, qui est un type plus moderne de contrôleur de stockage pour un transport de données du disque dur plus rapide, où on a

3 Configurer des machines virtuelles

rattaché les disques durs virtuels. Au départ, vous aurez en principe un disque comme celui-ci, mais comme vous pouvez le voir dans l'impression d'écran ci-dessus, vous pouvez en avoir plusieurs, chacun représenté par un fichier image de disque (fichiers (VDI, dans ce cas).

Si vous avez créé votre VM avec une version antérieure de VirtualBox, la présentation des stockages par défaut peut varier. Il y a plusieurs options que vous ayez un contrôleur IDE auquel on a connecté le lecteur CD/DVD et les disques durs. Cela pourrait aussi être le cas si vous avez sélectionné un ancien type de système d'exploitation quand vous avez créé la VM. Comme les anciens systèmes d'exploitation ne supportent pas le SATA sans pilotes supplémentaires, VirtualBox veillera à ce qu'aucun périphérique de ce type ne soit présent au départ. Merci de voir chapitre [5.1, Les contrôleurs de disque dur : IDE, SATA \(AHCI\), SCSI, SAS](#), page [85](#) pour des informations supplémentaires.

VirtualBox fournit aussi un **contrôleur amovible** spécial : vous ne pouvez pas y ajouter d'autres périphériques que des lecteurs amovibles. Les périphériques amovibles virtuels comme les lecteurs CD/DVD virtuels peuvent être connectés soit à un lecteur amovible hôte (si vous en avez un), soit associés à une image de disque qui doit alors être au format RAW.

Vous pouvez modifier librement ces connexions de médias. Par exemple, si vous copier des fichiers depuis un autre disque dur virtuel que vous avez créé, vous connecter ce disque comme deuxième disque dur, comme dans l'impression d'écran ci-dessus. Vous pourriez aussi ajouter un deuxième lecteur CD/DVD virtuel ou modifier l'endroit où ces éléments sont branchés. Les options suivantes sont disponibles :

- Pour **ajouter un autre disque dur virtuel ou un lecteur CD/DVD ou de disquette**, sélectionnez un contrôleur de stockage auquel l'ajouter (IDE, SATA, SCSI, SAS, contrôleur amovible), puis cliquez sur le bouton "ajouter un disque" sous l'arborescence. Vous pouvez alors sélectionner soit "Ajouter un périphérique CD/DVD" soit "Ajouter un disque dur". (Si vous avez cliqué sur un contrôleur amovible, vous pouvez plutôt ajouter un lecteur amovible.) Sinon, effectuez un clic droit sur le contrôleur de stockage et sélectionnez-y un élément de menu.

Sur la partie droite de la fenêtre, vous pouvez alors définir ce qui suit :

1. Vous pouvez sélectionner à quel **slot de périphérique** du contrôleur devrait se connecter le disque virtuel. Les contrôleurs IDE ont quatre slots qu'on appelle traditionnellement le "maître primaire", "esclave primaire", "maître secondaire" et "esclave secondaire". Au contraire, les contrôleurs SATA et SCSI vous offrent jusqu'à 30 slots pour connecter des périphériques virtuels.
2. Vous pouvez sélectionner le **fichier image** à utiliser.
 - Pour les disques durs virtuels, un ascenseur en liste déroulante apparaît à droite, vous offrant la sélection soit de **fichier de disque dur virtuel** en utilisant une boîte de dialogue fichier standard soit de **créer un nouveau disque dur** (fichier image), qui ouvrira l'assistant "Créer un nouveau disque", décrite au chapitre [1.7, Créez votre première machine virtuelle](#), page [19](#).

Pour des détails sur les types de fichiers images supportés, merci de voir chapitre [5.2, Fichiers images de disque \(VDI, VMDK, VHD, HDD\)](#), page [88](#).

- Pour les lecteurs CD/DVD virtuels, les fichiers image seront en général au format standard ISO. La plupart du temps, vous sélectionnerez cette option quand vous installerez un système d'exploitation à partir d'une image ISO que vous avez récupéré sur Internet. Par exemple, la plupart des distributions Linux sont disponibles par ce biais.

Pour les lecteurs CD/DVD virtuels, les options suivantes sont disponibles :

- * Si vous sélectionnez "**Lecteur hôte**" dans la liste, le périphérique physique de l'ordinateur hôte sera connecté à la VM, pour que le système d'exploitation

invité puisse lire et écrire sur votre périphérique physique. Cela sert, par exemple, si vous voulez installer Windows depuis un vrai CD d'installation. Dans ce cas, sélectionnez votre lecteur hôte dans la liste déroulante affichée.

Si vous voulez écrire (graver) des CDs ou des DVDs en utilisant le lecteur hôte, vous devez également activer l'option “**Passthrough**” ; voir chapitre [5.9, Support des CD/DVD](#), page [96](#).

- * Si vous sélectionnez “**Supprimer un disque du lecteur virtuel**”, VirtualBox présentera un lecteur CD/DVD vide à l'invité dans lequel on a inséré le média.
- Pour **supprimer une connexion**, sélectionnez-la et cliquez sur l'icône “Supprimer” tout en bas (ou effectuez un clic droit dessus et sélectionnez l'élément du menu).

On peut changer de média amovible (de CD/DVDs et de disquettes) lendant que l'invité fonctionne. Comme la boîte de dialogue “Paramètres” n'est pas disponibles à ce moment-là, vous pouvez aussi accéder à ces paramètres à partir du menu “Périphériques” de la fenêtre de votre machine virtuelle.

3.7 Paramètres de son

La section “Son” de la fenêtre de paramètres d'une machine virtuelle détermine si la VM verra une carte son connectée et si la sortie son se fait en dur sur le système hôte.

Si le son est activé pour un invité, vous pouvez choisir entre l'émulation d'un contrôleur Intel AC'97, un contrôleur Intel HD Audio⁶ ou un e carte SoundBlaster 16. Dans tous les cas, vous pouvez sélectionner le pilote son que VirtualBox utilisera sur l'hôte.

Sur un hôte Linux, selon votre configuration hôte, vous pouvez aussi choisir entre les sous-systèmes OSS, ALSA ou the PulseAudio sur les distributions Linux récentes (Fedora 8 et supérieur, Ubuntu 8.04 et supérieur) le sous-système PulseAudio devrait être privilégié.

3.8 Paramètres réseau

La section “Réseau” de la fenêtre des paramètres d'une machine virtuelle vous permet de configurer la façon dont VirtualBox présente des cartes réseaux virtuelles à votre VM et dont elles agissent.

Quand vous créez une machine virtuelle la première fois, VirtualBox active par défaut une carte réseau virtuelle et sélectionne le mode “Network Address Translation” (NAT) pour celle-ci. De cette manière, l'invité peut se connecter au monde extérieur en utilisant le réseau de l'hôte et le monde extérieur peut se connecter aux services de l'invité que vous choisissez de rendre visibles à l'extérieure de la machine virtuelle.

Ce comportement par défaut convient probablement à 95% des utilisateurs de VirtualBox. Cependant, VirtualBox est extrêmement flexible quant à la manière de virtualiser le réseau. Il supporte de nombreuses cartes réseaux par machine virtuelle, les quatre premières peuvent être configurées en détail dans la fenêtre du gestionnaire. Des cartes réseaux supplémentaires peuvent être configurées en ligne de commande avec VBoxManage.

Du fait de la large gamme d'options disponibles, nous avons consacré un chapitre complet de ce manuel à la configuration réseau ; merci de voir chapitre [6, Le réseau virtuel](#), page [98](#).

⁶Le support d'Intel HD Audio a été ajouté avec VirtualBox 4.0 car Windows 7 (versions 32 bits et 64 bits) ainsi que Windows Vista 64 bits ne supportent pas le contrôleur Intel AC'97.

3.9 Ports série

VirtualBox supporte pleinement les ports série virtuels d'une machine virtuelle d'une manière facile à utiliser.⁷

Jadis, les PC originels d'IBM, les ordinateurs personnels (Personal Computers) étaient équipés d'un ou deux ports série (appelés aussi des ports COM par DOS et Windows). Les ports série sont généralement utilisés avec des modems et certaines souris se connectaient en port série avant que l'USB ne devienne omnipotent.

Si les ports série ne sont plus aussi omniprésents qu'avant, il leur reste encore des cas d'usage importants. Par exemple, on peut utiliser des ports série pour paramétrier un réseau primitif par un câble null-modem, au cas où Ethernet n'est pas disponible. De plus, les ports série sont indispensables pour les programmeurs système ayant besoin de faire du débogage de noyau, car les logiciels de débogage de noyaux interagissent avec les développeurs par le port série. Avec les ports série virtuels, les programmeurs système peuvent faire du débogage de noyau sur une machine virtuelle plutôt qu'un vrai ordinateur où ils se connecteraient.

Si un port série virtuel est activé, le système d'exploitation invité voit un périphérique UART 16550A compatible standard. La réception et la transmission de données est supportée. La manière dont le port série virtuel est alors connecté à l'hôte peut se configurer et les détails dépendent de votre système d'exploitation hôte.

Vous pouvez utiliser soit l'outil graphique, soit VBoxManage en ligne de commande pour paramétrier des ports série virtuels. Pour le dernier cas, merci de vous reporter à chapitre 8.8, [VBoxManage modifyvm](#), page 130 ; dans cette section, cherchez les options `--uart` et `--uartmode`.

Dans le premier cas, vous pouvez configurer jusqu'à deux ports série virtuels par machine virtuelle. Pour chaque périphérique, vous devrez déterminer

1. le type de port série que la machine virtuelle devrait voir en sélectionnant un I/O base address and interrupt (IRQ). Pour ceux-ci, nous vous recommandons d'utiliser les valeurs traditionnelles⁸, qui sont :
 - a) COM1: I/O base 0x3F8, IRQ 4
 - b) COM2: I/O base 0x2F8, IRQ 3
 - c) COM3: I/O base 0x3E8, IRQ 4
 - d) COM4: I/O base 0x2E8, IRQ 3
2. Puis, vous devrez déterminer à quoi ce port série virtuel devrait être connecté. Pour chaque port série virtuel, vous avez les options suivantes :
 - Vous pouvez choisir que le port série virtuel soit "déconnecté", ce qui signifie que l'invité verra le périphérique mais il se comportera comme si aucun câble n'y avait été connecté.
 - Vous pouvez connecter le port série virtuel à un port série existant sur votre hôte. (Sur un hôte Windows, cela sera un nom comme COM1 ; sur des hôtes Linux ou Solaris, ce sera un noeud de périphérique comme `/dev/ttys0`). VirtualBox redirigera alors simplement toutes les données reçues et envoyées sur le port série virtuel vers le périphérique physique.
 - Vous pouvez dire à VirtualBox de connecter le port série virtuel à un tunnel logiciel sur l'hôte. Cela dépend de votre système d'exploitation hôte :
 - Sur un hôte Windows, les données seront envoyées et reçues par un tunnel nommé. Le nom du tunnel doit être au format `\.\pipe\<nom>` où `<name>` devrait identifier la machine virtuelle mais vous êtes libre dans votre choix.

⁷Le support du port série a été ajouté avec VirtualBox 1.5.

⁸Voir, par exemple, [http://en.wikipedia.org/wiki/COM_\(hardware_interface\)](http://en.wikipedia.org/wiki/COM_(hardware_interface)).

3 Configurer des machines virtuelles

Pour rediriger du trafic série, vous pouvez utiliser un programme d'aide appelé VMware Serial Line Gateway, disponible en téléchargement sur <http://www.14ka.org/91.php>. Cet outil fournit un mode serveur fixé nommé tunnel sur \\.\pipe\vmwaredebug et il connecte les connexions TCP entrantes sur le port 567 avec le tunnel nommé.

- Sur un hôte Mac, Linux ou Solaris, une socket locale est plutôt utilisée. Le nom de fichier de la socket doit être choisi de telle sorte que l'utilisateur de VirtualBox ait assez de droits pour créer et écrire dessus. Le répertoire /tmp est souvent un bon candidat.

Sur Linux, plusieurs outils peuvent se connecter à une socket de domaine local ou en créer une en mode serveur. L'outil le plus flexible est `socat` et il est disponible dans beaucoup de distributions.

Dans ce cas, vous pouvez configurer si VirtualBox devrait créer le tunnel nommé (ou, sur les hôtes non Windows, la socket de domaine local) lui-même ou si VirtualBox devrait supposer que le tunnel (ou la socket) existe déjà. Avec les options en ligne de commande de `VBoxManage`, ceci est désigné respectivement comme le mode "serveur" ou "client".

Pour une connexion directe entre deux machines virtuelles, (ce qui correspond à un câble null-modem), configurez simplement une VM pour créer un tunnel/socket et un autre pour s'y relier.

- Vous pouvez envoyer la sortie du port série virtuel vers un fichier. Cette option est très utile pour récupérer des sorties de diagnostic sur un invité. Vous pouvez utiliser n'importe quel fichier dans ce but, tant que l'utilisateur de VirtualBox a assez de droits pour créer et écrire dans le fichier.

Vous pouvez configurer jusqu'à deux ports série par machine virtuelle, mais vous pouvez choisir n'importe quel numéro de port hors de ceux ci-dessus. Cependant, les ports série ne sont pas capables de partager de manière fiable des interruptions ; si deux ports doivent être utilisés en même temps, ils doivent utiliser différents niveaux d'interruption, par exemple COM1 et COM2, mais pas COM1 et COM3.

3.10 Support USB

3.10.1 Paramètres USB

La section "USB" de la fenêtre de paramètres d'une machine virtuelle vous permet de configurer le support USB sophistiqué de VirtualBox.

VirtualBox peut permettre à des machines virtuelles d'accéder aux périphériques USB directement sur votre hôte. Pour cela, VirtualBox présente au système d'exploitation invité un contrôleur USB virtuel. Dès que le système invité démarre en utilisant un périphérique USB, il apparaîtra comme indisponible sur l'hôte.

Note:

1. Faites attention avec les périphériques USB utilisés sur l'hôte ! Par exemple, si vous permettez à votre invité de se connecter à votre disque dur USB actuellement monté sur l'hôte, lorsque l'invité est actif, il sera déconnecté de l'hôte sans débranchement propre. Cela peut entraîner une perte de données.
2. Les hôtes Solaris ont quelques limites connues avec le support USB ; merci de voir chapitre 14, *Limites connues*, page 231.

Au-delà de permettre un accès de l'invité à vos périphériques USB locaux, VirtualBox permet même à vos invités de se connecter à des périphériques USB distants en utilisant le VirtualBox Remote Desktop Extension (VRDE). Pour des détails sur cela, voir chapitre [7.1.4, *USB distant*](#), page [113](#).

Dans la boîte de dialogue des paramètres, vous pouvez d'abord configurer si l'USB est disponible dans l'invité et éventuellement activer le contrôleur USB 2.0 (EHCI) pour l'invité. Si tel est le cas, vous pouvez déterminer en détail les périphériques disponibles. Pour ce faire, vous devez créer ce qu'on appelle des "filtres" en spécifiant certaines propriétés du périphérique USB.

Note: Le contrôleur EHCI est inclus dans une extension de VirtualBox qu'il faut installer séparément. Voir chapitre [1.5, *Installer VirtualBox et les packs d'extension*](#), page [17](#) pour plus d'informations.

Un clic sur le bouton "+" à droite de la fenêtre "Filtres des périphériques USB" crée un **nouveau filtre**. Vous pouvez donner au filtre un nom (pour le retrouver plus tard) et spécifier les critères du filtre. Plus vous spécifiez ces critères, plus les périphériques seront sélectionnés avec précision. Par exemple, si vous ne spécifiez qu'un ID de fabricant 046d, tous les périphériques fabriqués par Logitech seront disponibles pour l'invité. Si vous complétez tous les champs, le filtre ne s'appliquera plutôt qu'à un modèle de périphérique particulier d'un fabricant particulier et pas aux autres périphériques du même type ayant un autre numéro de série ou de révision.

Dans le détail, les critères suivants sont disponibles:

1. **ID du fabricant et du produit.** Avec l'USB, chaque fabricant de produits USB a un numéro d'identification unique au monde, l'"ID fabricant". De la même façon, chaque ligne de produits se voit attribué un numéro "ID de produit". Les deux numéros sont écrits en général en hexadécimal (c'est-à-dire qu'ils se composent des chiffres 0 à 9 et des lettres A à F), et deux-points séparent l'ID du fabricant et du produit. Par exemple, 046d:c016 signifie le fabricant Logitech et le produit Souris optique à roulette M-UV69a".

Sinon, vous pouvez aussi spécifier un nom de "**fabriquant**" et de "**Produit**".

Pour lister tous les périphériques USB connectés à votre machine hôte avec leurs IDs de fabricant et de produit respectifs, vous pouvez utiliser la commande suivante (voir chapitre [8, *VBoxManage*](#), page [119](#)):

```
VBoxManage list usbhost
```

Sur Windows, vous pouvez aussi voir tous les périphériques USB connectés à votre système dans le gestionnaire de périphériques. Sur Linux, vous pouvez utiliser la commande `lsusb`.

2. **Numéro de série.** Si l'ID du fabricant et du produit sont déjà très spécifiques pour identifier des périphériques USB, si vous avez deux périphériques identiques de la même gamme et de la même ligne de produits, vous aurez aussi besoin de leur numéro de série pour les filtrer correctement.
3. **Distant.** Ce paramètre spécifie si le périphérique est seulement en local, distant (par VRDP), ou autrement.

Sur un hôte Windows, vous devrez débrancher et reconnecter un périphérique USB pour l'utiliser après avoir créé un filtre pour lui.

Par exemple, vous pourriez créer un nouveau filtre USB et spécifier un ID fabricant de 046d (Logitech, Inc), à un index de fabricant à 1, et "non distant". Tous les périphériques USB de l'hôte fabriqués par Logitech, Inc ayant un numéro d'index 1 seront visibles pour le système invité.

Plusieurs filtres peuvent sélectionner un périphérique unique – par exemple, un filtre qui sélectionne tous les périphériques Logitech et un qui sélectionne une webcam en particulier.

Vous pouvez **désactiver** des filtres sans les supprimer en cliquant dans la case à cocher à côté du nom du filtre.

3.10.2 Notes d'implémentation pour les hôtes Windows et Linux

Sur les hôtes Windows, un pilote de périphérique en mode noyau fournit un support proxy USB. Il implémente un moniteur USB, qui permet à VirtualBox de capturer des périphériques quand ils sont branchés et un pilote de périphérique USB qui amène les périphériques USB à une machine virtuelle VirtualBox antérieures à 1.4.0, un redémarrage du système n'est plus nécessaire après l'installation du pilote. De plus, vous n'avez plus besoin de rebrancher des périphériques pour que VirtualBox les gère.

Sur les hôtes Linux récents, VirtualBox accède aux périphériques USB par des fichiers spéciaux du système de fichiers. Quand VirtualBox est installé, ils sont rendus disponibles pour tous les utilisateurs dans le groupe système `vboxusers`. Pour pouvoir accéder à l'USB à partir de systèmes invités, assurez-vous d'être membre de ce groupe.

Sur les anciens hôtes Linux, on accède aux périphériques USB en utilisant le système de fichiers `usbfs`. Donc, l'utilisateur qui exécute VirtualBox a besoin des droits en lecture et écriture sur le système de fichiers USB. La plupart des distributions fournissent un groupe (comme `usbusers`) où doit être ajouté l'utilisateur VirtualBox. En outre, VirtualBox ne peut pas faire un proxy avec les périphériques USB de la machine virtuelle qui ne sont pas gérés par un pilote USB de l'hôte Linux. L'entrée `Driver=` de `/proc/bus/usb/devices` vous montrera les périphériques actuellement reconnus. Merci de vous reporter aussi à chapitre [12.7.7, L'USB ne fonctionne pas](#), page [223](#) pour des détails sur `usbfs`.

3.11 Dossiers partagés

Les dossiers partagés vous permettent d'échanger facilement des données entre une machine virtuelle et votre hôte. Cette fonctionnalité exige que les suppléments invité de VirtualBox soient installés dans une machine virtuelle et ceci est décrit en détail au chapitre [4.3, Dossiers partagés](#), page [75](#).

3.12 Autre firmware (EFI)

À partir de la version 3.1, VirtualBox inclut un support expérimental pour l'Extensible Firmware Interface (EFI), qui est un nouveau standard industriel conçu pour remplacer éventuellement, à terme, le BIOS traditionnel comme interface pour les ordinateurs faisant du bootstrapping et certains services système.

Par défaut, VirtualBox utilise le firmware BIOS pour les machines virtuelles. Pour utiliser l'EFI pour une machine virtuelle donnée, vous pouvez activer l'EFI dans la boîte de dialogue "Paramètres" de la machine (voir chapitre [3.4.1, Onglet "Carte mère"](#), page [52](#)). Sinon, utilisez l'interface en ligne de commande `VBoxManage` comme ceci :

```
VBoxManage modifyvm "nom VM" --firmware efi
```

Pour revenir à l'utilisation du BIOS, utilisez :

```
VBoxManage modifyvm "nom VM" --firmware bios
```

Un utilisateur notable de l'EFI est Mac OS X d'Apple, mais les Linux (tels que Fedora 11) et Windows récents (à partir de Vista) offrent des versions spéciales qu'on peut démarrer en utilisant l'EFI.

Une autre utilisation possible de l'EFI dans VirtualBox est le développement et le test d'applications EFI, sans démarrer d'OS.

Remarquez que le support EFI de VirtualBox est expérimental et il sera amélioré au fur et à mesure des progrès d'EFI et de son extension. Mac OS X et Linux sont connus pour très bien fonctionner, les invités Windows sont actuellement incapables de démarrer avec l'implémentation EFI de VirtualBox.

3.12.1 Modes graphiques dans EFI

EFI fournit deux interfaces graphiques distinctes : GOP (Graphics Output Protocol) et UGA (Universal Graphics Adapter). Mac OS X utilise GOP, tandis que Linux a tendance à utiliser UGA. VirtualBox fournit une option de configuration pour contrôler la taille du framebuffer pour les deux interfaces.

Pour contrôler GOP, utilisez la commande VBoxManage suivante :

```
VBoxManage setextradata "nom VM" VBoxInternal2/EfiGopMode N
```

Où N peut être 0,1,2,3,4, respectivement pour des résolutions d'écran 640x480, 800x600, 1024x768, 1280x1024, 1440x900.

Pour modifier la résolution UGA, :

```
VBoxManage setextradata "nom VM" VBoxInternal2/UgaHorizontalResolution 1440  
VBoxManage setextradata "nom VM" VBoxInternal2/UgaVerticalResolution 900
```

Le mode graphique pour GOP et UGA ne peut être modifié que quand la VM est éteinte il reste permandot jusqu'à ce qu'il soit modifié.

4 Les suppléments invité

Le chapitre précédent traitait de la manière de commencer avec VirtualBox et d'installer des systèmes d'exploitation dans une machine virtuelle. Pour une utilisation interactive et sérieuse, les suppléments invité de VirtualBox vous faciliteront beaucoup la vie, en offrant une intégration approfondie entre l'hôte et l'invité et en améliorant la performance d'interactivité des systèmes invités. Ce chapitre décrit en détail les suppléments invité.

4.1 Introduction

Comme indiqué au chapitre [1.2, *Un peu de terminologie*](#), page [12](#), les suppléments invité sont conçus pour s'installer *à l'intérieur* d'une machine virtuelle après qu'un système d'exploitation a été installé. Il s'agit de pilotes de périphériques et d'applications système qui optimisent le système d'exploitation invité pour une meilleure performance et plus d'utilisabilité. Merci de voir chapitre [3.1, *Systèmes d'exploitation invités supportés*](#), page [48](#) pour des détails sur les systèmes d'exploitation entièrement supportés avec les suppléments invité par VirtualBox.

Les suppléments invité de VirtualBox pour tous les systèmes d'exploitation invités supportés sont fournis sous forme d'un fichier unique d'image de CD-ROM qui s'appelle `VBoxGuestAdditions.iso`. Ce fichier image se trouve dans le répertoire d'installation de VirtualBox. Pour installer les suppléments invité sur une VM en particulier, vous montez ce fichier ISO dans votre VM comme un CD-ROM virtuel et vous installez à partir de là.

Les suppléments invité offrent les fonctions suivantes :

Intégration du pointeur de souris Pour dépasser la limite du support de la souris décrite au chapitre [1.8.2, *Capturer et rendre le clavier et la souris*](#), page [23](#), ceci vous offre un support de la souris transparent. Vous n'aurez qu'un pointeur de souris et l'appui sur la touche hôte n'est plus nécessaire pour "libérer" la souris de sa capture par l'OS invité. Pour que cela fonctionne, un pilote de souris spécial est installé dans l'invité pour communiquer avec le pilote de la "vraie" souris de votre hôte et il déplace le pointeur de la souris de l'invité en conséquence.

Dossiers partagés Ceci fournit une manière facile d'échanger des fichiers entre l'hôte et l'invité. Tout comme le voisinage réseau Windows ordinaire, vous pouvez dire à VirtualBox de traiter un répertoire particulier de l'hôte comme un dossier partagé et VirtualBox le rendra disponible pour le système d'exploitation invité en tant que partage réseau, que l'invité ait ou non un réseau. Pour les détails, merci de vous reporter à chapitre [4.3, *Dossiers partagés*](#), page [75](#).

Meilleur support graphique Si la carte graphique virtuelle émulée par VirtualBox pour tous les systèmes d'exploitation invités offre toutes les fonctions de base, les pilotes graphiques personnalisés installés avec les suppléments invité vous offrent ces modes graphiques supplémentaires élevés et non standards ainsi qu'une performance graphique accélérée.

De plus, avec les invités Windows, Linux et Solaris, vous pouvez redimensionner la fenêtre de la machine virtuelle si les suppléments invité sont installés. La résolution graphique de l'invité sera automatiquement ajustée (comme si vous aviez entré à la main une résolution de votre choix dans les paramètres d'affichage de l'invité). Merci de voir aussi chapitre [1.8.5, *Redimensionner la fenêtre de la machine*](#), page [25](#).

Enfin, si les suppléments invité sont installés, les graphismes 3D et 2D des applications invités peuvent être accélérés ; voir chapitre [4.4, L'accélération graphique matérielle](#), page [77](#).

Fenêtres transparentes Avec cette fonctionnalité, les fenêtres individuelles affichées sur le bureau de la machine virtuelle peuvent se placer sur le bureau de l'hôte comme si l'application à leur origine fonctionnait vraiment sur l'hôte. Voir chapitre [4.5, Fenêtres transparentes](#), page [79](#) pour les détails.

Canaux de communication hôte/invité génériques Les suppléments invité vous permettent de contrôler et de surveiller l'exécution de l'invité autrement que comme indiqué ci-dessus. Ce qu'on appelle les "propriétés invité" fournit un mécanisme générique à base de chaînes pour échanger des bits de données entre un invité et un hôte, certains d'entre eux ayant des significations spéciales pour contrôler et surveiller l'invité ; voir chapitre [4.6, Propriétés invité](#), page [80](#) pour les détails.

En outre, des applications peuvent être démarrées dans l'invité à partir de l'hôte ; voir chapitre [4.7, Contrôle de l'invité](#), page [82](#).

Synchronisation du temps Quand les suppléments invité sont installés, VirtualBox peut s'assurer que l'horloge système de l'invité est mieux synchronisée avec celle de l'hôte.

Pour plusieurs raisons, il se peut que l'horloge de l'invité tourne à un rythme légèrement plus lent que celle de l'hôte. L'hôte pourrait recevoir des mises à jour par NTP et sa propre horloge pourrait ne pas tourner de manière rectiligne. Une VM pourrait aussi être mise en pause, ce qui arrête le cours du temps dans l'invité pendant une durée plus ou moins longue. Quand le temps des horloges séparées entre l'invité et l'hôte ne diffère que légèrement, le service de synchronisation du temps essaie d'ajuster progressivement et doucement l'heure de l'invité, par petites accélérations, soit pour "ratrapper", soit pour "perdre" du temps. Quand la différence est trop importante (par exemple si la VM a été mise en pause pendant des heures puis restaurée d'un état sauvegardé), l'heure de l'invité est modifiée immédiatement sans ajustement progressif.

Les suppléments invité resynchronisent l'heure régulièrement. Voir chapitre [9.15.3, Peaufiner les paramètres de synchronisation du temps des suppléments invité](#), page [184](#) pour savoir comment configurer les paramètres du mécanisme de synchronisation du temps.

Presse-papier partagé Quand les suppléments invité sont installés, le presse-papier du système d'exploitation invité peut être éventuellement partagé avec votre système d'exploitation hôte ; voir chapitre [3.3, Paramètres généraux](#), page [51](#).

Connexions automatiques (passer des droits) Pour les détails, merci de voir chapitre [9.2, Identifications automatiques dans l'invité](#), page [163](#).

Chaque version de VirtualBox, même celles mineures, inclut sa propre version des suppléments invité. Si les interfaces par lesquelles le cœur de VirtualBox communique avec les suppléments invité sont stables pour que les suppléments invité déjà installés dans une VM continuent de fonctionner, quand VirtualBox est mis à jour sur l'hôte, pour de meilleurs résultats, il est recommandé d'avoir des suppléments invité de la même version.

À partir de VirtualBox 3.1, les suppléments invité pour Windows et Linux vérifient donc automatiquement si ils doivent être mis à jour. Si l'hôte fait tourner une version plus récente de VirtualBox que celle des suppléments invité, une notification avec des instructions complémentaires s'affiche dans l'invité.

Pour désactiver cette vérification des mises à jour des suppléments invité dans une machine virtuelle donnée, définissez la valeur de la propriété invité /VirtualBox/GuestAdd/CheckHostVersion à 0 ; voir chapitre [4.6, Propriétés invité](#), page [80](#) pour des détails.

4.2 Installer et maintenir les suppléments invité

Les suppléments invité sont disponibles pour les machines virtuelles faisant fonctionner Windows, Linux, Solaris ou OS/2. Les sections suivantes décrivent en détail les spécificités de chaque variante.

4.2.1 Suppléments invité pour Windows

Les suppléments invité de VirtualBox pour Windows sont conçus pour s'installer dans une machine virtuelle exécutant un système d'exploitation Windows. Les versions suivantes des invités Windows sont supportées :

- Microsoft Windows NT 4.0 (tous les packs service)
- Microsoft Windows 2000 (tous les packs service)
- Microsoft Windows XP (tous les packs service)
- Microsoft Windows Server 2003 (tous les packs service)
- Microsoft Windows Server 2008
- Microsoft Windows Vista (toutes les éditions)
- Microsoft Windows 7 (toutes les éditions)
- Microsoft Windows 8 (toutes les éditions)
- Microsoft Windows Server 2012

4.2.1.1 Installation

Dans le menu “Périphériques” de la barre de menu de la machine virtuelle, VirtualBox a un élément de menu tout prêt nommé “Installer les suppléments invité”, qui monte le fichier ISO des suppléments invité dans votre machine virtuelle. L'invité Windows devrait alors démarrer automatiquement l'installateur des suppléments invité qui installe les suppléments invité dans votre invité Windows. Les autres systèmes d'exploitation invités (ou si le démarrage automatique d'un logiciel sur CD est désactivé) exigent un démarrage manuel de l'installateur.

Note: Pour que l'accélération Direct3D de base fonctionne dans un invité Windows, vous devez installer les suppléments invité en “Mode sans échec”. Cela **ne vaut pas** pour le pilote graphique expérimental WDDM Direct3D disponible pour les invités Vista and Windows 7, voir chapitre 14, *Limites connues*, page 231 pour les détails.^a

^aLe pilote expérimental WDDM a été ajouté avec VirtualBox 4.1.

Si vous préférez monter les suppléments à la main, vous pouvez suivre les étapes suivantes :

1. Démarrez la machine virtuelle dans laquelle vous avez installé Windows.
2. Sélectionnez “Monter un CD/DVD-ROM” dans le menu “Périphériques” de la barre de menus de la machine virtuelle, puis “image CD/DVD-ROM”. Ceci ouvre le gestionnaire de médias virtuels décrit au chapitre 5.3, *Le gestionnaire de médias virtuels*, page 88.
3. Dans le gestionnaire de médias virtuels, appuyez sur le bouton “Ajouter” et parcourez le système de fichiers de votre hôte pour trouver le fichier `VBoxGuestAdditions.iso` :

- Sur un hôte Windows, vous pouvez trouver ce fichier dans le répertoire d'installation de VirtualBox (en général sous C:\Program files\Oracle\VirtualBox).
 - Sur les hôtes Mac OS X, vous pouvez trouver ce fichier dans le groupe de l'application VirtualBox. (Effectuez un clic droit sur l'icône de VirtualBox dans Chercheur et choisissez *Afficher le contenu du paquet*. S'y trouve le fichier, dans le dossier Contents/MacOS.)
 - Sur un hôte Linux, vous pouvez trouver ce fichier dans le dossier additions sous lequel vous avez installé VirtualBox (normalement, /opt/VirtualBox/).
 - Sur les hôtes Solaris, vous pouvez trouver ce fichier dans le dossier additions sous lequel vous avez installé VirtualBox (normalement /opt/VirtualBox).
4. De retour dans le gestionnaire de médias virtuels, sélectionnez ce fichier ISO et appuyez sur le bouton “Sélectionner”. Ceci montera le fichier ISO et le présentera à votre invité Windows comme un CD-ROM.

Sauf si vous avez désactivé la fonction “Exécution automatique” de votre invité Windows, Windows démarrera automatiquement le programme d'installation des suppléments invité de VirtualBox depuis l'ISO Additions. Si vous avez désactivé la fonction d'exécution automatique, choisissez VBoxWindowsAdditions.exe dans le lecteur CD/DVD dans l'invité pour démarrer l'installateur.

L'installateur ajoutera plusieurs pilotes de périphériques à la base de données des pilotes Windows puis appellera l'assistant Nouveau matériel détecté.

Selon votre configuration, il pourrait afficher des avertissements selon lequel les pilotes n'ont pas de signature numérique. Vous devez les confirmer afin de continuer l'installation et d'installer correctement les suppléments.

Après l'installation, redémarrez votre système d'exploitation invité pour activer les suppléments.

4.2.1.2 Mettre à jour les suppléments invité Windows

Les suppléments invité Windows peuvent être mis à jour en lançant de nouveau le programme d'installation comme décrit précédemment. Ceci remplacera les pilotes des suppléments précédents par des versions mises à jour.

Vous pouvez aussi ouvrir le gestionnaire de périphériques de Windows et sélectionner “Mettre à jour le pilote...” pour deux périphériques :

1. l'adaptateur graphique de VirtualBox et
2. le périphérique système VirtualBox.

Pour chacun d'eux, choisissez de fournir votre propre pilote et utilisez “Vous avez un disque” pour orienter l'assistant sur le lecteur de CD-ROM contenant les suppléments invité.

4.2.1.3 Installation sans efforts

Avant d'effectuer une installation sans efforts des suppléments invité de VirtualBox sur un invité Windows, il doit y avoir absolument des Oracle CA (Certificate Authority) installés, pour empêcher des fenêtres d'intervention de l'utilisateur qui contrarieront une installation silencieuse.

Note: Sur certaines versions de Windows comme Windows 2000 et Windows XP les fenêtres d'intervention de l'utilisateur mentionnées ci-dessus s'afficheront toujours, même après avoir importé les certificats d'Oracle.

Depuis VirtualBox 4.2, vous pouvez installer ces certificats CA sur un invité Windows de façon automatisée en utilisant l'outil `VBoxCertUtil.exe` qui se trouve sur le CD d'installation des suppléments invité dans le dossier `cert` :

- Connectez-vous en tant qu'administrateur sur l'invité.
- Montez le .ISO des suppléments invité de VirtualBox.
- Ouvrez une fenêtre de ligne de commande sur l'invité et rendez-vous dans le dossier `cert` du CD des suppléments invité de VirtualBox.
- Faites `VBoxCertUtil add-trusted-publisher oracle-vbox.cer --root oracle-vbox.cer`.

Cela installera les certificats dans le stockage des certificats. Si on installe les mêmes certificats plus d'une fois, une erreur circonstanciée s'affichera.

Avant VirtualBox 4.2, il faut importer les certificats CA d'Oracle de manière plus manuelle en utilisant l'outil `certutil.exe` inclu depuis Windows Vista. Pour des versions de Windows antérieures à Vista, vous devez télécharger et installer `certutil.exe` à la main. Les certificats n'étant pas sur le CD-ROM des suppléments invité de VirtualBox précédant la 4.2, vous devez les extraire préalablement d'un exécutable de VirtualBox signé.

Dans l'exemple suivant, les certificats nécessaires seront extraits de l'installateur des suppléments invité pour Windows sur le CD-ROM :

VeriSign Code Signing CA

- Dans l'explorateur Windows, faites un clic droit sur `VBoxWindowsAdditions- <Architecture>.exe`, cliquez sur "Propriétés"
- Allez sur l'onglet "Signatures numériques", choisissez "Oracle Corporation" et cliquez sur "Détails"
- Dans l'onglet "Général" cliquez sur "Afficher le Certificat"
- Dans l'onglet "Chemin de Certification", sélectionnez "VeriSign Class 3 Public Primary CA"
- Cliquez sur "Afficher le certificat"
- Dans l'onglet "Détails" cliquez sur "Copier vers le fichier..."
- Dans l'assistant qui apparaît, choisissez "DER encoded binary (binaire encodé DER X.509 (.CER)" et enregistrez le fichier du certificat dans un chemin local, puis terminez l'assistant
- Fermez la boîte de dialogue du certificat "Verisign Class 3 Code Signing 2010 CA"

Oracle Corporation

- Dans l'explorateur Windows, effectuez un clic droit sur `VBoxWindowsAdditions- <Architecture>.exe`, cliquez sur "Propriétés"
- Allez sur l'onglet "Signatures numériques", choisissez "Oracle Corporation" et cliquez sur "Détails"
- Dans l'onglet "Général", cliquez sur "Afficher le Certificat"
- Dans l'onglet "Détails" cliquez sur "Copier vers le fichier ..."
- Dans l'assistant qui apparaît, choisissez "DER encoded binary X.509 (.CER)" et enregistrez le fichier du certificat dans un chemin local, terminez l'assistant

- Fermez la boîte du dialogue du certificat “Oracle Corporation”

Après avoir exporté les deux certificats ci-dessus, vous pouvez les importer dans le trousseau des certificats en utilisant l'outil `certutil.exe` :

```
certutil -addstore -f Root "<Chemin du fichier du certificat exporté>"
```

Afin de permettre des installations invité entièrement sans efforts, vous pouvez spécifier un paramètre en ligne de commande au lanceur de l'installation :

```
VBoxWindowsAdditions.exe /S
```

Ceci installe automatiquement les bons fichiers et les bons pilotes pour la plateforme correspondante (32 ou 64 bits).

Pour plus d'options concernant les installations d'invité sans efforts, consultez l'aide en ligne de commande en utilisant la commande :

```
VBoxWindowsAdditions.exe /?
```

4.2.1.4 Extraction manuelle du fichier

Si vous souhaitez installer les fichiers et les pilotes à la main, vous pouvez extraire les fichiers des suppléments invité Windows en tapant :

```
VBoxWindowsAdditions.exe /extract
```

Pour extraire explicitement les suppléments invité Windows pour une autre plateforme que celle où vous êtes (par exemple des fichiers 64 bits sur une plateforme 32 bits), vous devez exécuter l'installateur pour la plateforme adéquate (`VBoxWindowsAdditions-x86.exe` ou `VBoxWindowsAdditions-amd64.exe`) avec le paramètre `/extract`.

4.2.2 Suppléments invité pour Linux

Comme les suppléments invité Windows, les suppléments invité de VirtualBox pour Linux sont un ensemble de pilotes de périphérique et d'applications système qui s'installent dans le système d'exploitation invité.

Les distributions Linux suivantes sont supportées officiellement :

- Fedora à partir de Fedora Core 4;
- Redhat Enterprise Linux à partir de la version 3;
- SUSE et openSUSE Linux à partir de la version 9;
- Ubuntu à partir de la version 5.10.

Beaucoup d'autres distributions sont connues pour fonctionner avec les suppléments invité.

La version du noyau Linux fournie par défaut dans SUSE et openSUSE 10.2, dans Ubuntu 6.10 (toutes les versions) et Ubuntu 6.06 (édition serveur) contient un bogue qui peut le faire planter au démarrage quand il tourne sur une machine virtuelle. Les suppléments invité fonctionnent dans ces distributions.

Remarquez que certaines distributions Linux sont déjà fournies avec tout ou partie des suppléments invité de VirtualBox. Vous pouvez choisir de garder la version des suppléments invité de votre distribution, mais ils sont souvent obsolètes et limités dans leurs fonctionnalités, donc nous recommandons de les remplacer par les suppléments invité fournis avec VirtualBox. L'installateur des suppléments invité de VirtualBox pour Linux essaie de détecter l'installation existante et de les remplacer, mais selon la façon dont votre distribution intègre les suppléments invité, cela peut exiger un peu d'intervention manuelle. Il est fortement recommandé de prendre un instantané de la machine virtuelle avant de remplacer les suppléments invité préinstallés.

4.2.2.1 Installer les suppléments invité pour Linux

The suppléments invité de VirtualBox pour Linux sont fournis sur le même fichier de CD-ROM virtuel que les suppléments invité pour Windows décrits ci-dessus. Ils sont également fournis avec un programme d'installation qui vous guide à travers le processus de paramétrage bien que du fait des différences significatives entre les distributions Linux, l'installation peut être légèrement plus complexe.

L'installation implique généralement les étapes suivantes :

1. Avant d'installer les suppléments invité, vous devrez préparer votre système invité à construire les modules externes du noyau. Ceci fonctionne comme décrit au chapitre [2.3.2, Le module noyau de VirtualBox](#), page [40](#), sauf que cette étape doit être maintenant effectuée dans votre *invité* Linux et non sur un système hôte Linux, comme décrit ici.

De nouveau, comme avec les hôtes Linux, nous recommandons d'utiliser DKMS s'il est disponible pour le système invité. S'il n'est pas installé, utilisez cette commande pour les systèmes Ubuntu/Debian :

```
sudo apt-get install dkms
```

ou, pour les systèmes Fedora :

```
yum install dkms
```

Assurez-vous d'installer DKMS *avant* d'installer les suppléments invité Linux. Si DKMS n'est pas disponible ou pas installé, il faudra recréer à la main les modules noyau de l'invité à chaque fois que le noyau invité sera mis à jour en utilisant la commande

```
/etc/init.d/vboxadd setup
```

en tant qu'administrateur.

2. Insérez le fichier de CD `VBoxGuestAdditions.iso` dans le lecteur CD-ROM virtuel de votre système invité, exactement de la même façon que ce qui est décrit pour un invité Windows dans chapitre [4.2.1.1, Installation](#), page [66](#).

3. Rendez-vous dans le répertoire où est monté votre lecteur CD-ROM et exécutez, en tant qu'administrateur :

```
sh ./VBoxLinuxAdditions.run
```

Pour votre confort, nous fournissons les instructions pas à pas suivantes pour les copies fraîchement installées des versions récentes de les distributions Linux les plus populaires. Après ces étapes préparatoires, vous pouvez exécuter l'installateur des suppléments invité VirtualBox comme décrit ci-dessus.

Ubuntu

1. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez


```
apt-get update
```

 en tant qu'administrateur suivi de


```
apt-get upgrade
```
2. Installez DKMS en utilisant


```
apt-get install dkms
```
3. Redémarrez votre système invité afin d'activer les mises à jour puis poursuivez comme décrit ci-dessus.

Fedora

1. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez
`yum update`
en tant qu'administrateur.
2. Installez DKMS et le compilateur GNU C en utilisant
`yum install dkms`
suivi de
`yum install gcc`
3. Redémarrez votre système invité afin d'activer les mises à jour puis poursuivez comme décrit ci-dessus.

openSUSE

1. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez
`zypper update`
en tant qu'administrateur.
2. Installez l'outil make et le compilateur GNU C en utilisant
`zypper install make gcc`
3. Redémarrez votre système invité afin d'activer les mises à jour.
4. Cherchez le noyau que vous exécutez en utilisant
`uname -a`
Un exemple serait `2.6.31.12-0.2-default` qui renvoie au noyau “par défaut”. Puis installez le bon paquet de développement du noyau. Dans l'exemple ci-dessus, il s'agirait de
`zypper install kernel-default-devel`
5. Assurez-vous que votre noyau actuel (`uname -a`) et les paquets du noyau que vous avez installés (`rpm -qa kernel*`) ont exactement le même numéro de version. Continuez l'installation comme décrit ci-dessus.

SuSE Linux Enterprise Desktop (SLED)

1. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez
`zypper update`
en tant qu'administrateur.
2. Installez le compilateur the GNU C en utilisant
`zypper install gcc`
3. Redémarrez votre système invité afin d'activer les mises à jour.
4. Cherchez le noyau que vous exécutez en utilisant
`uname -a`
Un exemple serait `2.6.27.19-5.1-default` qui renvoie au noyau “par défaut”. Puis installez le bon paquet de développement du noyau. Dans l'exemple ci-dessus, il s'agirait de
`zypper install kernel-syms kernel-source`
5. Assurez-vous que votre noyau actuel (`uname -a`) et les paquets du noyau que vous avez installés (`rpm -qa kernel*`) ont exactement le même numéro de version. Continuez l'installation comme décrit ci-dessus.

Mandrake

1. Mandrake inclut des suppléments invité VirtualBox qui seront remplacés si vous suivez ces étapes.
2. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez
`urpmi --auto-update`
en tant qu'administrateur.
3. Redémarrez le système afin d'activer les mises à jour.
4. Installez DKMS en utilisant
`urpmi dkms`
et assurez-vous de choisir le bon paquet kernel-devel quand l'installateur vous le demande (utilisez `uname -a` pour comparer).

CentOS, Red Hat Enterprise Linux et Oracle Enterprise Linux

1. Pour les versions antérieures à 6, ajoutez `divider=10` aux options de démarrage du noyau dans `/etc/grub.conf` pour réduire la charge du processeur actif.
2. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez
`yum update`
en tant qu'administrateur.
3. Installez le compilateur GNU C et les paquets de développement du noyau en utilisant
`yum install gcc`
suivi de
`yum install kernel-devel`
4. Redémarrez votre système invité afin d'activer les mises à jour puis poursuivez comme décrit ci-dessus.
5. Si Oracle Enterprise Linux ne trouve pas les paquets requis, soit vous devez les installer à partir d'une autre source (comme un DVD), soit utilisez le serveur public Yum d'Oracle qui se trouve sur <http://public-yum.oracle.com>.

Debian

1. Afin de mettre à jour complètement votre système invité, ouvrez un terminal et lancez
`apt-get update`
en tant qu'administrateur suivi de
`apt-get upgrade`
2. Installez l'outil make et le compilateur GNU C en utilisant
`apt-get install make gcc`
3. Redémarrez votre système invité afin d'activer les mises à jour.
4. Déterminez la version exacte de votre noyau en utilisant `uname -a` et installez la bonne version du paquet linux-headers, par exemple en utilisant
`apt-get install linux-headers-2.6.26-2-686`

4.2.2.2 Intégration graphique et de la souris

Dans les invités Linux et Solaris, l'intégration graphique et de la souris de VirtualBox passe par le système X Window. VirtualBox peut utiliser la variante X.Org du système (ou XFree86 version 4.3 qui est identique à la première version de X.Org). Au cours du processus d'installation, le serveur d'affichage X.Org sera paramétré pour utiliser les pilotes graphiques et de souris fournis avec les suppléments invité.

Après l'installation des suppléments invité dans une installation neuve d'une distribution Linux ou d'un système Solaris supportés, (beaucoup de systèmes non supportés fonctionneront correctement aussi), le mode graphique de l'invité changera pour s'adapter à la taille de la fenêtre VirtualBox et de l'hôte lorsqu'elle est redimensionnée. Vous pouvez aussi demander au système invité de passer à une résolution en particulier en envoyant une "suggestion de mode graphique" en utilisant l'outil VBoxManage.

Plusieurs écrans invités sont supportés dans les invités utilisant le serveur X.Org version 1.3 (qui fait partie de la version 7.3 du système X Window version 11) ou une version supérieure. La présentation des écrans invités peut être ajustée au besoin en utilisant les outils fournis avec le système d'exploitation invité.

Si vous voulez mieux comprendre les détails de la manière dont les pilotes X.Org sont paramétrés (en particulier si vous souhaitez les utiliser dans un réglage non géré correctement par notre installateur), vous devriez lire chapitre [9.4.2, Paramétrage approfondi des pilotes graphique et souris de l'invité](#), page [169](#).

4.2.2.3 Mettre à jour les suppléments invité Linux

Les suppléments invité peuvent être mis à jour simplement en refaisant la procédure d'installation avec une image de CD-ROM mise à jour. Ceci remplacera les pilotes par des versions mises à jour. Vous devriez redémarrer après avoir mis à jour les suppléments invité.

4.2.2.4 Désinstaller les suppléments invité Linux

Si vous avez une version des suppléments invité installée sur votre machine virtuelle et si vous souhaitez l'enlever sans en installer d'autres, vous pouvez le faire en insérant l'image CD des suppléments invité dans le lecteur de CD-ROM virtuel comme décrit ci-dessus et en lançant l'installateur des suppléments invité actuels avec le paramètre "uninstall" à partir de l'endroit où est monté l'image du CD sur l'invité :

```
sh ./VBoxLinuxAdditions.run uninstall
```

Si cela fonctionnera en principe sans problème, il se peut que vous deviez faire certains nettoyages à la main sur l'invité (en particulier du fichier XFree86Config ou xorg.conf) dans certains cas, surtout si la version installée des suppléments ou le système d'exploitation invité étaient très anciens ou si vous avez fait vos propres modifications du paramétrage des suppléments invité après les avoir installé.

À partir de la version 3.1.0, vous pouvez désinstaller les suppléments en appelant

```
/opt/VBoxGuestAdditions-4.2.51/uninstall.sh
```

Merci de remplacer /opt/VBoxGuestAdditions-4.2.51 par le bon répertoire d'installation des suppléments invité.

4.2.3 Suppléments invité pour Solaris

Comme les suppléments invité pour Windows, les suppléments invité de VirtualBox pour Solaris incluent un ensemble de pilotes de périphériques et d'applications système qui peuvent s'installer dans le système d'exploitation invité.

Les distributions Solaris suivantes sont officiellement supportées :

- Solaris 11 y compris Solaris 11 Express;
- Solaris 10 (u5 et supérieur);

Il se peut que d'autres distributions fonctionnent si elles se basent sur des versions des logiciels comparables.

4.2.3.1 Installer les suppléments invité Solaris

Les suppléments invité de VirtualBox pour Solaris sous fournis sur le même ISO de CD-ROM que les suppléments pour Windows et Linux décrits ci-dessus. Ils sont aussi fournis avec un programme d'installation qui vous guide à travers le processus d'initialisation.

L'installation implique les étapes suivantes :

1. Monter le fichier `VBoxGuestAdditions.iso` comme votre lecteur de CD-ROM virtuel de votre invité Solaris, exactement comme expliqué pour un invité Windows au chapitre [4.2.1.1, Installation](#), page [66](#).

Au cas où le lecteur de CD-ROM de l'invité ne se monte pas (ce qu'on a constaté avec certaines versions de Solaris 10), exécutez en tant qu'administrateur :

```
svcadm restart volfs
```

2. Rendez-vous dans le répertoire où votre lecteur de CD-ROM est monté et exécutez en tant qu'administrateur :

```
pkgadd -G -d ./VBoxSolarisAdditions.pkg
```

3. Choisissez “1” et confirmez l'installation du paquet des suppléments invité. Après la fin de l'installation, reconnectez-vous au serveur X de votre invité pour activer suppléments invité de X11.

4.2.3.2 Désinstaller les suppléments invité Solaris

Les suppléments invité Solaris peuvent être supprimés en toute sécurité en retirant de l'invité le paquet. Ouvrez une session de terminal administrateur et exécutez :

```
pkgrm SUNWvboxguest
```

4.2.3.3 Mettre à jour les suppléments invité

Les suppléments invité devraient être mis à jour en désinstallant d'abord les suppléments invité existants puis en installant les nouveaux. Essayer d'installer de nouveaux suppléments invité sans supprimer ceux existant n'est pas possible.

4.2.4 Suppléments invité pour OS/2

VirtualBox inclut aussi un jeu de pilotes qui améliore l'exécution d'OS/2 dans une machine virtuelle. À cause de restrictions d'OS/2 lui-même, cette variante des suppléments invité a un jeu de fonctionnalités limité ; voir chapitre [14, Limites connues](#), page [231](#) pour les détails.

Les suppléments invité OS/2 sont fournis sur la même ISO de CD-ROM que ceux des autres plateformes. Donc, montez l'ISO dans OS/2 comme décrit précédemment. Les suppléments invité OS/2 se trouvent dans directory `\32bit\OS2`.

Comme on ne fournit pas pour le moment d'installateur automatique, merci de vous reporter au fichier `readme.txt` de ce répertoire, qui décrit la façon d'installer les suppléments invité OS/2 à la main.

4.3 Dossiers partagés

Avec la fonction “dossiers partagés”, de VirtualBox, vous pouvez accéder à des fichiers de votre système hôte depuis votre système invité. Cela est identique à utiliser des partages réseau dans des réseaux Windows – sauf que les dossiers partagés ne nécessitent pas de réseau mais seulement les suppléments invité. Les dossiers partagés sont supportés avec les invités Windows (2000 ou plus récent), Linux et Solaris.

Les dossiers partagés doivent être présents physiquement sur *l'hôte* et sont alors partagés avec l'invité qui utilise un pilote de système de fichiers spécial des suppléments invité pour dialoguer avec l'hôte. Pour les invités Windows, les dossiers partagés sont implémentés comme un redirecteur pseudo-réseau ; pour les invités Linux et Solaris, les suppléments invité fournissent un système de fichiers virtuel.

Pour partager un dossier hôte avec une machine virtuelle de VirtualBox, vous indiquer le chemin vers ce dossier et choisir pour lui un “nom partagé” que l'invité peut utiliser pour y accéder. D'où: créez tout d'abord le dossier partagé sur l'hôte, puis dans l'invité, connectez-vous-y.

Il existe plusieurs façons de paramétrier des dossiers partagés pour une machine virtuelle en particulier :

- Dans la fenêtre d'une VM en fonction, vous pouvez sélectionner “Dossiers partagés” du menu “Périphériques” ou cliquer sur l'icône de dossier dans la barre d'état dans le coin en bas à droite.
- Si une VM est en fonction, vous pouvez configurer des dossiers partagés dans chaque boîte de dialogue “Paramètres” d'une machine virtuelle.
- En ligne de commande, vous pouvez créer des dossiers partagés en utilisant VBoxManage, comme suit :

```
VBoxManage sharedfolder add "nom VM" --name "sharename" --hostpath "C:\test"
```

Voir chapitre [8.29, VBoxManage sharedfolder add/remove](#), page [151](#) pour les détails.

Il existe deux types de partages :

1. Les partages de VM disponibles seulement pour la VM pour laquelle ils ont été définis ;
2. Partages inter-VM, qui peuvent être ajoutés et supprimés pendant l'exécution et qui ne durent pas après l'arrêt d'une VM ; ajoutez pour eux l'option `--transient` à la ligne de commande ci-dessus.

Les dossiers partagés donnent par défaut un accès en lecture/écriture aux fichiers de l'emplacement de l'hôte. Pour restreindre l'invité à un accès en lecture seule, créez un dossier partagé en lecture seule. Cela peut se faire soit en interface graphique, soit mettant l'option `--readonly` lors de la création du dossier partagé avec VBoxManage.

À partir de la version 4.0, les dossiers partagés de VirtualBox supportent aussi les liens symboliques (**symlinks**), dans les conditions suivantes :

1. Le système d'exploitation hôte doit supporter les liens symboliques (donc, un hôte Mac, Linux ou Solaris est requis).
2. Actuellement, seuls les suppléments invité pour Linux et Solaris supportent les liens symboliques.

4.3.1 Montage manuel

Vous pouvez monter des dossiers partagés depuis l'intérieur d'une VM de la même façon que vous monteriez un partage réseau ordinaire :

- Dans un invité Windows, on peut naviguer dans les dossiers partagés qui apparaissent donc dans l'explorateur Windows. Donc, pour connecter un dossier partagé à votre invité Windows, ouvrez l'explorateur Windows et cherchez-le dans "Favoris réseaux" -> "Tout le réseau" -> "Dossier partagé VirtualBox". En effectuant un clic droit sur un dossier partagé et en sélectionnant "Connecter un lecteur réseau" dans le menu qui apparaît, vous pouvez affecter une lettre de lecteur à ce dossier partagé.

Sinon, sur la ligne de commande Windows, utilisez ce qui suit :

```
net use x: \\vboxsvr\sharename
```

Si `vboxsvr` est un nom figé (remarquez que `vboxsrv` fonctionnerait aussi), remplacez "x:" par la lettre de lecteur que vous voulez utiliser pour le partage, et `sharename` par le nom du partage spécifié avec `VBoxManage`.

- Dans un invité Linux, utilisez la commande suivante :

```
mount -t vboxsf [-o OPTIONS] sharename mountpoint
```

Pour monter un dossier partagé au moment du démarrage, ajoutez l'entrée suivante à `/etc/fstab` :

```
sharename  pointmontage  vboxsf  defaults  0  0
```

- Dans un invité Solaris, utilisez la commande suivante :

```
mount -F vboxfs [-o OPTIONS] nompartage mountpoint
```

Remplacez `nompartage` (utilisez des minuscules) par le nom du partage spécifié avec `VBoxManage` ou par l'interface graphique, et `pointmontage` par l'endroit de l'invité où vous voulez monter le partage (par exemple `/mnt/share`). Les règles habituelles de montage s'appliquent, c'est-à-dire créer d'abord un répertoire s'il n'existe pas déjà.

Voici un exemple de montage d'un dossier partagé pour l'utilisateur "jack" sur Solaris :

```
$ id
uid=5000(jack) gid=1(other)
$ mkdir /export/home/jack/mount
$ pfexec mount -F vboxfs -o uid=5000,gid=1 jackshare /export/home/jack/mount
$ cd ~/mount
$ ls
sharedfile1.mp3 sharedfile2.txt
$
```

Par-delà les options standard fournies par la commande `mount`, celles suivantes sont disponibles :

`iocharset CHARSET`

pour définir l'encodage utilisé pour les opérations d'E/S (utf8 par défaut) et

`convertcp CHARSET`

pour définir l'encodage utilisé pour le nom du dossier partagé (utf8 par défaut) et

Les options de montage génériques (documentées dans la page de manuel de `mount`) s'appliquent aussi. Celles particulièrement utiles sont `uid`, `gid` et `mode`, car elles permettent un accès par des utilisateurs ordinaires (en mode lecture/écriture selon les réglages) même si l'administrateur a monté le système de fichiers.

4.3.2 Montage automatique

À partir de la version 4.0, VirtualBox peut monter automatiquement des dossiers partagés avec vos options. Si le montage automatique est activé pour un dossier partagé spécifique, les suppléments invité monteront automatiquement ce dossier dès qu'un utilisateur se connecte à l'OS invité. Les détails dépendent du type d'OS invité :

- Avec des **invités Windows**, tout dossier monté automatiquement aura sa propre lettre de lecteur (comme E:) selon les lettres de lecteur disponibles dans l'invité.

Si aucune lettre de lecteur n'est libre, le montage automatique échouera ; donc le nombre de lecteurs montés automatiquement est limité en général à 22 ou moins avec des invités Windows.

- Avec des **invités Linux**, les dossiers partagés montés automatiquement sont montés dans le répertoire /media, avec le préfixe sf_. Par exemple, le dossier partagé myfiles serait monté dans /media/sf_myfiles sur Linux et dans /mnt/sf_myfiles sur Solaris.

La propriété invité /VirtualBox/GuestAdd/SharedFolders/MountPrefix détermine le préfixe utilisé. Modifiez cette propriété invité en une valeur différente de "sf" pour changer ce préfixe ; voir chapitre [4.6, Propriétés invité](#), page [80](#) pour les détails.

Note: L'accès aux dossiers partagés montés automatiquement n'est autorisé que pour le groupe utilisateur vboxsf, créé par l'installateur des suppléments invité de VirtualBox. Donc les utilisateurs de l'invité doivent être membres de ce groupe pour avoir un accès en lecture/écriture ou en lecture seule si le dossier n'est pas inscriptible.

Pour modifier le répertoire de montage en autre chose que /media, vous pouvez régler la propriété invité /VirtualBox/GuestAdd/SharedFolders/MountDir.

- Les **invités Solaris** se comportent comme les invités Linux, sauf que /mnt est utilisé comme répertoire de montage par défaut au lieu de /media.

Pour effectuer des changements sur les dossiers partagés montés automatiquement pendant qu'une VM est en fonction, l'OS invité doit être relancé. (Cela ne vaut que pour les dossiers partagés montés automatiquement, pas pour ceux montés à la main.)

4.4 L'accélération graphique matérielle

4.4.1 Accélération 3D matérielle (OpenGL and Direct3D 8/9)

Les suppléments invité de VirtualBox contiennent un support 3D matériel expérimental pour les invités Windows, Linux et Solaris.¹

Avec cette fonction, si une application de votre machine virtuelle utilise des fonctions 3D via les interfaces de programmation OpenGL ou Direct3D 8/9, plutôt qu'une émulation logicielle (qui serait lente), VirtualBox essaiera d'utiliser le matériel 3D de votre hôte. Cela fonctionne pour toutes les plateformes hôtes supportées (Windows, Mac, Linux, Solaris), pourvu que votre système d'exploitation hôte puisse utiliser votre accélération 3D matérielle au premier plan.

L'accélération 3D suppose actuellement les conditions suivantes :

1. elle n'est disponible que pour certains invités Windows, Linux et Solaris. En particulier :

¹Le support OpenGL pour les invités Windows a été ajouté avec VirtualBox 2.1 ; le support pour Linux et Solaris a suivi avec VirtualBox 2.2. Avec VirtualBox 3.0, le support Direct3D 8/9 a été ajouté pour les invités Windows. OpenGL 2.0 est maintenant supporté aussi. Avec VirtualBox 4.1, le support du thème Windows Aero a été ajouté pour les invités Windows Vista et Windows 7 (expérimental)

- L'accélération 3D des invités Windows exige Windows 2000, Windows XP, Vista ou Windows 7. OpenGL et Direct3D 8/9 (pas avec Windows 2000) sont supportés (expérimental).
- OpenGL sur Linux exige un noyau 2.6.27 et supérieur et le serveur X.org version 1.5 et supérieur. Ubuntu 10.10 et Fedora 14 ont été testées et confirmées comme opérationnelles.
- OpenGL sur les invités Solaris exige le serveur X.org version 1.5 et supérieur.

2. Les suppléments invité doivent être installés.

Note: Pour l'accélération Direct3D de base fonctionne avec les invités Windows, VirtualBox a besoin de remplacer des fichiers système de Windows dans la machine virtuelle. Donc, le programme d'installation suppléments invité offre l'accélération Direct3D en option que vous devez activer explicitement. De plus, vous devez installer les suppléments invité en mode "Sans échec". Cela **ne s'applique pas** au pilote graphique expérimental WDDM Direct3D disponible pour les invités Vista et Windows 7, voir chapitre 14, *Limites connues*, page 231 pour les détails.

3. Le support 3D étant pour l'instant expérimental, il est désactivé par défaut et il faut **l'activer à la main** dans les paramètres de la VM (voir chapitre 3.3, *Paramètres généraux*, page 51).

Note: Les systèmes invité douteux ne devraient pas être autorisés à utiliser les fonctions d'accélération 3D de VirtualBox tout comme les logiciels douteux de l'hôte ne devraient pas être autorisés à utiliser l'accélération 3D. Les pilotes du matériel 3D sont en général trop complexes pour être parfaitement sécurisés et tout logiciel autorisé à y accéder peut être en mesure de compromettre le système d'exploitation qui les exécute. En outre, l'activation de l'accélération 3D donne à l'invité un accès direct à une grande étendue de code informatique supplémentaire du processus hôte de VirtualBox, donc il pourrait être concevable de l'utiliser pour planter la machine virtuelle.

Avec VirtualBox 4.1, le support du thème Windows Aero est ajouté pour les invités Windows Vista et Windows. Pour activer le support du thème Aero, le pilote graphique expérimental WDDM de VirtualBox doit être installé, il est disponible avec l'installation des suppléments invité. Comme le pilote graphique WDDM est pour l'instant expérimental, il n'est pas installé par défaut et il faut **le sélectionner à la main** dans l'installateur des suppléments invité en répondant "Non" dans la boîte de dialogue "Voulez-vous installer le support Direct3D de base" qui s'affiche quand la fonction Direct3D est sélectionnée.

Note: Contrairement au support Direct3D de base actuel, l'installation du pilote graphique WDDM **n'exige pas** le "Mode sans échec".

Le thème Aero n'est pas activé par défaut. Pour l'activer

- Dans l'invité Windows Vista, effectuez un clic droit sur le bureau, dans le menu contextuel, sélectionnez "Personnaliser" et sélectionnez "Couleur de fenêtre et Apparence" dans la fenêtre "Personnalisation", dans la boîte de dialogue "Paramètres de l'apparence", sélectionnez "Windows Aero" et appuyez sur "OK"

- Dans l'invité Windows 7, effectuez un clic droit sur le bureau, dans le menu contextuel, sélectionnez “Personnaliser” et sélectionnez un thème Aero dans la fenêtre “Personnalisation”

Techniquement, VirtualBox implémente cela en installant un pilote 3D matériel supplémentaire dans votre invité quand les suppléments invité sont installés. Ce pilote agit comme un pilote 3D matériel et signale au système d'exploitation invité que le matériel (virtuel) est capable de faire de l'accélération matérielle 3D. Quand une application de l'invité demande alors l'accélération matérielle via les interfaces de programmation OpenGL ou Direct3D, elles sont envoyées vers l'hôte par un tunnel de communication spécial fourni par VirtualBox, puis l'hôte effectue l'opération 3D demandée via les interfaces de programmation de l'hôte.

4.4.2 L'accélération matérielle 2D pour les invités Windows

À partir de la version 3.1, les suppléments invité de VirtualBox contiennent le support expérimental de l'accélération graphique 2D matériel pour les invités Windows.

Avec cette fonctionnalité, si une application (comme un lecteur de vidéos) de votre VM Windows utilise les couches graphiques 2D pour lire un clip vidéo, VirtualBox essaiera d'utiliser l'accélération graphique matérielle de votre hôte plutôt que de faire de l'interprétation de couches ni conversion de couleurs dans le logiciel (ce qui serait lent). Cela fonctionne actuellement avec les plateformes hôtes Windows, Linux et Mac, pourvu que votre système d'exploitation hôte puisse utiliser l'accélération graphique 2D au premier plan.

L'accélération graphique 2D est possible aujourd'hui sous les conditions suivantes :

1. Elle n'est disponible que pour des invités Windows (XP ou supérieur).
2. Les suppléments invité doivent être installés.
3. Le support 2D étant pour l'instant expérimental, il est désactivé par défaut et il faut **l'activer à la main** dans les paramètres de la VM (voir chapitre [3.3, Paramètres généraux](#), page [51](#)).

Techniquement, VirtualBox implémente ceci en montrant les possibilités DirectDraw de la couche vidéo dans le pilote graphique des suppléments invité. Le pilote envoie toutes les commandes de la couche à l'hôte via un tunnel de communication spécial implémenté par VirtualBox. Côté hôte, OpenGL est alors utilisé pour implémenter la transformation des couleurs et l'échelonage

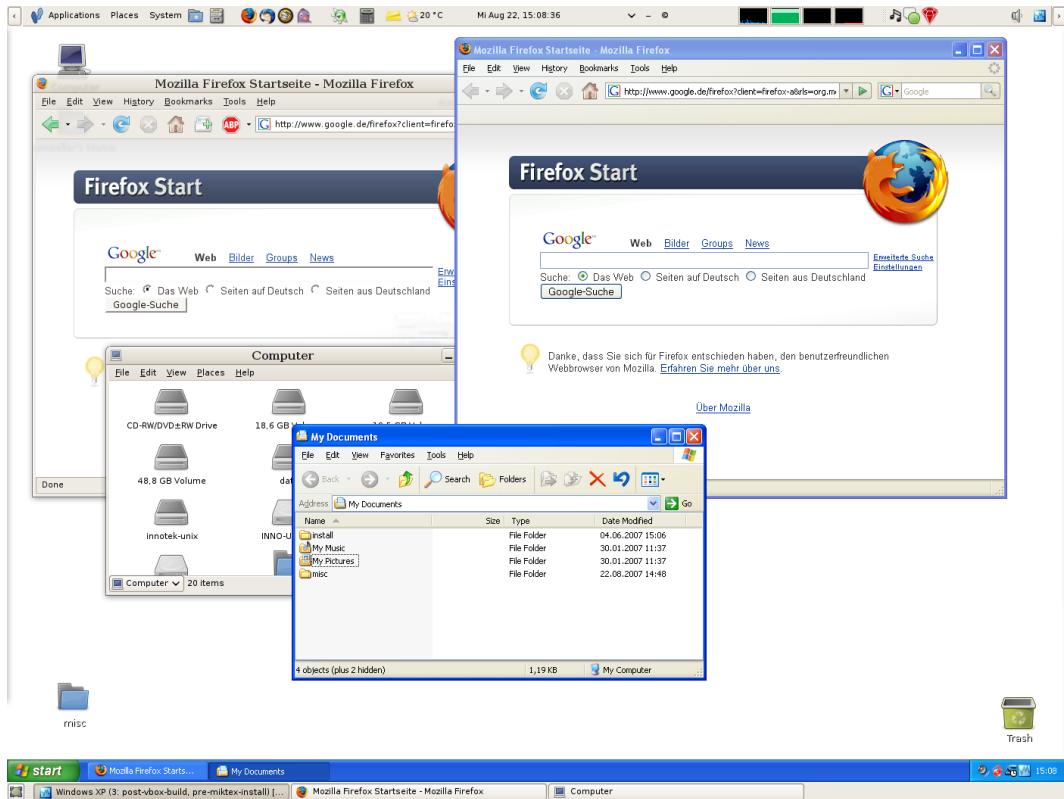
4.5 Fenêtres transparentes

Avec la fonction “fenêtres transparentes”, de VirtualBox, vous pouvez afficher vos fenêtres d'une machine virtuelle côté à côté près des fenêtres de votre hôte. Cette fonctionnalité est supportée pour les systèmes d'exploitation invités suivants (si les suppléments invité sont installés) :

- Les invités Windows (support ajouté à VirtualBox 1.5);
- Les invités supportés Linux ou Solaris exécutant le système X Window (ajouté avec VirtualBox 1.6).

Après que les fenêtres transparentes ont été activées (voir ci-dessous), VirtualBox supprime l'affichage de l'arrière-plan du bureau de votre invité, ce qui vous permet de lancer les fenêtres de votre système d'exploitation invité de manière transparente à côté des fenêtres de votre hôte :

4 Les suppléments invité



Pour activer le mode transparent, après avoir démarré la machine virtuelle, appuyez sur Touche hôte (en principe contrôle droit) et "L" simultanément. Ceci agrandira la taille de l'affichage de la VM jusqu'à la taille de l'écran hôte et masquera l'arrière-plan du système d'exploitation invité. Pour revenir à l'affichage normal de la VM (donc pour désactiver les fenêtres transparentes), appuyez de nouveau sur Touche hôte et "L".

4.6 Propriétés invité

À partir de la version 2.1, VirtualBox permet de demander certaines propriétés depuis un invité existant, si les suppléments invité de VirtualBox sont installés et si la VM est en fonction. C'est intéressant pour deux choses :

1. Un certain nombre de caractéristiques prédéfinies de la VM sont maintenues automatiquement par VirtualBox et peuvent être répercutées sur l'hôte, par exemple, pour gérer les performances de la VM et les statistiques.
2. De plus, des chaînes de données peuvent être échangées entre l'invité et l'hôte. Cela fonctionne dans les deux sens.

Pour faire cela, VirtualBox établit un canal de communication privé entre les suppléments invité de VirtualBox et l'hôte, et les logiciels des deux côtés peuvent utiliser ce canal pour échanger des chaînes de données pour des objectifs de votre choix. Les propriétés invité ne sont que des clés de chaîne auxquelles est rattachée une valeur. Elles peuvent être définies (donc éditées) soit par l'hôte, soit par l'invité, et on peut les lire des deux côtés.

En plus d'établir le mécanisme général de lecture et d'écriture de valeurs, un ensemble de propriétés invité prédéfinies est maintenu automatiquement par les suppléments invité de VirtualBox pour permettre de récupérer des données intéressantes de l'invité telles que le système exact de l'invité et le niveau du service pack, la version installée des suppléments invité, les utilisateurs

4 Les suppléments invité

connectés à l'OS invité, les statistiques réseau et davantage. Ces propriétés prédéfinies ont toutes pour préfixe `/VirtualBox/` et sont organisées en arborescence hiérarchique de clés.

Certaines informations de l'exécution s'affichent quand vous sélectionnez "Boîte de dialogue d'informations sur la session" du menu "Machine" de la machine virtuelle.

Une manière plus flexible d'utiliser ce manuel est de passer par la commande `VBoxManage guestproperty` ; voir chapitre [8.30, *VBoxManage guestproperty*](#), page [151](#) pour les détails. Par exemple, pour voir *toutes* les propriétés invité disponibles pour une Vm donnée en fonction, listées avec leurs valeurs respectives, utilisez ceci :

```
$ VBoxManage guestproperty enumerate "Windows Vista III"
VirtualBox Command Line Management Interface Version 4.2.51
(C) 2005-2013 Oracle Corporation
All rights reserved.

Name: /VirtualBox/GuestInfo/OS/Product, value: Windows Vista Business Edition,
      timestamp: 1229098278843087000, flags:
Name: /VirtualBox/GuestInfo/OS/Release, value: 6.0.6001,
      timestamp: 1229098278950553000, flags:
Name: /VirtualBox/GuestInfo/OS/ServicePack, value: 1,
      timestamp: 1229098279122627000, flags:
Name: /VirtualBox/GuestAdd/InstallDir,
      value: C:/Program Files/Oracle/VirtualBox
      suppléments invité, timestamp: 1229098279269739000, flags:
Name: /VirtualBox/GuestAdd/Revision, value: 40720,
      timestamp: 1229098279345664000, flags:
Name: /VirtualBox/GuestAdd/Version, value: 4.2.51,
      timestamp: 1229098279479515000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxControl.exe, value: 4.2.51r40720,
      timestamp: 1229098279651731000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxHook.dll, value: 4.2.51r40720,
      timestamp: 1229098279804835000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxDisp.dll, value: 4.2.51r40720,
      timestamp: 1229098279880611000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxMRXNP.dll, value: 4.2.51r40720,
      timestamp: 1229098279882618000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxService.exe, value: 4.2.51r40720,
      timestamp: 1229098279883195000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxTray.exe, value: 4.2.51r40720,
      timestamp: 1229098279885027000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxGuest.sys, value: 4.2.51r40720,
      timestamp: 1229098279886838000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxMouse.sys, value: 4.2.51r40720,
      timestamp: 1229098279890600000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxSF.sys, value: 4.2.51r40720,
      timestamp: 1229098279893056000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxVideo.sys, value: 4.2.51r40720,
      timestamp: 1229098279895767000, flags:
Name: /VirtualBox/GuestInfo/OS/LoggedInUsers, value: 1,
      timestamp: 1229099826317660000, flags:
Name: /VirtualBox/GuestInfo/OS/NoLoggedInUsers, value: false,
      timestamp: 1229098455580553000, flags:
Name: /VirtualBox/GuestInfo/Net/Count, value: 1,
      timestamp: 1229099826299785000, flags:
Name: /VirtualBox/HostInfo/GUI/LanguageID, value: C,
      timestamp: 1229098151272771000, flags:
Name: /VirtualBox/GuestInfo/Net/0/V4/IP, value: 192.168.2.102,
      timestamp: 1229099826300088000, flags:
Name: /VirtualBox/GuestInfo/Net/0/V4/Broadcast, value: 255.255.255.255,
      timestamp: 1229099826300220000, flags:
Name: /VirtualBox/GuestInfo/Net/0/V4/Netmask, value: 255.255.255.0,
      timestamp: 1229099826300350000, flags:
Name: /VirtualBox/GuestInfo/Net/0/Status, value: Up,
      timestamp: 1229099826300524000, flags:
Name: /VirtualBox/GuestInfo/OS/LoggedInUsersList, value: username,
      timestamp: 1229099826317386000, flags:
```

To query the value of a single property, use the “get” subcommand like this:

```
$ VBoxManage guestproperty get "Windows Vista III"  
  "/VirtualBox/GuestInfo/OS/Product"  
VirtualBox Command Line Management Interface Version 4.2.51  
(C) 2005-2013 Oracle Corporation  
All rights reserved.  
  
Value: Windows Vista Business Edition
```

Pour ajouter ou modifier des propriétés invité depuis l’invité, utilisez l’outil `VBoxControl`. Cet outil est inclus dans les suppléments invité de VirtualBox 2.2 ou supérieur. S’il est lancé avec un invité Linux, cet outil exige les priviléges administrateur pour des raisons de sécurité :

```
$ sudo VBoxControl guestproperty enumerate  
VirtualBox suppléments invité Command Line Management Interface Version 4.2.51  
(C) 2009-2013 Oracle Corporation  
All rights reserved.  
  
Name: /VirtualBox/GuestInfo/OS/Release, value: 2.6.28-18-generic,  
      timestamp: 1265813265835667000, flags: <NULL>  
Name: /VirtualBox/GuestInfo/OS/Version, value: #59-Ubuntu SMP Thu Jan 28 01:23:03 UTC 2010,  
      timestamp: 1265813265836305000, flags: <NULL>  
      ...
```

Pour des besoins plus complexes, vous pouvez utiliser l’interface de programmation de VirtualBox ; voir chapitre 11, [Interfaces de programmation de VirtualBox](#), page 207.

4.7 Contrôle de l’invité

À partir de la version 3.2, les suppléments invité de VirtualBox permettent de démarrer les applications d’une VM à partir du système hôte.

Pour que cela fonctionne, l’application doit être installée dans l’invité ; pas besoin de logiciels supplémentaires sur l’hôte. De plus, une sortie en mode texte (sur `stdout` et `stderr`) peut s’afficher sur l’hôte pour un traitement ultérieur avec des options pour spécifier les droits des utilisateurs et une valeur de timeout (en millisecondes) pour limiter le temps où l’application peut s’exécuter.

Cette fonctionnalité peut être utilisée pour automatiser le déploiement d’un logiciel dans l’invité.

À partir de la version 4.0, les suppléments invité pour Windows permettent une mise à jour automatique (seulement si les suppléments invité 4.0 ou supérieurs sont déjà installés). En outre, la copie de fichiers de vhôte vers l’invité ainsi que la création à distance de répertoires invités est disponible.

Pour utiliser ces fonctionnalités, utilisez la ligne de commande de VirtualBox, voir see chapitre 8.31, [VBoxManage guestcontrol](#), page 152.

4.8 Transfert de mémoire

Dans des environnements serveur ayant beaucoup de VMs ; les suppléments invité peuvent être utilisés pour partager de la mémoire physique de l’hôte entre plusieurs VMs, réduisant la quantité totale de mémoire utilisée par les VMs. Si l’utilisation de la mémoire est le facteur limitatif et que des ressources processeur sont encore disponibles, cela peut aider à empaqueter davantage de VMs sur chaque hôte.

4.8.1 Faire du ballon avec la mémoire

À partir de la version 3.2, les suppléments invité de VirtualBox peuvent changer la quantité de mémoire hôte utilisée par une VM pendant que la machine est en fonction. Vu comme cette fonction est implémentée, elle s'appelle le “jeu de ballon de mémoire”.

Note: VirtualBox ne supporte le jeu de ballons avec la mémoire que sur les hôtes 64 bits et il n'est pas supporté sur les hôtes Mac OS X.

En principe, pour changer la quantité de mémoire affectée à une machine virtuelle, il faut éteindre la machine virtuelle complètement et modifier ses paramètres. Avec le jeu de ballon de mémoire, la mémoire affectée à une machine virtuelle peut être donnée à une autre machine virtuelle sans devoir arrêter la machine.

Quand on veut faire du ballon avec la mémoire, les suppléments invité de VirtualBox (qui tournent dans l'invité) affectent de la mémoire physique du système d'exploitation invité au niveau noyau et il verrouille cette mémoire dans l'invité. Cela garantit que l'invité n'utilisera plus cette mémoire : aucune application de l'invité ne peut l'affecter, et le noyau invité ne l'utilisera pas non plus. VirtualBox peut alors réutiliser cette mémoire et la donner à une autre machine virtuelle.

La mémoire rendue disponible par ce jeu de ballon n'est pas disponible pour être réutilisée par VirtualBox. Elle *n'est pas* rendue sous forme de mémoire libre à l'hôte. Jouer au ballon avec la mémoire à partir d'un invité en fonction n'augmentera pas la taille de la mémoire libre non affectée de l'hôte. En effet, le jeu de ballon avec la mémoire est donc un mécanisme de transfert de mémoire entre plusieurs machines virtuelles pendant leur fonctionnement. Cela peut servir pour démarrer temporairement une autre machine, ou, dans des environnements plus compliqués, pour une gestion sophistiquée de la mémoire de nombreuses machines virtuelles en fonction en parallèle, selon la façon dont la mémoire est utilisée par les invités.

Pour l'instant, le jeu de ballon avec la mémoire n'est supporté que via VBoxManage. Utilisez la commande suivante pour agrandir ou réduire la taille du ballon de mémoire dans une machine virtuelle en fonction où les suppléments invité sont installés :

```
VBoxManage controlvm "nom VM" guestmemoryballoon <n>
```

où "nom VM" est le nom ou l'UUID de la machine virtuelle en question et <n> est la quantité de mémoire à affecté à partir de l'invité, en mégaoctets. Voir chapitre 8.13, [VBoxManage controlvm](#), page 140 pour plus d'informations.

Vous pouvez aussi définir un ballon par défaut qui sera automatiquement récupéré sur la VM à chaque fois qu'elle démarrera avec la commande suivante :

```
VBoxManage modifyvm "nom VM" --guestmemoryballoon <n>
```

Par défaut, aucun ballon de mémoire n'est affecté. C'est un paramètre de VM comme les autres paramètres de modifyvm et donc, on ne peut les définir que quand la machine est éteinte ; voir chapitre 8.8, [VBoxManage modifyvm](#), page 130.

4.8.2 Fusion de page

Alors que le jeu de ballon avec la mémoire réduit simplement la quantité de RAM disponible pour une VM, la fusion de page fonctionne différemment : elle évite les doublons de mémoire entre plusieurs VMs identiques et en fonction.

Dans un environnement serveur qui exécute plusieurs VMs identiques (comme avec des systèmes d'exploitation identiques) sur le même hôte, beaucoup de pages de mémoire sont identiques. La technologie de fusion de page de VirtualBox, introduite avec VirtualBox 3.2, est une technique de novel pour identifier efficacement ces pages de mémoire identiques et les partager entre plusieurs VMs.

Note: VirtualBox ne supporte la fusion de page que sur des hôtes 64 bits et il n'est pas supporté sur les hôtes Mac OS X. La fusion de page ne fonctionne aujourd'hui qu'avec des invités Windows (2000 et supérieur).

Plus les VMs se ressemblent sur un hôte donné, plus la fusion de page peut réduire efficacement la quantité de mémoire hôte utilisée. Il fonctionne donc mieux si toutes les VMs d'un hôte exécutent des systèmes d'exploitation identiques (par exemple Windows XP Pack Service 2). Plutôt que d'avoir une copie complète de chaque système d'exploitation dans chaque VM, la fusion de page identifie les pages mémoire identiques utilisées par les systèmes d'exploitation et elle élimine les doublons, partageant la mémoire de l'hôte entre plusieurs machines ("deduplication"). Si une VM essaie de modifier une page partagée avec d'autres VMs, une nouvelle page est de nouveau affectée pour cette VM avec une copie de la page partagée ("copie à l'écriture"). Tout ceci est entièrement transparent pour la machine virtuelle.

Il se peut que vous soyez familier de ce genre de transfert de mémire via d'autres produits d'hypervision qui appellent cette fonction le "partage de page" ou le "fusion d'une même page". Cependant, la fusion de page est très différente des autres solutions, dont les approches ont plusieurs inconvénients :

1. Les hyperviseurs traditionnels analysent *toute* la mémoire invité et calculent les sommes de contrôle (hachages) pour chaque page de mémoire individuelle. Puis ils cherchent des endroits avec des hachages identiques et comparent et comparent tout le contenu de ces pages ; si deux pages donnent le même hachage, il est très probable que les pages soient identiques par leur contenu. Cela peut, bien sûr, prendre du temps, surtout si le système n'est pas inactif. La mémoire supplémentaire ne devient donc disponibles qu'après une durée importante (cela peut prendre des heures voire des jours !). Pire, ce genre d'algorithme de partage de page consomme en général beaucoup de ressources et augmente la vitesse de la virtualisation jusqu'à 10-20%.

La fusion de page dans VirtualBox utilise une logique des suppléments invité de VirtualBox pour identifier rapidement les cellules mémoire les plus vraisemblablement identiques dans les VMs. Elle peut faire la plupart des sauvegardes possibles des partages de pages presqu'immédiatement et avec pratiquement aucune charge.

2. La fusion de page a aussi beaucoup moins de chances d'être perdue avec de la mémoire identique qu'elle éliminera, pour apprendre seulement quelques secondes plus tard que la mémoire va maintenant changer, devant effectuer une réaffectation très gourmande et gênant souvent le service.

Pour l'instant, la fusion de page ne peut être contrôlée qu'avec VBoxManage et pendant qu'une VM est éteinte. Pour activer la fusion de page pour une VM, utilisez the following command:

```
VBoxManage modifyvm "nom VM" --pagefusion on
```

Vous pouvez voir l'opération de fusion de page en utilisant des métriques. RAM/VMM/Shared affiche la quantité totale de pages fusionnées, tandis que la métrique par VM Guest/RAM/Usage/Shared renverra la quantité de mémoire fusionnée d'une VM donnée. Merci de vous reporter à chapitre 8.33, *VBoxManage metrics*, page 157 pour des informations sur la manière de demander des métriques.

5 Le stockage virtuel

Comme la machine virtuelle s'attendra, la plupart du temps, à voir un disque dur construit dans son ordinateur virtuel, VirtualBox doit pouvoir présenter du “vrai” stockage à l’invité comme un disque dur virtuel. Il existe aujourd’hui trois méthodes pour faire cela :

1. Le plus souvent, VirtualBox utilisera de gros fichiers images sur un vrai disque dur et les présentera à un invité comme des disques durs virtuels. Cela est décrit au chapitre [5.2, Fichiers images de disque \(VDI, VMDK, VHD, HDD\)](#), page [88](#).
2. Sinon, si vous avez des serveurs de stockage iSCSI, vous pouvez connecter un tel serveur à VirtualBox comme le décrit le chapitre [5.10, Serveurs iSCSI](#), page [97](#).
3. Enfin, comme fonction avancée, vous pouvez permettre à une machine virtuelle d'accéder directement à un des disques durs de votre hôte ; cette fonction avancée est décrite au chapitre [9.8.1, Utiliser un disque dur brut de l'hôte à partir de l'invité](#), page [173](#).

Chaque périphérique virtuel (fichier image, cible iSCSI ou disque physique) devra être connecté au contrôleur de disque dur virtuel présenté par VirtualBox à une machine virtuelle. Ceci est expliqué à la prochaine section.

5.1 Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS

Dans un vrai PC, les disques durs et les lecteurs de CD/DVD sont connectés à un périphérique appelé le contrôleur de disque dur, qui pilote les opérations du disque dur et les transferts de données. VirtualBox peut émuler les quatre types contrôleurs de disque les plus courants qu’on trouve en général dans les PCs d’aujourd’hui : IDE, SATA (AHCI), SCSI et SAS.¹

- Les contrôleurs **IDE (ATA)** sont une extension très avancée de la rétrocompatibilité du contrôleur de disque dur dans les PC/AT IBM (1984). Au début, cette interface ne fonctionnait qu’avec des disques durs mais elle a ensuite été étendue aussi au support des lecteurs CD-ROM et d’autres types de médias amovibles. Dans des PCs physiques, ce standard utilise des nappes de 40 ou 80 fils. Chaque câble peut connecter deux périphériques à un contrôleur, ce qu’on appelait traditionnellement le “maître” et “l’esclave”. Les PCs classiques avaient deux connecteurs par câbles, le support jusqu’à quatre périphériques était donc classique.

Dans VirtualBox, chaque machine virtuelle peut avoir un contrôleur IDE activé, ce qui vous donne jusqu’à quatre périphériques de stockage virtuels connectables à la machine. (Par défaut, un des quatre – le maître secondaire –) est préconfiguré pour être le lecteur CD/DVD de la machine virtuelle, mais ceci peut être modifié.²)

Donc, même si votre système d’exploitation invité ne supporte pas les périphériques SCSI ou SATA, il devrait toujours pouvoir voir un contrôleur IDE.

¹Le support SATA a été ajouté avec VirtualBox 1.6 ; le support SCSI expérimental a été ajouté à la 2.1 et complètement implémenté avec la 2.2. En général, les connexions de stockage ont été rendus beaucoup plus flexibles avec VirtualBox 3.1 ; voir ci-dessous. Le support pour le contrôleur LSI Logic SAS a été ajouté avec VirtualBox 3.2.

²L’affectation du lecteur CD/DVD de la machine au maître secondaire était figé avant VirtualBox 3.1 ; il est maintenant modifiable et le lecteur peut être sur d’autres slots du contrôleur IDE et il peut y avoir plus de deux lecteurs.

Vous pouvez aussi sélectionner le type exact de contrôleur IDE que VirtualBox devrait matériellement présenter à la machine virtuelle (PIIX3, PIIX4 ou ICH6). Il n'y a pas de différence en termes de performance, mais si vous importez une machine virtuelle d'un autre produit de virtualisation, le système d'exploitation de cette machine peut attendre un type de contrôleur particulier et planter s'il n'est pas trouvé.

Après avoir créé une nouvelle machine virtuelle avec l'assistant "Nouvelle Machine virtuelle" de l'interface graphique, vous verrez généralement un contrôleur IDE dans les paramètres de "Stockage" de la machine, auquel le lecteur CD/DVD sera connecté, sur un des quatre ports du contrôleur.

- **Serial ATA (SATA)** est un standard récent introduit en 2003. Par rapport à l'IDE, il supporte plus de périphériques par contrôleur et à bien plus haute vitesse. De plus, avec du matériel physique, les périphériques peuvent être ajoutés et supprimés pendant que le système fonctionne. L'interface standard des contrôleurs SATA est appelé Advanced Host Controller Interface (AHCI).

Comme un vrai contrôleur SATA, le contrôleur SATA virtuel de VirtualBox travaille plus vite et consomme moins de ressources processeur que le contrôleur IDE virtuel. De plus, il vous permet de connecter jusqu'à 30 disques durs virtuels à une machine au lieu de seulement trois comme avec le contrôleur IDE de VirtualBox (avec le lecteur DVD déjà connecté).

Pour cette raison, à partir de la version 3.2 et selon le système d'exploitation invité sélectionné, VirtualBox utilise SATA par défaut pour les machines virtuelles nouvellement créées. Un contrôleur virtuel SATA est créé par défaut et le disque par défaut créé avec une nouvelle VM est connecté à ce contrôleur.

Avertissement: Tout le contrôleur SATA et les disques virtuels qui s'y connectent (y compris ceux en mode de compatibilité IDE) ne seront pas vus par les systèmes d'exploitation n'ayant pas de support physique pour AHCI. En particulier, **il n'y a pas de support pour AHCI dans Windows avant Windows Vista**, donc Windows XP (même SP3) ne verra pas de tels disques sauf si vous installez des pilotes supplémentaires. Il est possible de passer de l'IDE à SATA après l'installation en installant les pilotes SATA et en changeant le type de contrôleur dans la boîte de dialogue des paramètres de la VM.^a

^aVirtualBox recommande les pilotes Intel Matrix Storage qui sont téléchargeables sur http://downloadcenter.intel.com/Product_Filter.aspx?ProductID=2101.

Pour ajouter un contrôleur SATA à une machine pour laquelle il n'a pas été activé par défaut (soit parce qu'elle a été créée par une version antérieure VirtualBox, soit parce que SATA n'est pas supporté par défaut par le système d'exploitation invité sélectionné), allez dans l'onglet "Stockage" de la boîte de dialogue des paramètres de la machine, cliquez sur le bouton "Ajouter un contrôleur" sous la case "Arborescence de stockage" puis sélectionnez "Ajouter un contrôleur SATA". Après quoi, le contrôleur traditionnel apparaîtra comme un périphérique PCI distinct dans la machine virtuelle et vous pouvez y ajouter des disques virtuels.

Pour modifier les paramètres du mode de compatibilité IDE pour le contrôleur SATA, merci de voir chapitre [8.19, *VBoxManage storagectl*](#), page [145](#).

- **SCSI** est un autre standard industriel signifiant "Small Computer System Interface". SCSI a été standardisé dès 1986 comme une interface générique pour transférer des données entre tous les types de périphériques, y compris ceux de stockage. Aujourd'hui, SCSI est encore utilisé pour connecter des disques durs et des lecteurs à bandes, mais il est généralement devenu un matériel de convenance. On l'utilise encore souvent sur des stations de travail et des serveurs à haute performance.

En priorité pour des raisons de compatibilité avec d'autres logiciels de virtualisation, VirtualBox supporte en option les contrôleurs LSI Logic et BusLogic SCSI, chacun d'entre eux pouvant gérer jusqu'à 15 disques durs virtuels.

Pour activer un contrôleur SCSI, sur l'onglet "Stockage" de la boîte de dialogue des paramètres d'une machine virtuelle, cliquez sur le bouton "Ajouter un contrôleur" sous la case "Arborescence de stockage" puis sélectionnez "Ajouter un contrôleur SCSI". Après quoi, le contrôleur supplémentaire apparaîtra comme périphérique PCI distinct dans la machine virtuelle.

Avertissement: Comme avec les autres types de contrôleurs, un contrôleur SCSI ne sera vu que par les systèmes d'exploitation ayant un support de ce périphérique. Windows 2003 et supérieur incluent des pilotes pour le contrôleur LSI Logic, tandis que Windows NT 4.0 et Windows 2000 incluent des pilotes pour le contrôleur BusLogic. Windows XP n'inclut aucun pilote.

- **Serial Attached SCSI (SAS)** est un autre standard de bus qui utilise le jeu de commandes SCSI. Mais contrairement au SCSI, avec les périphériques physiques, on utilise des câbles série au lieu de ceux parallèles, ce qui simplifie la connexion de périphériques physiques. D'une certaine manière, SAS est donc au SCSI ce que SATA est à l'IDE: il permet des connexions plus fiables et plus rapides.

Pour supporter des invités de haut niveau exigeant des contrôleurs SAS, VirtualBox émule un contrôleur a LSI Logic SAS qui peut être activé de la même façon qu'un contrôleur SCSI. Pour le moment, on peut connecter jusqu'à huit périphériques au contrôleur SAS.

Avertissement: Comme avec SATA, le contrôleur SAS ne sera vu que par les systèmes d'exploitation supportant ce périphérique. En particulier, **il n'y a pas de support de SAS dans Windows avant Windows Vista**, donc Windows XP (même SP3) ne verra pas de tels disques, sauf si vous installez des pilotes supplémentaires.

En résumé, VirtualBox vous offre les catégories de slots de stockage virtuels suivantes :

1. quatre slots attachés au contrôleur IDE traditionnel, qui sont toujours présents (un d'eux est en général un lecteur de CD/DVD virtuel) ;
2. 30 slots attachés au contrôleur SATA s'il est activé et si votre système d'exploitation invité le supporte ;
3. 15 slots attachés au contrôleur SCSI s'il est activé et si le système d'exploitation invité peut le voir ;
4. huit slots attachés au contrôleur SAS, s'il est activé et si le système d'exploitation invité peut le voir ;

Étant donné le vaste choix de contrôleurs de stockage, il se peut que vous vous demandiez lequel choisir. En général, vous devriez éviter l'IDE, à moins que ce soit le seul contrôleur supporté par votre invité. Que vous utilisez SATA, SCSI ou SAS ne présente pas de vraie différence. La variété des contrôleurs n'est fournie par VirtualBox que pour des raisons de compatibilité avec le matériel existant et d'autres hyperviseurs.

5.2 Fichiers images de disque (VDI, VMDK, VHD, HDD)

Les fichiers images de disque résident sur le système hôte et sont vus par les systèmes invités comme des disques durs d'une certaine composition. Quand un système d'exploitation invité lit ou écrit sur un disque dur, VirtualBox redirige la requête vers le fichier image.

Comme un disque physique, un disque virtuel est d'une taille (capacité), qui doit être indiquée quand le fichier image est créé. Mais contrairement à un disque physique, VirtualBox vous permet d'étendre un fichier image après l'avoir créé, même s'il contient déjà des données (voir chapitre [8.23, VBoxManage modifyhd](#), page [147](#) pour les détails).³

VirtualBox supporte quatre variantes de fichiers images de disque :

- Normalement, VirtualBox utilise son propre format de container pour les disques durs invités – des fichiers Virtual Disk Image (VDI). En particulier, ce format sera utilisé quand vous créerez une nouvelle machine virtuelle avec un nouveau disque.
- VirtualBox supporte aussi complètement le format de container VMDK ouvert et populaire, utilisé par de nombreux produits de virtualisation, en particulier, par VMware.⁴
- VirtualBox supporte complètement le format VHD utilisé par Microsoft.
- Les fichiers images de Parallels version 2 (format HDD) sont aussi supportés.⁵ Par manque de documentation du format, les nouveaux formats (3 et 4) ne sont pas supportés. Vous pouvez cependant convertir de tels fichiers images à leur version 2 en utilisant des outils fournis par Parallels.

Indépendamment de la capacité et du format du disque, comme on l'a brièvement indiqué au chapitre [1.7, Créer votre première machine virtuelle](#), page [19](#), il y a deux options sur la façon de créer une image de disque : la taille fixe ou dynamique..

- Si vous créez une **image à taille fixe**, un fichier image sera créé sur votre système hôte ayant rigoureusement la même taille que la capacité du disque virtuel. Donc, pour un disque de 10G, vous aurez un fichier de 10G. Remarquez que la création d'une image à taille fixe peut prendre du temps selon la taille de l'image et les performances d'écriture de votre disque dur.
- Pour une gestion du stockage plus flexible, utilisez une **image dynamique**. Elle sera très petite au départ et n'occupera pas de place pour des secteurs de disque virtuel inutilisés, mais elle grandira à chaque fois qu'un secteur de disque sera écrit pour la première fois, jusqu'à ce que le lecteur ne atteigne la capacité maximale choisie quand le lecteur a été créé. Si ce format prend moins de place au départ, le fait que VirtualBox doit étendre le fichier image consomme des ressources de calcul supplémentaires, donc jusqu'à ce que la taille du fichier de disque ait été stabilisée, les opérations d'écriture peuvent être plus lentes qu'avec des disques à taille fixe. Cependant, après un certain temps, la vitesse de grossissement ralentira l'inconvénient moyen des opérations d'écriture deviendra négligeable.

5.3 Le gestionnaire de médias virtuels

VirtualBox garde une trace de toutes les images de disque dur, de lecteur CD/DVD-ROM et de disquette utilisés par les machines virtuelles. On les désigne souvent comme des “médias connus” et ils proviennent de deux sources :

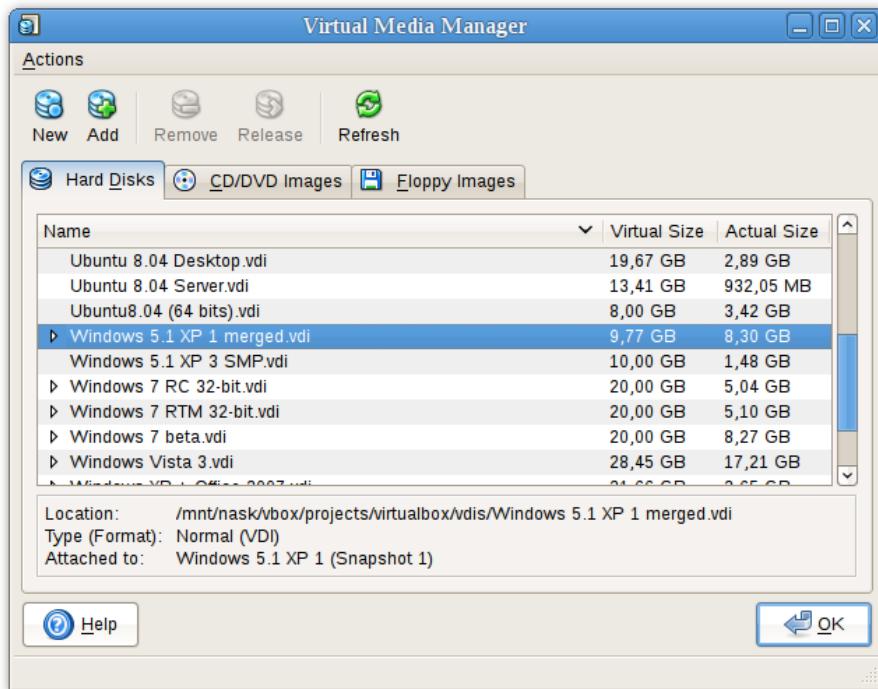
³Le redimensionnement d'image a été ajouté avec VirtualBox 4.0.

⁴Le premier support du VMDK a été ajouté avec VirtualBox 1.4 ; depuis la version 2.1, VirtualBox supporte complètement le VMDK, ce qui veut dire que vous pouvez créer des dépôts et utiliser toutes les autres fonctions avancées décrites ci-dessus pour les images VDI avec VMDK.

⁵Le support a été ajouté avec VirtualBox 3.1.

- tous les médias actuellement attachés aux machines virtuelles ;
- les médias “enregistrés” pour la compatibilité avec les versions de VirtualBox inférieures à 4.0. Pour les détails sur les modalités du changement de l’enregistrement des médias, avec la version 4.0, merci de vous reporter à chapitre [10.1, Où VirtualBox stocke ses fichiers](#), page [196](#).

Vous pouvez visualiser et modifier les médias connus dans le **gestionnaire de médias virtuels**, auquel vous pouvez accéder à partir du menu “Fichier” de la fenêtre principale de VirtualBox :



Les médias connus sont regroupés, par commodité, dans trois onglets pour les trois formats possibles. Ces formats sont :

- Les images de disque dur, soit au format de VirtualBox Virtual Disk Image(VDI), soit dans des formats tiers listés au chapitre précédent ;
- Les images CD/DVD au format ISO standard ;
- les images de disquette au format standard RAW.

Comme vous pouvez le voir dans l’impression d’écran ci-dessus, pour chaque image, le gestionnaire de médias Virtuels vous montre le chemin complet vers le fichier image et d’autres informations, telles que la machine virtuelle à laquelle est attachée l’image, s’il y en a.

Le gestionnaire de média virtuels vous permet de

- **supprimer** une image du registre (et effacer éventuellement le fichier image en même temps) ;
- “**libérer** une image, c’est-à-dire la détacher d’une machine virtuelle si elle est rattachée actuellement à une d’elles en tant que disque dur virtuel.

À partir de la version 4.0, pour **créer de nouvelles images de disque**, merci d'utiliser l'onglet "Stockage" dans la boîte de dialogue des paramètres d'une machine virtuelle, car les images de disque sont désormais stockées par défaut dans le dossier de chaque machine.

Vous pouvez copier des fichiers images de disque dur entre systèmes hôtes et les importer alors dans les machines virtuelles, bien que certains systèmes invités (surtout Windows 2000 et XP) exigeront que la nouvelle machine virtuelle soit réglée de façon identique à l'ancienne.

Note: Ne faites pas de simples copies des images de disque dur virtuels. Si vous importez une seconde copie dans une machine virtuelle, VirtualBox se plaindra avec une erreur puisque VirtualBox affecte un identifiant unique (UUID) à chaque image de disque pour garantir qu'elle n'est utilisée qu'une fois. Voir chapitre [5.6, Cloner des images de disque](#), page [94](#) pour des instructions à ce sujet. De plus, si vous voulez copier une machine virtuelle sur un autre système, VirtualBox a un outil d'importation/exportation qui pourrait mieux convenir à vos besoins ; voir chapitre [1.14, Importer et exporter des machines virtuelles](#), page [33](#).

5.4 Modes spéciaux d'écriture d'images

Pour chaque image de disque dur virtuel supportée par VirtualBox, vous pouvez déterminer distinctement la façon dont elle peut être touchée par les opérations d'écriture par une machine virtuelle et les opérations de prise d'instantanés. Ceci vaut pour tous les formats d'image précités (VDI, VMDK, VHD ou HDD) et indépendamment du fait qu'une image soit de taille fixe ou dynamique.

Par défaut, les images sont en mode "normal". Pour marquer une image existante avec un des modes non standards listés ci-dessous, utilisez `VBoxManage modifyhd` ; voir chapitre [8.23, VBoxManage modifyhd](#), page [147](#). Sinon, utilisez `VBoxManage` pour attacher l'image à une VM et utilisez l'argument `--mtype` ; voir chapitre [8.18, VBoxManage storageattach](#), page [142](#).

1. Avec des **images normales** (le réglage par défaut), il n'y a aucune restriction de lecture et d'écriture pour les systèmes invités.

Quand vous prenez un instantané de votre machine virtuelle comme décrit au chapitre [1.10, Instantanés](#), page [28](#), l'état d'un "disque dur normal" sera enregistré avec l'instantané et, lors du retour à l'instantané, son état sera complètement réinitialisé.

(Techniquement, pour parler rigoureusement, le fichier image lui-même n'est pas "réinitialisé". Quand un instantané est pris, VirtualBox "gèle" plutôt le fichier image et n'écrit plus dedans. Pour les opérations d'écriture de la VM, un deuxième fichier image de "différenciation" est créé, qui ne reçoit que les modifications de l'image d'origine ; voir la section suivante pour les détails.)

Si vous pouvez attacher une même image "normale" à plus d'une machine virtuelle, une seule de ces machines virtuelles attachée au même fichier image peut être exécuté en même temps, sans quoi il y aurait des conflits si plusieurs machines écrivent dans le même fichier image.⁶

2. À l'opposé, les **disques durs write-through** sont complètement concernés par les instantanés : leur état *n'est pas* sauvegardé quand on prend un instantané, et il n'est pas restauré quand on restaure un instantané.

⁶Cette restriction est plus légère qu'avant VirtualBox 2.2. Jadis, chaque image de disque "normale" ne pouvait être attachée qu'à une seule machine. Maintenant, elle peut être attachée à plus d'une machine du moment qu'une seule des machines soit en fonction.

3. Les **Disques durs partageables** sont des variantes des disques durs write-through. En principe, ils se comportent exactement de la même façon, à savoir que leur état *n'est pas* sauvegardé quand on prend un instantané et il n'est pas restauré quand on restaure un instantané. La différence n'apparaît que si vous attachez de tels disques à plusieurs VMs. Les disques partageables peuvent être attachés à plusieurs VMs qui peuvent fonctionner en même temps. Cela les rend adaptés pour l'utilisation de systèmes de fichiers cluster entre des VMs et des applications identiques qui sont explicitement préparés pour accéder en même temps à un disque. Seules des images de disques fixes peuvent être utilisées dans ce mode, les images dynamiques sont rejetées.

Avertissement: C'est une fonctionnalité pour experts, dont la mauvaise utilisation peut provoquer une perte de données – les systèmes de fichiers réguliers ne sont pas préparés à gérer des modifications simultanées par plusieurs éléments.

4. Ensuite, les **images immuables** ne se souviennent des accès en écriture que de manière temporaire pendant que la machine est en fonction ; tous les changements sont perdus quand la machine virtuelle est allumée la fois d'après. Il s'en suit qu'à l'inverse des images "normales", une même image immuable peut être utilisée avec plusieurs machines virtuelles sans restrictions.

La *création* d'une image immuable a peu de sens puisqu'elle serait vide au départ et elle perdrait son contenu à chaque redémarrage de la machine (sauf si vous voulez vraiment avoir un disque toujours non formaté quand la machine démarre). Du coup, normalement, vous créeriez d'abord une image "normale" puis, quand vous estimez son contenu utile, vous la marquez plus tard comme immuable.

Si vous prenez l'instantané d'une machine avec des images immuables, sur chaque machine allumée, ces images sont réinitialisées à leur état du dernier instantané (actuel) (et non à l'état de l'image immuable d'origine).

Note: En guise d'exception particulière, les images immuables *ne sont pas* réinitialisées si elles sont attachées à une machine dont le dernier instantané a été pris pendant que la machine était en fonction (ce qu'on appelle un instantané "en ligne"). Il s'en suit que si l'instantané actuel de la machine est "en ligne", ses images immuables se comportent exactement comme les images "normales" décrites précédemment. Pour réactiver la réinitialisation automatique de telles images, effacez l'instantané actuel de la machine.

De nouveau, techniquement, VirtualBox n'écrit jamais directement sur image immuable. Toutes les opérations d'écriture de la machine seront envoyées dans une image de différenciation ; la prochaine fois que la VM sera allumée, l'image de différenciation sera rétablie à chaque démarrage de la machine, ses images immuables ont exactement le même contenu.⁷ L'image de différenciation n'est réinitialisée que lorsque la machine est allumée à partir de VirtualBox, pas quand vous redémarrez en demandant un redémarrage à l'intérieur de la machine. C'est également pourquoi les images immuables se comportent comme décrit ci-dessus quand des instantanés sont aussi présents, ce qui utilise également des images de différenciation.

Si la désactivation automatique de l'image de différenciation au démarrage d'une VM ne correspond pas à vos besoins, vous pouvez la désactiver en utilisant le paramètre `autoreset` de `VBoxManage modifyhd` ; voir chapitre [8.23, *VBoxManage modifyhd*](#), page [147](#) for details.

⁷Ce comportement a aussi changé avec VirtualBox 2.2. Jadis, les images de différenciation étaient désactivées quand la session de la machine *se terminait* ; maintenant, elles sont désactivées à chaque fois que la machine est allumée.

5. Une image en **mode multiattachée** peut être attachée à plus d'une machine virtuelle en même temps même si ces machines sont en fonction en même temps. Pour chaque machine virtuelle à laquelle une image est attachée, une image de différenciation est créée. Il s'en suit que les données écrites sur un tel disque dur virtuel par une machine n'est pas vue par les autres machines auxquelles l'image est attachée ; chaque machine crée son propre historique des écritures de l'image multiattachée.

Techniquement, une image "multiattachée" se comporte de la même façon qu'une image "immuable", sauf que l'image de différenciation n'est pas réinitialisée à chaque fois que la machine démarre.

6. Enfin, l'**image en lecture seule** est utilisée automatiquement pour les images de CD/DVD, vu que les CDs/DVDs ne sont jamais inscriptibles.

Pour illustrer les différences entre les différents types au regard des instantanés : supposons que avez installé votre système d'exploitation hôte dans votre VM et que vous avez pris un instantané. Imaginons que vous avez accidentellement infecté votre VM avec un virus et vous voulez revenir à l'instantané. Avec une image de disque dur normale, vous restaurerez simplement l'instantané et l'état antérieur de votre image de disque dur seront restaurées également (et votre infection virale sera annulée). Avec un disque dur immuable, il suffit d'éteindre et de rallumer votre VM et l'infection virale sera désactivée. Par contre, avec une image write-through, vous ne pouvez pas annuler facilement l'infection virale par la virtualisation, mais vous devrez désinfecter votre machine virtuelle comme un vrai ordinateur.

Là encore, vous pourriez trouver les images write-through utiles si vous voulez préserver des données critiques indépendamment des instantanés, et comme vous pouvez attacher plus d'une image à une VM, vous pourriez vouloir avoir une image immuable pour le système d'exploitation et une en write-through pour vos fichiers de données.

5.5 Images de différenciation

La section précédente portait sur les images de différenciation et la façon de les utiliser avec des instantanés, des images immuables et des attachements immuables. Pour l'utilisateur curieux de VirtualBox, cette section décrit avec plus de détails la façon dont elles fonctionnent.

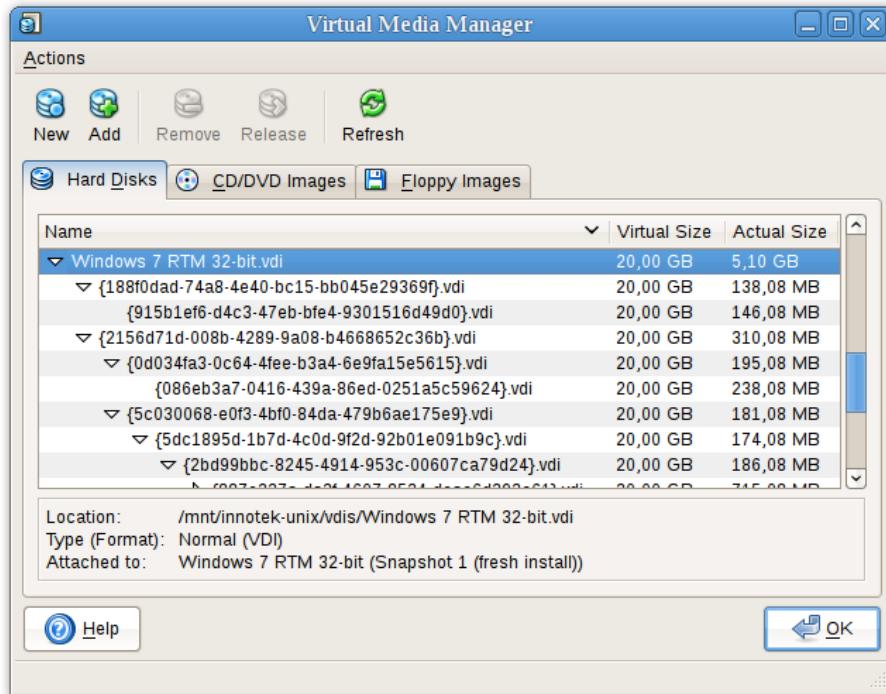
Une image de différenciation est une image de disque spéciale qui ne garde que les différences avec une autre image. En elle-même, une image de différenciation est inutile, elle doit toujours se référer à une autre image. On parle donc généralement d'une image de différenciation comme d'un "enfant" qui garde les différences d'avec son "parent".

Quand une image de différenciation est active, elle reçoit toutes les opérations d'écriture de la machine virtuelle à la place de son parent. L'image de différenciation ne contient que les secteurs du disque dur virtuel qui ont changé depuis que l'image de différenciation a été créée. Quand la machine lit un secteur à partir d'un tel disque dur virtuel, elle regarde d'abord dans l'image de différenciation. Si le secteur est présent, il est renvoyé à partir de là ; sinon VirtualBox regarde dans le parent. En d'autres termes, le parent devient en lecture seule ; on n'écrit plus jamais dedans mais il est lu si un secteur n'a pas changé.

On peut enchaîner les images de différenciation. Si une autre image de différenciation est créée pour un disque virtuel qui a déjà une image de différenciation, il devient le "petit-fils" du parent d'origine. La première image de différenciation devient alors également en lecture seule et les opérations d'écriture ne vont que dans l'image de différenciation du second niveau. Lors de la lecture à partir d'un disque virtuel, VirtualBox a besoin de regarder d'abord dans la deuxième image de différenciation, puis dans la première si le secteur n'a pas été trouvé, puis dans l'image d'origine.

Il peut y avoir un nombre illimité d'images de différenciation et chaque image peut avoir plus d'un enfant. Il s'en suit que les images de différenciation peuvent constituer une arborescence

complexe avec des parents, des “fratries” et des enfants, en fonction de la complexité de la configuration de votre machine. Les opérations d’écriture vont toujours dans l’image de différenciation “active” attachée à la machine, et pour les opérations de lecture, VirtualBox peut avoir besoin de regarder jusqu’aux parents dans la chaîne, jusqu’à ce qu’il trouve le secteur en question. Vous pouvez regarder l’arborescence dans le gestionnaire de médias virtuels :



Dans toutes ces situations, du point de vue de la machine virtuelle, le disque dur virtuel se comporte comme n’importe quel autre disque. Pendant que la machine virtuelle est en fonction, il y a un léger ralentissement (overhead) E/S d’exécution car il se peut que VirtualBox doive regarder des secteurs plusieurs fois. Cela ne se voit cependant pas, puisque les tables d’informations des secteurs sont toujours gardées en mémoire et peuvent être consultées rapidement.

On utilise des images de différenciation dans les situations suivantes :

1. **Les instantanés.** Quand vous créez un instantané comme expliqué dans la section précédente, VirtualBox “gèle” les images attachées à la machine virtuelle et crée des images de différenciation pour chacun d’eux (pour être précis, une par image qui n’est pas en mode “write-through”). Du point de vue de la machine virtuelle, les disques virtuels continuent d’agir comme avant, mais toutes les opérations d’écriture vont dans les images de différenciation. Chaque fois que vous créez un autre instantané, pour chaque disque dur attaché, une autre image de différenciation est créée et attachée, formant une chaîne ou une arborescence.

Dans l’impression d’écran ci-dessus, vous voyez que l’image du disque d’origine est maintenant attachée à un instantané, ce qui représente l’état du disque quand on a pris l’instantané.

Si vous **restaurez** maintenant un instantané – à savoir, si vous voulez revenir à l’état exact de la machine stocké dans le dépôt –, il se produit la chose suivante :

- a) VirtualBox copie les paramètres de la machine virtuelle copiés dans vinstantané vers la machine virtuelle. Du coup, si vous avez fait des modifications dans la configuration après avoir pris l’instantané, elles sont annulées.

- b) Si vous avez pris l'instantané quand la machine était en fonction, il contient un état sauvegardé de la machine et cet état est restauré également ; après la restauration de l'instantané, la machine sera en état “sauvegardé” et elle reprendra son exécution là où elle avait été commencée la fois suivante. Sinon, la machine sera dans l'état “éteint” et elle fera un démarrage complet.
- c) Pour chaque image de disque attachée, à la machine, l'image de différenciation contenant les opérations d'écriture depuis que l'instantané actuel a été pris est supprimée et l'image du parent originel est réactivée. (Si vous avez restauré l'instantané “racine”, ce sera l'image de disque racine de tous les attachements ; sinon ce sera d'autres images de différenciation descendantes). Cela restaure de fait l'ancien état de la machine.

Si vous **effacez** plus tard un instantané afin de gagner de l'espace disque, chaque attachement de disque des images de différenciation devient obsolète. Dans ce cas, l'image de différenciation du disque ne peut pas être tout simplement effacée. VirtualBox doit plutôt regarder chaque secteur de l'image de différenciation et le copier dans le parent ; ceci s'appelle des images de “synchronisation” et cela peut être une procédure longue selon la taille de l'image de différenciation. Il se peut qu'il faille temporairement une importante quantité d'espace disque supplémentaire avant que l'image de différenciation rendue obsolète par l'opération de synchronisation ne soit effacée.

2. **Les images immuables.** Quand une image passe en mode “immuable”, une image de différenciation est créée également. Comme avec les instantanés, l'image parent devient alors en lecture seule et l'image de différenciation reçoit toutes les opérations d'écriture. Chaque fois qu'on démarre la machine virtuelle, toutes les images immuables qui y sont attachées voient leur image de différenciation spécifique nettoyée, ce qui réinitialise de fait le disque virtuel de la machine virtuelle à chaque redémarrage.

5.6 Cloner des images de disque

Vous pouvez dupliquer des fichiers images de disque dur sur le même hôte pour produire rapidement une deuxième machine virtuelle avec la même configuration de système d'exploitation. Néanmoins, vous devriez ne faire des copies d'images de disques durs virtuels *que* en utilisant l'outil fourni par VirtualBox ; voir chapitre 8.24, *VBoxManage clonehd*, page 148. Ceci car VirtualBox affecte un numéro d'identifiant unique (UUID) à chaque image de disque, qui est stocké dans l'image et VirtualBox refusera de fonctionner avec deux images ayant le même numéro. Si vous essayez, par accident, de réimporter une image de disque que vous avez copiée normalement, vous pouvez faire une seconde copie en utilisant l'outil de VirtualBox et l'importer à la place.

Remarquez que les distributions Linux récentes identifient le disque dur de démarrage à partir de l'ID du disque. L'ID utilisé par VirtualBox pour un lecteur est déterminé à partir de l'UUID de l'image du disque dur virtuel. Donc si vous clonez une image de disque et si vous essayez de démarrer l'image, copiée il se peut que l'invité ne puisse pas déterminer son propre disque de démarrage car l'UUID a changé. Dans ce cas, vous devez adapter l'ID du disque dans le script de votre chargeur de démarrage (par exemple */boot/grub/menu.lst*). L'ID d'un disque ressemble à ceci :

```
scsi-SATA_VBOX_HARDDISK_VB5cfdb1e2-c251e503
```

L'ID de l'image copiée peut être déterminée avec

```
hdparm -i /dev/sda
```

5.7 Mise en cache des E/S dans l'hôte

À partir de la version 3.2, VirtualBox peut éventuellement désactiver la mise en cache de l'E/S qu'effectuerait le système d'exploitation hôte sur les fichiers images du disque.

Traditionnellement, VirtualBox ouvrait des fichiers images de disque comme des fichiers normaux, ce qui faisait qu'ils étaient mis en cache par le système d'exploitation hôte comme n'importe quel autre fichier. Le principal avantage en est la vitesse : quand l'OS invité écrit sur le disque et quand le cache de l'hôte utilise l'écriture différée, l'opération d'écriture peut être déclarée terminée pour l'OS invité rapidement alors que l'OS hôte peut effectuer l'opération sans synchronisation. De même, quand vous démarrez une VM une deuxième fois et si vous avez assez de mémoire disponible pour que l'OS l'utilise pour la mise en cache, de grandes parties du disque dur virtuel peuvent aller dans la mémoire du système et la VM peut accéder plus rapidement aux données.

Remarquez que cela ne s'applique qu'aux fichiers images ; la mise en tampon ne fonctionnait jamais pour les disques présents sur des supports iSCSI distants, ce qui est le scenario le plus classique dans les paramétrages du type enterprise (voir chapitre [5.10, Serveurs iSCSI](#), page [97](#)).

Si la mise en tampon est un paramètre par défaut utile pour virtualiser quelques machines sur un ordinateur de bureau, elle présente quelques inconvénients :

1. L'écriture différée dans le cache de l'OS hôte est moins sécurisée. Quand l'OS invité écrit des données, il considère que les données sont écrites même si elles ne sont pas encore arrivées sur le disque physique. Si, pour une raison quelconque, l'écriture n'a pas lieu (problème électrique, plantage de l'hôte), les chances de perdre des données augmentent.
2. Les fichiers images de disque ont tendance à être importantes. Leur mise en cache utilise donc très vite tout le cache de l'OS hôte. Selon l'efficacité de la gestion du cache par l'OS hôte, cela peut ralentir énormément l'hôte, surtout si plusieurs VMs fonctionnent en même temps. Par exemple, sur des hôtes Linux, la mise en cache de l'hôte peut aboutir à un report, par Linux, de toutes les écritures jusqu'à ce que le cache de l'hôte soit presque plein, avant d'écrire tous ces changements en une fois, ce qui peut suspendre l'exécution d'une VM pendant quelques minutes. Cela peut donner des erreurs d'E/S dans l'invité car les requêtes E/S excéderaient le timeout.
3. La mémoire physique est souvent gaspillée, car les systèmes d'exploitation hôtes ont en général leur propre système de mise en cache des E/S, ce qui aboutit à la mise en cache double des données (à la fois dans le cache de l'invité et de l'hôte), avec peu d'effet.

Si vous décidez de désactiver la mise en cache des E/S dans l'hôte, pour les raisons ci-dessus, VirtualBox utilise son propre petit cache pour mettre les écritures en tampon, mais en général, aucune lecture de ce cache n'est fait par l'OS invité. De plus, VirtualBox supporte complètement l'E/S asynchrone pour ses contrôleurs SATA, SCSI et SAS virtuels via plusieurs threads d'E/S.

Les E/S asynchrones n'étant pas supportées par les contrôleurs IDE, pour des raisons de performance, vous pourriez vouloir laisser la mise en cache de l'hôte pour les contrôleurs IDE virtuels de votre VM.

Pour cette raison, VirtualBox vous permet de configurer si la mise en cache des E/S est utilisée pour chaque contrôleur E/S, indépendamment. Soit décochez la case "Utiliser la mise en cache des E/S de l'hôte" des paramètres de stockage, d'un contrôleur de stockage donné, soit utilisez la commande VBoxManage suivante pour désactiver la mise en cache des E/S de l'hôte pour un contrôleur de stockage virtuel :

```
VBoxManage storagectl <vm> --name <nomcontrôleur> --hostiocache off
```

Voir chapitre [8.19, VBoxManage storagectl](#), page [145](#) pour les détails.

Également pour les raisons ci-dessus, VirtualBox utilise maintenant par défaut des contrôleurs SATA pour les nouvelles machines virtuelles.

5.8 Limiter la bande passante des images de disque

À partir de la version 4.0, VirtualBox permet de limiter la bande passante maximale utilisée pour les E/S asynchrones. De plus, il supporte le partage des limites entre des groupes de bandes passantes pour plusieurs images. Il est possible d'avoir plus d'une limite.

Les limites sont configurées via VBoxManage. L'exemple ci-dessous crée un groupe de bandes passantes nommé "Limit", et pose la limite à 20 Mo/s et affecte le groupe aux disques attachés à la VM :

```
VBoxManage bandwidthctl "nom VM" add Limit --type disk --limit 20M
VBoxManage storageattach "nom VM" --controller "SATA" --port 0 --device 0 --type hdd
  --medium disk1.vdi --bandwidthgroup Limit
VBoxManage storageattach "nom VM" --controller "SATA" --port 1 --device 0 --type hdd
  --medium disk2.vdi --bandwidthgroup Limit
```

Tous les disques d'un groupe partagent la limite de la bande passante, c'est-à-dire que dans l'exemple ci-dessus,, la bande passante des deux images combinées ne peut jamais dépasser 20Mo/s. Toutefois, si un disque n'a pas besoin de bande passante, l'autre peut utiliser la bande passante restante dans son groupe.

Les limites pour chaque groupe peuvent être modifiées pendant que la VM est en fonction, ce qui applique immédiatement les modifications. L'exemple ci-dessous modifie le groupe créé dans l'exemple ci-dessus en 10 Mo/s:

```
VBoxManage bandwidthctl "nom VM" set Limit --limit 10M
```

5.9 Support des CD/DVD

Le/les lecteur(s) CD/DVD virtuels ne supportent par défaut que la lecture. Vous pouvez modifier la configuration d'un média pendant l'exécution. Vous pouvez choisir entre trois options pour présenter les données d'un média :

- **Lecteur hôte** définit que l'invité peut lire sur un média du lecteur hôte.
- **Fichier image** (en général un fichier ISO) donne à l'invité un accès en lecture seule aux données de l'image.
- **EVide** signifie un lecteur sans média dedans.

La modification entre les situations ci-dessus, le changement de média dans le lecteur hôte ou de fichier image signalera un changement de méca au système d'exploitation invité, lequel peut réagir au changement (par exemple, en démarrant un programme d'installation).

Les changements de médias peuvent être empêchés par l'invité et VirtualBox répercute cela en verrouillant le lecteur hôte si besoin. Vous pouvez forcer le retrait d'un média en pareilles situations via l'interface graphique de VirtualBox ou l'outil en ligne de commandes VBoxManage. En fait, cela revient à une éjection en urgence, ce que supportent de nombreux lecteurs de CD/DVD avec tous les effets collatéraux associés : l'OS invité peut renvoyer des messages d'erreur comme sur du vrai matériel, les applications invitées peuvent mal se comporter. Utilisez ceci avec précaution.

Note: La chaîne d'identification du lecteur fournie à l'invité (qui serait affichée, dans l'invité, par les outils de configuration tels que le gestionnaire de périphériques de Windows) est toujours "VBOX CD-ROM", indépendamment de la configuration actuelle du lecteur virtuel. Ceci pour empêcher la détection d'être faite en retard dans le système d'exploitation invité à chaque fois que la configuration change.

L'émulation standard des CD/DVD ne permet de lire que des formats de CD et de DVD de données standards. Une possibilité supplémentaire, expérimentale, est de donner un accès direct de l'invité au lecteur CD/DVD de l'hôte en activant le mode "passthrough". Selon le matériel hôte, cela peut potentiellement faire marcher trois choses :

- L'écriture sur CD/DVD depuis l'invité, si le lecteur DVD de l'hôte est un graveur de CD/DVD ;
- la lecture de CDs audio ;
- la lecture de DVDs chiffrés.

Il existe une case à cocher "Passthrough" dans la boîte de dialogue graphique de configuration du média attaché aux contrôleurs de stockage, ou vous pouvez utiliser l'option `--passthrough` de `VBoxManage storageattach` ; voir chapitre [8.18, *VBoxManage storageattach*, page 142](#) pour les détails.

Même si pass-through est activé, les commandes non sûres telles que la mise à jour du firmware du lecteur, seront bloquées. Les formats de CD vidéo ne sont pas du tout supportés, même pas en mode passthrough, et on ne peut pas les lire à partir d'une machine virtuelle.

Sur les hôtes Solaris, pass-through exige de lancer VirtualBox avec de vrais droits d'administrateur du fait de mesures de sécurité renforcées par l'hôte.

5.10 Serveurs iSCSI

iSCSI signifie "Internet SCSI" et c'est un standard qui permet d'utiliser le protocole SCSI à travers des connexions Internet. (TCP/IP). En particulier, avec l'arrivée du Gigabit Ethernet, on peut désormais se permettre d'attacher des serveurs de stockage iSCSI simplement comme des disques durs distants à un réseau d'ordinateurs. Dans la terminologie iSCSI, le serveur fournissant les ressources de stockage s'appelle la "cible iSCSI", tandis que le client qui se connecte au serveur et qui accède à ses ressources s'appelle "l'initiateur iSCSI".

VirtualBox peut présenter de manière transparente du stockage distant iSCSI à une machine virtuelle en tant que disque dur. Le système d'exploitation ne verra pas de différence entre une image de disque virtuel (fichier VDI) et une cible iSCSI. Pour obtenir cela, VirtualBox comporte un initiateur iSCSI intégré.

Le support iSCSI de VirtualBox a été développé selon le standard iSCSI et il devrait fonctionner avec toutes les cibles iSCSI conformes au standard. Pour utiliser une cible iSCSI avec VirtualBox, vous devez utiliser la ligne de commande ; voir chapitre [8.18, *VBoxManage storageattach*, page 142](#).

6 Le réseau virtuel

Comme indiqué brièvement au chapitre [3.8, Paramètres réseau](#), page [58](#), VirtualBox fournit jusqu'à huit cartes Ethernet PCI virtuelles pour chaque machine virtuelle. Pour chaque carte, vous pouvez sélectionner individuellement

1. le matériel virtualisé ainsi que
2. le mode de virtualisation effectué par la carte virtuelle par rapport à votre matériel réseau physique sur l'hôte.

Quatre des cartes réseaux peuvent être configurées dans la section "Réseau" de la boîte de dialogue des paramètres de l'interface graphique de VirtualBox. Vous pouvez configurer les huit cartes réseaux en ligne de commande avec VBoxManage modifyvm ; voir chapitre [8.8, VBoxManage modifyvm](#), page [130](#).

Ce chapitre explique les différents paramètres réseaux avec davantage de détails.

6.1 Matériel réseau virtuel

Pour chaque carte, vous pouvez sélectionner individuellement le type de *matériel* qui sera présenté à la machine virtuelle. VirtualBox peut virtualiser les six types de matériel réseau suivants :

- AMD PCNet PCI II (Am79C970A) ;
- AMD PCNet FAST III (Am79C973, par défaut) ;
- Intel PRO/1000 MT Desktop (82540EM) ;
- Intel PRO/1000 T Server (82543GC) ;
- Intel PRO/1000 MT Server (82545EM) ;
- Adaptateur réseau paravirtualisé (virtio-net).

PCNet FAST III est celle par défaut parce qu'elle est supportée par presque tous les systèmes d'exploitation non inclus ainsi que par le chargeur de démarrage GNU GRUB. Par exception, les adaptateurs de la famille Intel PRO/1000 ont été choisis pour certains types de systèmes d'exploitation invités qui n'incluent plus de pilotes pour la carte PCNet, tel que Windows Vista.

Le type Intel PRO/1000 MT Desktop fonctionne avec Windows Vista et les versions supérieures. La variante T Server de la carte Intel PRO/1000 est reconnue par les invités Windows XP sans installer de pilotes supplémentaires. La variante MT Server facilite les imports d'OVF à partir d'autres plateformes.

“L'adaptateur réseau paravirtualisé (virtio-net)“ est spécial. Si vous le sélectionnez, VirtualBox *ne virtualise pas* du matériel réseau classique (à savoir supporté par les systèmes d'exploitation invités non intégrés). VirtualBox s'attend alors à ce qu'une interface logicielle spéciale pour les environnements virtualisés provienne de l'invité, évitant ainsi la complexité de l'émulation du matériel réseau et de la performance d'importation du réseau. À partir de la version 3.1, VirtualBox fournit un support des pilotes réseaux du standard industriel “virtio”, qui font partie du projet libre KVM.

Les pilotes réseaux “virtio” sont disponibles pour les systèmes d'exploitation invités suivants :

- Les noyaux Linux version 2.6.25 ou supérieure peuvent être configurés pour fournir le support virtio ; certaines distributions ont back-porté aussi virtio dans d'anciens noyaux.
- Pour Windows 2000, XP et Vista, les pilotes virtio peuvent être téléchargés et installés sur la page Web du projet KVM.¹

VirtualBox contient aussi un support limité pour ce qu'on appelle **jumbo frames**, c'est-à-dire les paquets réseaux de plus de 1500 octets de données, si vous utilisez le réseau Intel de virtualisation bridé. En d'autres termes, jumbo frames ne sont pas supportés avec les périphériques réseaux AMD ; dans ce cas, jumbo packets will seront rejetés en silence côté récepteur et transmetteur. Les systèmes d'exploitation invités qui essaient d'utiliser cette fonctionnalité verront cela comme une perte de paquets, ce qui peut provoquer un comportement inattendu de l'application dans l'invité. Cela ne pose pas problème avec les systèmes d'exploitation invités dans leur configuration par défaut, vu que jumbo frames doit être explicitement activé.

6.2 Introduction aux modes réseaux

Chacun des huit adaptateurs réseaux peut être configuré séparément pour agir dans l'un des modes suivants :

Non attaché Dans ce mode, VirtualBox dit à l'invité qu'une carte réseau est présente, mais qu'il n'y a pas de connexion – comme si aucun câble Ethernet n'était branché dans la carte. De cette façon, il est possible de “retirer” le câble réseau virtuel Ethernet et de couper la connexion, ce qui peut être utile pour informer un système d'exploitation invité qu'aucune connexion réseau n'est disponible, et ceci renforce une reconfiguration.

Network Address Translation (NAT) Si vous ne voulez que naviguer sur le Web, télécharger des fichiers et lire des messages dans l'invité, ce mode par défaut devrait vous suffir et vous pouvez sauter sans souci le reste de cette section. Merci de remarquer qu'il existe certaines limitations quand on utilise le partage de fichiers Windows (voir chapitre 6.3.3, *Limites du NAT*, page 101 pour des détails).

Réseau avec pont Ceci est pour les besoins réseaux plus avancés tels que des simulations de réseaux et des exécutions de serveurs dans un invité. Lorsque vous l'activez, VirtualBox se connecte à une de vos cartes réseaux installées et il échange des paquets réseaux directement, dépassant la pile réseau du système d'exploitation de votre hôte.

Réseau interne On peut l'utiliser pour créer un type différent de réseau sur une base logicielle, visible pour les machines sélectionnées, mais pas pour les applications de l'hôte ou du monde extérieur.

Réseau Host-only On peut l'utiliser pour créer un réseau contenant l'hôte et un ensemble de machines virtuelles, sans avoir besoin de l'interface réseau physique de l'hôte. À la place, une interface réseau virtuelle (identique à une interface loopback) est créée sur l'hôte, offrant une connectivité entre les machines virtuelles et l'hôte.

Réseau générique Mode rarement utilisé, il partage la même interface réseau générique en permettant à l'utilisateur de sélectionner un pilote qui peut être inclus dans VirtualBox ou distribué dans un pack d'extension.

Pour l'instant, il existe potentiellement deux sous-modes disponibles :

Tunnel UDP On peut l'utiliser pour interconnecter directement, facilement et de manière transparente des machines virtuelles qui fonctionnent sur différents hôtes, via une infrastructure réseau existante.

¹<http://www.linux-kvm.org/page/WindowsGuestDrivers>.

Réseau VDE (Virtual Distributed Ethernet) Cette option peut être utilisée pour se connecter à un service Ethernet distribué virtuel sur un hôte Linux ou FreeBSD. Pour l'instant ceci nécessite de compiler VirtualBox à partir des sources car les paquets d'Oracle ne l'incluent pas.

Les sections suivantes décrivent les modes réseaux disponibles avec plus de détails.

6.3 Network Address Translation (NAT)

Network Address Translation (NAT) est la manière la plus simple d'accéder à un réseau externe à partir d'une machine virtuelle. Habituellement, cela n'exige aucune configuration sur le réseau hôte ou le système invité. C'est pourquoi c'est le mode réseau par défaut de VirtualBox.

Une machine virtuelle dont NAT est activé agit exactement comme un vrai ordinateur qui se connecte à Internet par un routeur. Le “routeur”, dans ce cas, est le moteur réseau de VirtualBox, qui dirige le trafic depuis et vers la machine virtuelle de façon transparente. Dans VirtualBox, ce routeur se place entre chaque machine virtuelle et l'hôte. Cette séparation maximise la sécurité puisque, par défaut, les machines virtuelles ne peuvent pas se parler.

L'inconvénient du mode NAT est que, comme dans un réseau privé, derrière un routeur, la machine virtuelle est invisible et injoignable depuis le réseau extérieur ; vous ne pouvez pas lancer de serveur de cette façon, sauf si vous réglez une redirection de ports (décrise ci-dessous).

Les blocs réseaux envoyés par le système d'exploitation invité sont reçus par le moteur NAT de VirtualBox qui extrait les données TCP/IP et les envoie en utilisant le système d'exploitation hôte. Pour une application de l'hôte ou un autre ordinateur du même réseau comme l'hôte, cela fonctionne comme si des données étaient envoyées par l'application VirtualBox de l'hôte, en utilisant une adresse IP appartenant à l'hôte. VirtualBox écoute les réponses aux paquets envoyés et les réempaquette et les renvoie à la machine invitée sur son réseau privé.

La machine virtuelle reçoit son adresse et sa configuration réseau sur le réseau privé à partir d'un serveur DHCP intégré à VirtualBox. L'adresse IP ainsi affectée à la machine virtuelle se trouve en général sur un réseau complètement différent de l'hôte. On peut paramétriser l'utilisation de NAT pour autant de cartes qu'a une machine virtuelle, la première carte est connectée au réseau privé sur 10.0.2.0, la deuxième carte sur 10.0.3.0 et ainsi de suite. Si vous avez besoin de modifier la plage d'adresses affectées à l'invité pour une raison quelconque, merci de vous reporter à la chapitre [9.12, Peaufiner le moteur NAT de VirtualBox](#), page [178](#).

6.3.1 Configurer la redirection de ports avec NAT

Comme la machine virtuelle est connectée à un réseau privé interne de VirtualBox et invisible pour l'hôte, les services réseaux de l'invité ne sont pas accessibles à la machine hôte ou à d'autres ordinateurs du même réseau. Cependant, comme un routeur physique, VirtualBox peut rendre disponibles des services sélectionnés pour le monde extérieur à l'invité via la **redirection de port**. Cela veut dire que VirtualBox écoute certains ports sur l'hôte et renvoie tous les paquets qui y arrivent vers l'invité, sur le même port ou sur un autre.

Pour une application de l'hôte ou d'autres machines physiques (ou virtuelles) du réseau, cela fonctionne comme si les services étaient derrière un proxy qui tournerait en fait sur l'hôte. Cela signifie également que vous ne pouvez pas lancer le même service sur les mêmes ports de l'hôte. Néanmoins, vous pouvez toujours tirer parti de lancer un service dans une machine virtuelle – par exemple, les services de la machine hôte ou d'autres machines virtuelles ne peuvent pas -être atteintes ou plantées par une faille ou un bogue du service, et le service peut fonctionner dans un autre système d'exploitation que le système hôte.

Pour configurer la redirection de ports, vous pouvez utiliser l'éditeur graphique de redirection de ports que vous trouverez dans la boîte de dialogue des paramètres réseaux des adaptateurs réseaux configurés pour utiliser NAT. Vous pouvez y orienter les ports de l'hôte vers les ports de l'invité pour permettre au trafic réseau d'être acheminé sur un port spécifique de l'invité.

Vous pourriez utiliser un autre outil en ligne de commande, `VBoxManage` ; pour les détails, merci de vous reporter au chapitre [8.8, `VBoxManage modifyvm`, page 130](#).

Vous devrez savoir les ports de l'invité utilisés par les services de l'invité et décider des ports à utiliser sur l'hôte (souvent, mais pas toujours, vous voudrez utiliser les mêmes ports sur l'invité et sur l'hôte). Vous pouvez utiliser n'importe quel port de l'hôte qui ne sont pas déjà utilisés par un service. Par exemple, pour régler les connexions NAT entrantes pour un serveur `ssh` de l'invité, utilisez la commande suivante :

```
VBoxManage modifyvm "nom VM" --natpf1 "guestssh,tcp,,2222,,22"
```

Avec l'exemple ci-dessus, tout le trafic TCP arrivant sur le port 2222 de n'importe quelle interface de l'hôte sera redirigé sur le port 22 de l'invité. Le nom du protocole `tcp` est un attribut obligatoire définissant le protocole qu'il faudrait utiliser pour la redirection (on pourrait utiliser `udp`). Le nom `guestssh` est purement descriptif et il sera auto-généré si vous n'en mettez pas. Le numéro après `--natpf` indique la carte réseau, comme dans d'autres endroits de `VBoxManage`.

Pour supprimer de nouveau cette règle de redirection, utilisez la commande suivante :

```
VBoxManage modifyvm "nom VM" --natpf1 delete "guestssh"
```

Si, pour une raison quelconque, l'invité utilise une adresse IP affectée de manière statique non gérée par le serveur DHCP interne, vous devez spécifier l'IP de l'invité lors de l'enregistrement de la règle de redirection :

```
VBoxManage modifyvm "nom VM" --natpf1 "guestssh,tcp,,2222,10.0.2.19,22"
```

Cet exemple est identique au précédent, sauf que qu'on dit au moteur NAT qu'il peut trouver l'invité à l'adresse 10.0.2.19.

Pour rediriger *tout* le trafic rentrant depuis une interface spécifique de l'hôte sur l'invité, spécifiez l'IP de cette interface de l'hôte comme ceci :

```
VBoxManage modifyvm "nom VM" --natpf1 "guestssh,tcp,127.0.0.1,2222,,22"
```

Ceci redirige tout le trafic TCP arrivant sur l'interface localhost (127.0.0.1) via le port 2222 sur le port 22 de l'invité.

Il est possible de configurer les connexions NAT entrantes pendant que la VM est en fonction, voir chapitre [8.13, `VBoxManage controlvm`, page 140](#).

6.3.2 Démarrer avec PXE avec NAT

Le démarrage avec PXE est désormais supporté en mode NAT. Le serveur DHCP de NAT fournit un fichier d'amorçage dont le nom ressemble à `nomvm.pxe` si le répertoire TFTP existe dans le répertoire où se trouve le fichier `VirtualBox.xml` de l'utilisateur. L'utilisateur est chargé de fournir `nomvm.pxe`.

6.3.3 Limites du NAT

Il y a quatre **limites** du NAT que les utilisateurs devraient savoir :

Limite du protocole ICMP : Certains outils de débogage réseau souvent utilisés (comme `ping` ou tracerouting) s'appuient sur le protocole ICMP pour envoyer/recevoir des messages. Si le support ICMP a été amélioré avec VirtualBox 2.1 (`ping` devrait maintenant fonctionner), d'autres outils peuvent ne pas marcher de manière fiable.

La réception des broadcasts UDP n'est pas fiable : L'invité ne reçoit pas de broadcasts fiables car, pour économiser des ressources, il n'écoute qu'un certain temps après que l'invité a envoyé des données UDP sur un port particulier. En conséquence, la résolution de nom NetBios basée sur les broadcasts ne fonctionne pas toujours (mais WINS fonctionne toujours). Un contournement est d'utiliser l'IP numérique du serveur désiré en notation `\server\share`.

Les protocoles tels que GRE ne sont pas supportés : Les protocoles autres que TCP et UDP ne sont pas supportés. Cela signifie que certains produits VPN (comme PPTP de Microsoft) ne peuvent pas être utilisés. Il existe d'autres produits VPN qui utilisent simplement TCP et UDP.

Redirection des ports de l'hôte < 1024 impossible : Sur les hôtes basés sur Unix, (comme Linux, Solaris, Mac OS X), il n'est pas possible de trouver des ports en-dessous de 1024 pour les applications non lancées par root. Il s'en suit que si vous essayez de configurer la redirection de tels port, la VM refusera de démarrer.

Ces limites ne concernent normalement pas les utilisations standards du réseau. Mais la présence de NAT a également des effets subtils qui peuvent interférer avec des protocoles qui, en principe, fonctionnent. Un exemple est NFS, où le serveur est souvent configuré pour refuser les connexions depuis des ports non privilégiés (donc les ports qui ne sont pas inférieurs à 1024).

6.4 Réseau Bridged

Avec le réseau bridgé, VirtualBox utilise un pilote de périphérique sur votre système hôte qui filtre les données de votre adaptateur réseau physique. Ce pilote s'appelle donc un pilote “net filter”. Il permet à VirtualBox d'intercepter les données du réseau physique et d'y envoyer des données, ce qui crée de fait une nouvelle interface réseau logicielle. Quand un invité utilise une telle interface, cela se passe, le sur le système hôte, comme si l'invité était connecté physiquement à l'interface réseau en utilisant un câble réseau : l'hôte peut envoyer des données à l'invité via cette interface et en reçoit des données. Cela veut dire que vous pouvez régler du routage ou des ponts entre l'invité et le reste de votre réseau.

Pour que cela fonctionne, VirtualBox a besoin d'un pilote de périphérique sur votre système hôte. La manière dont fonctionne le réseau bridgé a été complètement réécrite avec VirtualBox 2.0 et 2.1, selon le système d'exploitation hôte. Du point de vue utilisateur, la principale différence est qu'une configuration complexe n'est plus nécessaire, quel que soit le système d'exploitation hôte supporté.²

Note: Même si TAP n'est plus nécessaire sur Linux avec le réseau bridgé, vous *pouvez* toujours utiliser les interfaces TAP pour certains paramétrages avancés puisque vous pouvez connecter une VM à n'importe quel interface de l'hôte – qui pourrait être également une interface TAP.

Pour activer le réseau bridgé, tout ce que vous devez faire est d'ouvrir la boîte de dialogue des paramètres d'une machine virtuelle, d'aller sur l'onglet “Réseau” et de sélectionner “Réseau bridge” dans la boîte à liste déroulante du champ “Attaché à”. Au départ, sélectionnez l'interface désirée de l'hôte dans la liste en bas de la fenêtre, qui contient les interfaces réseaux physiques de vos systèmes. Sur un MacBook physique, par exemple, cela vous permettra de choisir entre “en1: AirPort” (qui est l'interface sans fil) et “en0: Ethernet”, qui représente l'interface avec câble réseau.

²Pour les hôtes Mac OS X et Solaris, les pilotes net filter étaient déjà ajoutés à VirtualBox 2.0 (vu que le support de Host Interface Networking existait à l'origine sur ces plateformes). Avec VirtualBox 2.1, les pilotes net filter ont été également ajoutés pour les hôtes Windows et Linux à la place des mécanismes précédemment présents dans VirtualBox pour ces plateformes ; surtout sur Linux, l'ancienne méthode impliquait de créer des interfaces TAP et des ponts, ce qui était complexe et variait d'une distribution à l'autre. Rien de tout cela n'est désormais nécessaire. Le réseau bridgé s'appelait jadis “Host Interface Networking” et on l'a renommé avec la version 2.2 sans changer ses fonctionnalités.

Note: Créer un pont avec une interface sans fil se fait différemment d'avec une interface filaire, car la plupart des adaptateurs sans fil ne supportent pas le mode promiscuus. Tout le trafic doit utiliser l'adresse MAC de l'adaptateur sans fil de l'hôte, donc VirtualBox doit remplacer l'adresse MAC source dans l'en-tête Ethernet d'un paquet sortant pour s'assurer que la réponse sera envoyée à l'interface hôte. Quand VirtualBox voit un paquet entrant ayant pour adresse IP de destination celle appartenant à un des adaptateurs d'une machine virtuelle, il remplace l'adresse MAC de destination dans l'en-tête Ethernet par l'adresse MAC de l'adaptateur de la VM et il l'envoie. VirtualBox examine les paquets ARP et DHCP afin de découvrir les adresses IP des machines virtuelles.

Selon votre système d'exploitation hôte, vous devriez garder en tête les limites suivantes :

- Sur les hôtes **Macintosh**, la fonctionnalité est limitée quand on utilise AirPort (le réseau sans fil de Mac) pour du réseau bridgé. Actuellement, VirtualBox ne supporte l'IPv4 qu'avec AirPort. Pour les autres protocoles tels qu'IPv6 et IPX, vous devez choisir une interface filaire.
- Sur les hôtes **Linux**, la fonctionnalité est limitée quand on utilise les interfaces sans fil pour le réseau bridgé. Actuellement, VirtualBox supporte le sans fil qu'en IPv4. Pour les autres protocoles tels qu'IPv6 et IPX, vous devez choisir une interface filaire.

De plus, le paramétrage du MTU sur moins de 1500 octets sur les interfaces filaires fournies par le pilote sky2 sur les Marvell Yukon II EC Ultra Ethernet NIC est connu pour provoquer une perte de paquets dans certaines conditions.

Certains adaptateurs nettoient les tags VLAN matériellement. Cela ne permet pas d'utiliser le troncage de VLAN entre une VM et le réseau extérieur avec les noyaux Linux pre-2.6.27, ni avec les systèmes d'exploitation hôtes autres que Linux.

- Sur les hôtes **Solaris**, il n'y a aucun support pour utiliser les interfaces sans fil. Le filtrage du trafic de l'invité par IPFilter n'est pas complètement supporté non plus à cause de restrictions techniques du sous-système réseau de Solaris. Ces problèmes devraient être résolus dans la future version Solaris 11.

À partir de VirtualBox 4.1, sur les hôtes Solaris 11 (construction 159 et supérieur), il est possible d'utiliser les Crossbow Virtual Network Interfaces (VNICS) de Solaris directement, avec VirtualBox, sans configuration dépassant l'exclusivité de chaque VNIC pour chaque interface réseau de l'invité. Avec VirtualBox 2.0.4 et supérieur, VNICS peuvent être utilisés, mais avec les précautions suivantes :

- Un VNIC ne peut pas être partagé entre plusieurs interfaces réseaux invitées, c'est-à-dire que chaque interface réseau invitée doit avoir son propre et exclusif VNIC.
- Il faut affecter au VNIC et à l'interface réseau invitée qui utilise VNIC des adresses MAC identiques.

Quand on utilise des interfaces VLAN avec VirtualBox, il faut les nommer selon le schéma de nommage PPA-hack (par exemple "e1000g513001"), sans quoi l'invité pourrait recevoir des paquets dans un format imprévu.

6.5 Réseau interne

Le réseau interne est identique à celui bridgé dans le sens où la VM peut communiquer directement avec le monde extérieur. Toutefois, le "monde extérieur" se limite aux autres VMs sur le même hôte et connectées au même réseau interne.

Même si, techniquement, on peut faire tout ce qu'on fait avec un réseau interne avec un réseau bridgé, il présente des avantages de sécurité. En mode réseau bridgé, tout le trafic passe

par l'interface physique du système hôte. Il est donc possible d'attacher un sniffeur de paquets (tel que Wireshark) à l'interface hôte et d'enregistrer tout le trafic qui y transite. Si, pour une raison quelconque, vous préférez que deux ou plusieurs VMs sur une même machine communiquent en privé, en cachant leurs données au système et à l'utilisateur hôtes, le réseau bridgé n'est donc pas envisageable.

Les réseaux internes sont créés automatiquement en tant que de besoin c'est-à-dire qu'il n'y a pas de configuration centrale. Chaque réseau interne est identifié simplement par son nom. Une fois qu'il y a plus d'une carte réseau virtuelle active avec le même ID réseau interne, le pilote support de VirtualBox "branchera" automatiquement les cartes et agira comme un switch. Les pilotes support de VirtualBox implémentent un switch Ethernet complet et supportent les frames broadcast/multicast et le mode promiscuous.

Afin d'attacher la carte réseau d'une VM à un réseau interne, réglez son mode réseau sur "réseau interne". Il existe de manières de faire cela :

- Vous pouvez utiliser une boîte de dialogue "Paramètres" de la VM dans l'interface graphique de VirtualBox. Dans la catégorie "Réseau" de la boîte de dialogue des paramètres, sélectionnez "réseau interne" dans la liste déroulante des modes réseaux. Maintenant, sélectionnez le nom d'un réseau interne existant dans la liste déroulante en-dessous ou tapez un nouveau nom dans la zone d'édition.
- Vous pouvez utiliser

```
VBoxManage modifyvm "nom VM" --nic<x> intnet
```

Éventuellement, vous pouvez spécifier un nom de réseau par la commande

```
VBoxManage modifyvm "nom VM" --intnet<x> "nom réseau"
```

Si vous ne spécifiez pas de nom réseau, la carte réseau sera attachée au réseau intnet par défaut.

Sauf si vous configurez les cartes réseaux (virtuelles) dans les systèmes d'exploitation invités qui participent au réseau interne pour utiliser des adresses IP statiques, vous pourriez vouloir utiliser le serveur DHCP qui est construit dans VirtualBox pour gérer des adresses IP pour le réseau interne. Merci de voir chapitre [8.35, *VBoxManage dhcpserver*, page 158](#) pour des détails.

Par mesure de sécurité, l'implémentation Linux du réseau interne n'autorise que les VMs en fonction sous le même utilisateur à établir un réseau interne.

6.6 Réseau Host-only

Le réseau Host-only est un autre mode réseau qui a été ajouté à la version 2.2 de VirtualBox. On peut le voir comme un mode hybride entre les modes réseaux bridgé et interne : comme en réseau bridgé, les machines virtuelles peuvent se parler entre elles et avec l'hôte comme si elles étaient connectées à un switch ethernet physique. Au contraire, comme avec un réseau interne, il faut une interface réseau physique et les machines virtuelles ne peuvent pas parler au monde extérieur à l'hôte puisqu'elles ne sont pas connectées à une interface réseau physique.

Quand on utilise le mode réseau host-only, VirtualBox crée une nouvelle interface logicielle sur l'hôte qui apparaît alors à côté vos interfaces réseaux existantes. En d'autres termes, alors que le réseau bridgé et que l'interface physique existante est utilisée pour y attacher des machines virtuelles, avec le réseau host-only, une nouvelle interface "loopback" est créée sur l'hôte. Et alors qu'avec le réseau interne, le trafic entre les machines virtuelles n'est pas visible, le trafic sur l'interface "loopback" de l'hôte peut être intercepté.

Le réseau Host-only est particulièrement utile pour les applicatifs virtuels préconfigués où plusieurs machines virtuelles sont groupées et conçues pour collaborer. Par exemple, une machine virtuelle peut contenir un serveur web et une deuxième une base de données, et comme

elles sont faites pour se parler, l'applicatif peut demander à VirtualBox de définir un réseau host-only pour les deux. Un deuxième réseau (bridgé) connecterait alors le serveur web au monde extérieur pour offrir des données, mais le monde extérieur ne peut pas se connecter à la base de données.

Pour passer l'interface réseau d'une machine virtuelle en mode "host only" :

- soit allez sur l'onglet "Réseau" de la boîte de dialogue des paramètres de la machine virtuelle dans l'interface graphique et sélectionnez "réseau host-only", soit
- en ligne de commandes, taper `VBoxManage modifyvm "nom VM" --nic<x> hostonly` ; voir chapitre 8.8, *VBoxManage modifyvm*, page 130 pour les détails.

Pour le réseau host-only, comme avec le réseau interne, vous pouvez trouver utile le serveur DHCP construit dans VirtualBox. Il peut être activé puis gérer les adresses IP dans le réseau host-only, puisque sans cela, vous devriez configurer toutes les adresses IP de manière statique.

- Dans l'interface graphique de VirtualBox, vous pouvez configurer tous ces éléments dans les paramètres globaux via "Fichier" -> "Paramètres" -> "Réseau", qui liste tous les réseaux host-only qui sont actuellement utilisés. Cliquez sur le nom du réseau puis sur le bouton "Éditer" à droite, et vous pouvez modifier les paramètres de l'adaptateur et du DHCP.
- Sinon, vous pouvez utiliser `VBoxManage dhcpserver` en ligne de commandes ; voir chapitre 8.35, *VBoxManage dhcpserver*, page 158 pour des détails.

Note: Sur les hôtes Linux et Mac OS X, le nombre d'interfaces host-only est limité à 128. Il n'y a pas de telles limites sur les hôtes Solaris et Windows.

6.7 Réseau en tunnel UDP

Ce mode réseau permet d'interconnecter des machines virtuelles qui fonctionnent sur des hôtes différents.

Techniquement, cela se fait en encapsulant des frames Ethernet envoyés ou reçus par la carte réseau de l'invité dans des datadrams UDP/IP, et en les envoyant via n'importe quel réseau disponible sur l'hôte.

Le mode Tunnel UDP a trois paramètres :

Port source UDP Le port sur lequel écoute l'hôte. Les datagrams arrivant sur ce port depuis n'importe quelle adresse source seront redirigés vers la partie réceptrice de la carte réseau invitée.

Adresse de destination L'adresse IP de l'hôte cible des données transmises.

Port de destination UDP Le numéro du port sur lequel sont envoyées les données transmises.

Quand on interconnecte deux machines virtuelles sur deux hôtes différents, leurs adresses IP doivent être échangées. Sur un seul hôte, les ports UDP source et de destination doivent être échangés.

Dans l'exemple suivant, l'hôte 1 utilise l'adresse IP 10.0.0.1 et l'hôte 2 utilise l'adresse IP 10.0.0.2. La configuration en ligne de commandes :

```
VBoxManage modifyvm "VM 01 on host 1" --nic<x> generic
VBoxManage modifyvm "VM 01 on host 1" --nicgenericdrv<x> UDPTunnel
VBoxManage modifyvm "VM 01 on host 1" --nicproperty<x> dest=10.0.0.2
VBoxManage modifyvm "VM 01 on host 1" --nicproperty<x> sport=10001
VBoxManage modifyvm "VM 01 on host 1" --nicproperty<x> dport=10002
```

et

```
VBoxManage modifyvm "VM 02 on host 2" --nic<y> generic
VBoxManage modifyvm "VM 02 on host 2" --nicgenericdrv<y> UDPTunnel
VBoxManage modifyvm "VM 02 on host 2" --nicproperty<y> dest=10.0.0.1
VBoxManage modifyvm "VM 02 on host 2" --nicproperty<y> sport=10002
VBoxManage modifyvm "VM 02 on host 2" --nicproperty<y> dport=10001
```

Bien entendu, vous pouvez toujours interconnecter deux machines virtuelles sur le même hôte en paramétrant le paramètre Adresse de destination sur 127.0.0.1 sur les deux. Cela agira de la même façon que le “réseau interne” dans ce cas, cependant l’hôte peut voir le trafic réseau, ce qui ne pourrait pas être le cas dans un réseau interne normal.

Note: Sur les hôtes basés sur Unix (comme Linux, Solaris, Mac OS X), il n'est pas possible de sonder les portss inférieurs à 1024 pour des applications non lancées par root . Il s'en suit que si vous essayez de configurer un tel port source UDP, la VM refusera de démarrer.

6.8 Réseau VDE

Virtual Distributed Ethernet (VDE³) est une infrastructure réseau flexible et virtuelle, qui couvre plusieurs hôtes d'une manière sécurisée. Elle permet de basculer entre L2/L3, y compris l'émulation du protocole spanning-tree, des VLANs et de WAN. C'est une partie optionnelle de VirtualBox qui n'est incluse que dans le code source.

Les blocs à construire de base de l'infrastructure sont les switches VDE, les prises VDE et les fils VDE qui inter-connectent les switches.

Le pilote VDe de VirtualBox prend un paramètre :

Réseau VDE Le nom de la socket du switch du réseau VDE à laquelle la VM sera connectée.

L'exemple basique suivant montre la manière de connecter une machine virtuelle à un switch VDE :

1. Créez un switch VDE :

```
vde_switch -s /tmp/switch1
```

2. Configuration en ligne de commandes :

```
VBoxManage modifyvm "nom VM" --nic<x> generic
VBoxManage modifyvm "nom VM" --nicgenericdrv<x> VDE
```

Pour se connecter automatiquement à un port du switch affecté, utilisez :

```
VBoxManage modifyvm "nom VM" --nicproperty<x> network=/tmp/switch1
```

Pour se connecter à un port du switch spécifique <n>, utilisez :

```
VBoxManage modifyvm "nom VM" --nicproperty<x> network=/tmp/switch1[<n>]
```

La dernière option est utile pour les VLANs.

3. Éventuellement, reliez le port du switch VDE et le VLAN : (à partir de la ligne de commande du switch)

```
vde$ vlan/create <VLAN>
```

³VDE est un projet développé par Renzo Davoli, Professeur associé à l'Université de Bologne, Italie.

```
vde$ port/setvlan <port> <VLAN>
```

VDE n'est disponible sur les hôtes Linux et FreeBSD que si le logiciel VDE est la bibliothèque supplément VDE du projet VirtualSquare sont installées sur le système hôte⁴. Pour plus d'informations sur le paramétrage de réseaux VDE, merci de voir la documentation accompagnant le logiciel.⁵

6.9 Limiter la bande passante des E/S réseaux

À partir de la version 4.2, VirtualBox permet de limiter la bande passante maximum utilisée pour la transmission réseau. Plusieurs adaptateurs réseaux d'une VM peuvent partager les limites des groupes de bande passante. Il est possible d'avoir plus d'une limite.

Note: VirtualBox ne gère le trafic de la VM que dans le sens de la transmission, en faisant attendre les paquets à envoyer par les machines virtuelles. Il ne limite pas le trafic reçu par les machines virtuelles.

On configure les limites avec VBoxManage. L'exemple ci-dessous crée **ÚJ** groupe de bande passante appelé "Limit", paramètre la limite à 20 Mo/s et affecte le groupe au premier et au deuxième adaptateurs de la VM :

```
VBoxManage bandwidthctl "nom VM" add Limit --type network --limit 20m
VBoxManage modifyvm "nom VM" --nicbandwidthgroup1 Limit
VBoxManage modifyvm "nom VM" --nicbandwidthgroup2 Limit
```

Tous les adaptateurs d'un groupe partagent la limite de la bande passante, ce qui veut dire que dans l'exemple ci-dessus, la bande passante des deux adaptateurs associés ne peut jamais dépasser 20 Mo/s. Par contre, si un adaptateur n'a pas besoin de bande passante, l'autre peut utiliser le reste de bande passante de son groupe.

On peut modifier les limites de chaque groupe pendant que la VM est en fonction, les changements étant répercutés immédiatement. L'exemple ci-dessous montre le passage de la limite du groupe créé dans l'exemple ci-dessus à 100 Ko/s :

```
VBoxManage bandwidthctl "nom VM" set Limit --limit 100k
```

Pour désactiver complètement l'encadrement du premier adaptateur de la VM, utilisez la commande suivante :

```
VBoxManage modifyvm "nom VM" --nicbandwidthgroup1 none
```

Il est également possible de désactiver l'encadrement de tous les adaptateurs affectés à un groupe de bande passante alors que la VM est en fonction, en spécifiant la limite zéro pour le groupe. Par exemple, pour le groupe de bande passante nommé "Limit", utilisez :

```
VBoxManage bandwidthctl "nom VM" set Limit --limit 0
```

⁴Pour les hôtes Linux, la bibliothèque partagée libvdeplug.so doit être disponible dans le chemin de recherche des bibliothèques partagées

⁵http://wiki.virtualsquare.org/wiki/index.php/VDE_Basic_Networking.

6.10 Améliorer les performances réseaux

VirtualBox offre une variété d'adaptateurs réseaux virtuels qu'on peut attacher au réseau de l'hôte d'un certain nombre de manières. Selon les types d'adaptateurs et d'attachments utilisés, les performances réseaux seront différentes. Dans une logique de performances, l'adaptateur réseau *virtio* est préférable aux adaptateurs *Intel PRO/1000* émulés, préférables eux-mêmes à la famille d'adaptateurs *PCNet*. Tant les adaptateurs *virtio* que *Intel PRO/1000* profitent de la segmentation et de l'offloading de vérification de somme. La segmentation offloading est essentielle pour de hautes performances car elle permet moins de changements de contextes, augmentant drastiquement les tailles des paquets croisés entre VM/boundary hôte.

Note: Ni les pilotes *virtio*, ni ceux *Intel PRO/1000* de Windows XP ne supportent la segmentation offloading. Donc, les invités Windows XP n'atteignent jamais les mêmes vitesses de transmission que les autres types d'invités. Reportez-vous à la base MS Knowledge article 842264 pour des informations supplémentaires.

Trois types d'attachments : *interne*, *bridgé* et *host-only*, ont des performances presqu'identiques, le type *internal* étant légèrement plus rapide et utilisant moins de cycles processeur puisque les paquets ne vont jamais dans la pile réseau de l'hôte. L'attachment *NAT* est le plus lent (et le plus sûr) de tous les types d'attachment car il fournit une traduction d'adresse réseau. L'attachment du pilote générique est spécial et ne peut pas être considéré comme une alternative à d'autres types d'attachments.

Le nombre de processeurs affectés à la VM n'améliore pas les performances et, dans certains cas, cela peut les réduire du fait d'une concurrence dans l'invité.

Voici un petit résumé des choses à vérifier afin d'améliorer les performances réseau :

1. Si possible utilisez l'adaptateur réseau *virtio*, ou utilisez un des adaptateurs *Intel PRO/1000* ;
2. Utilisez l'attachment *bridgé* plutôt que *NAT* ;
3. Assurez-vous que la segmentation offloading est activée dans l'OS invité. En général, elle sera activée par défaut. Vous pouvez vérifier et modifier les paramètres d'offloading en utilisant la commande *ethtool* dans les invités Linux.

7 Machines virtuelles distantes

7.1 Affichage distant (VRDP support)

VirtualBox peut afficher les machines virtuelles à distance, ce qui signifie qu'une machine virtuelle peut s'exécuter sur une machine même si la machine sera affichée sur un deuxième ordinateur, et la machine sera contrôlée également à partir de là, comme si la machine virtuelle fonctionnait sur ce deuxième ordinateur.

Pour une flexibilité maximum, à partir de VirtualBox 4.0, VirtualBox implémente l'affichage distant d'une machine par une interface d'extension générique, le VirtualBox Remote Desktop Extension (VRDE). Le paquet libre de base VirtualBox ne fournit pas cette interface, tandis que les implémentations peuvent être offerts par des tiers avec les paquets d'extension de VirtualBox qui doivent être installés séparément du paquet. Voir chapitre [1.5, Installer VirtualBox et les packs d'extension](#), page [17](#) pour plus d'informations.

Oracle fournit un support pour le **VirtualBox Remote Display Protocol (VRDP)** dans ce paquet d'extension de VirtualBox. Tuand on installe ce paquet, les versions de VirtualBox et 4.0 et supérieur supportent VRDP de la même façon que les versions binaires (non libre)) de VirtualBox avant 4.0.

VRDP est une extension rétro-compatible au Remote Desktop Protocol (RDP) de Microsoft. En général, le rafraîchissement vidéo et du son sont envoyés de la machine distante au client, tandis que les événements clavier et souris sont renvoyés. Il s'en suit que vous utilisez n'importe quel client RDP standard pour contrôler la VM distante.

Même quand l'extension est installée, le serveur VRDP est désactivé par défaut. On peut l'activer facilement pour chaque VM, soit dans le gestionnaire VirtualBox des paramètres d'“Affichage” (voir chapitre [3.5, Paramètres d'affichage](#), page [55](#)), soit avec)s VBoxManage:

```
VBoxManage modifyvm "nom VM" --vrde on
```

Si vous utilisez VBoxHeadless (décris plus loin en détails), le support VRDP sera automatiquement activé puisque VBoxHeadless n'a aucun autre moyen de sortie.

7.1.1 Visualiseurs RDP tiers classiques

Comme VRDP est rétro-compatiblavec RDP, vous pouvez utiliser n'importe quel visualiseur RDP standard pour vous connecter à une machine virtuelle distante (des exemples arrivent ci-dessous). Pour que cela fonctionne, vous devez spécifier **l'adresse IP** de votre système *hôte* (pas celle de la machine virtuelle !) l'adresse du serveur auquel se connecter, ainsi que le **numéro du port** qu'utilise le serveur RDP.

Par défaut, VRDP utilise le port TCP 3389. Vous devrez changer de port par défaut si vous exécutez plus d'un serveur VRDP car le port ne peut être utilisé que par un serveur en même temps ; il se peut aussi que vous deviez le modifier sur les hôtes Windows car le port par défaut pourrait être déjà utilisé par le serveur RDP construit dans Windows lui-même. Les ports 5000 à 5050 ne sont généralement pas utilisés il pourraient être un bon choix.

On peut changer le port soit dans les paramètres “Affichage” de l'interface graphique, soit avec l'option `--vrdeport` de la commande `VBoxManage modifyvm`. Vous pouvez spécifier une liste séparée par des virgules de ports ou de plages de ports. Utilisez un tiret entre deux numéros de ports pour spécifier une plage. Le serveur VRDP sondera **un** des ports disponibles de la liste spécifiée. Par exemple, `VBoxManage modifyvm "nom VM" --vrdeport 5000,5010-5012`

configurera le serveur pour sonder un des ports entre 5000, 5010, 5011 ou le 5012. Voir chapitre [8.8, VBoxManage modifyvm](#), page [130](#) pour des détails.

Vous pouvez rechercher le port utilisé au final par une VM en fonction soit avec la commande `VBoxManage showvminfo`, soit dans l'interface graphique de l'onglet “En cours d'exécution” de la “boîte de dialogue d'informations session”, accessible par le menu “Machine” de la fenêtre de la VM.

Voici des exemples de visualiseurs RDP les plus courants :

- Sur Windows, vous pouvez utiliser le Microsoft Terminal Services Connector (`mstsc.exe`) inclu dans Windows. Vous pouvez le lancer depuis la boîte de dialogue “Exécuter” (appuyez sur la touche Windows et “R”) en tapant “`mstsc`”. Vous pouvez également le trouver dans “Démarrer” -> “Tous les Programmes” -> “Accessoires” -> “Connexion bureau distant”. Si vous utilisez la boîte de dialogue “Exécuter”, vous pouvez y entrer des options directement :

```
mstsc 1.2.3.4[:3389]
```

Remplacez “1.2.3.4” par l'adresse IP de l'hôte et 3389 par un autre port si nécessaire.

Note: Lors d'une connexion à `localhost` pour tester la connexion, les adresses `localhost` et `127.0.0.1` pourraient ne pas fonctionner en utilisant `mstsc.exe`. Par contre, l'adresse `127.0.0.2[:3389]` doit être utilisée.

- Sur les autres systèmes, vous pouvez utiliser le programme libre standard `rdesktop`. Celui-ci est inclus avec la plupart des distributions Linux, mais VirtualBox est fourni avec une variante modifiée de `rdesktop` pour un support de l'USB à distance (voir chapitre [7.1.4, USB distant](#), page [113](#) ci-dessous).

Avec `rdesktop`, utilisez une ligne telle que celle-ci :

```
rdesktop -a 16 -N 1.2.3.4:3389
```

Comme indiqué pour le visualiseur Microsoft ci-dessus, remplacez “1.2.3.4” par l'adresse IP de l'hôte et 3389 par un autre port si nécessaire. L'option `-a 16` demande que la résolution soit de 16 bits par pixel, ce qui est recommandé. (Pour de meilleures performances, après l'installation du système d'exploitation invité, vous devriez définir sa profondeur d'affichage des couleurs à la même valeur). L'option `-N` permet d'utiliser les touches du pavé numérique.

- Si vous lancez le bureau KDE, vous pourriez préférer `krdc`, le visualiseur RDP de KDE. La ligne de commande ressemblerait à ceci :

```
krdc rdp://1.2.3.4[:3389]
```

De nouveau, remplacez “1.2.3.4” par l'adresse IP de l'hôte et 3389 par un port différent si nécessaire. Le champ “`rdp://`” est requis avec `krdc` pour qu'il passe en mode RDP.

- Avec les clients Sun Ray thin, vous pouvez utiliser `utts`, qui fait partie du paquet Sun Ray Windows Connector. Voir la documentation correspondante pour des détails.

7.1.2 VBoxHeadless, le serveur de bureau distant

Si toutes les VM lancées dans le gestionnaire VirtualBox sont capables de lancer des machines virtuelles à distance, ce n'est pas pratique de lancer l'interface graphique full-fledged si vous ne voulez jamais avoir de VMs affichées en local au premier plan. En particulier, si vous lancez un serveur matériel dont le seul objectif est d'héberger des VMs et où toutes les VMs sont supposées se lancer à distance via VRDP, il est inutile d'avoir une interface graphique quelconque sur le serveur – surtout que sur un hôte Linux ou Solaris, le gestionnaire VirtualBox est fourni avec les

dépendances des bibliothèques Qt et SDL. C'est un inconvénient si ne voulez pas du tout avoir de système X Window sur votre serveur.

VirtualBox est donc fourni avec une autre interface appelée `VBoxHeadless`, qui ne produit aucune sortie visible sur l'hôte, mais qui produit des données VRDP. Cette interface n'a aucune dépendance du système X Window sur Linux et des hôtes Solaris.¹

Pour démarrer une machine virtuelle avec `VBoxHeadless`, vous avez trois options :

- Vous pouvez utiliser

```
VBoxManage startvm "nom VM" --type headless
```

L'option supplémentaire `--type` amène VirtualBox à utiliser `VBoxHeadless` en tant qu'interface avec le moteur de virtualisation interne plutôt que l'interface Qt.

- Une alternative est d'utiliser `VBoxHeadless` directement, comme suit :

```
VBoxHeadless --startvm <uuid|name>
```

Cette manière de démarrer la VM aide au dépannage des problèmes signalés par `VBoxManage startvm ...` car vous pouvez voir parfois des messages d'erreur plus détaillés, surtout pour les échecs intervenant avant que l'exécution de la VM ne commence. Dans des situations normales, on préfère `VBoxManage startvm` car il lance la VM directement comme tâche de fond, ce qui doit se faire explicitement lors du démarrage direct de `VBoxHeadless`.

- L'autre alternative est de démarrer `VBoxHeadless` à partir de l'interface graphique du gestionnaire de VirtualBox, en maintenant appuyée la touche Majuscule au démarrage de la machine.

Remarquez que quand vous utilisez `VBoxHeadless` pour démarrer une VM, comme le serveur headless n'a aucun autre moyen d'affichage, le serveur VRDP sera *toujours* activé, indépendamment du fait que vous ayez activé le serveur VRDP dans les paramètres de la VM ou pas. Si vous n'en voulez pas (par exemple parce que vous voulez accéder à la VM via `ssh only`), démarrez la VM comme ceci :

```
VBoxHeadless --startvm <uuid|name> --vrde off
```

Pour activer le serveur VRDP en fonction de la configuration de la VM, comme le feraient les autres interfaces, utilisez ceci :

```
VBoxHeadless --startvm <uuid|name> --vrde config
```

Si vous démarrez la VM avec `VBoxManage startvm ...`, les paramètres de configuration de la VM sont toujours utilisés.

7.1.3 Pas à pas : créer une machine virtuelle sur un serveur headless

Les instructions suivantes peuvent vous donner une idée de la façon de créer une machine virtuelle sur un serveur headless via une connexion réseau. Nous allons créer une machine virtuelle, établir une connexion RDP et installer un système d'exploitation invité – tout ceci sans devoir toucher au serveur headless. Tout ce dont vous avez besoin est ce qui suit :

1. VirtualBox sur une machine serveur avec un système d'exploitation hôte supporté. La pack d'extension de VirtualBox du serveur VRDP doit être installé (voir la section précédente). Pour l'exemple suivant, nous supposerons un serveur Linux.

¹Avant VirtualBox 1.6, le serveur headless s'appelait `VBoxVRDP`. Toujours pour des raisons de rétro-compatibilité, l'installation de VirtualBox installe toujours un exécutable avec ce nom.

2. Un fichier ISO accessible depuis le serveur, contenant les données d'installation du système invité pour installer (nous supposons que c'est Windows XP dans l'exemple suivant).
3. Une connexion en terminal à l'hôte par lequel vous pouvez accéder à une ligne de commande (par exemple via ssh).
4. Un visualiseur RDP sur le client distant ; voir chapitre [7.1.1, Visualiseurs RDP tiers classiques](#), page [109](#) ci-dessus pour des exemples.

Remarquez encore que sur la machine serveur, comme nous n'allons utiliser que le serveur headless, ni Qt ni SDL ni le système X Window ne seront nécessaires.

1. Sur le serveur headless, créez une nouvelle machine virtuelle :

```
VBoxManage createvm --name "Windows XP" --ostype WindowsXP --register
```

Remarquez que si vous ne spécifiez pas `--register`, vous devrez utiliser manuellement la commande `registervm` plus tard.

Remarquez aussi que vous n'avez pas besoin de spécifier `--ostype`, mais en le faisant, cela sélectionne des valeurs par défaut appropriées pour certains paramètres de la VM, par exemple la taille de la RAM et le type de périphérique réseau virtuel. Pour avoir la liste complète des systèmes d'exploitation supportés, vous pouvez utiliser

```
VBoxManage list ostypes
```

2. Assurez-vous que les paramètres de la VM sont adaptés au système d'exploitation invité que nous allons installer. Par exemple :

```
VBoxManage modifyvm "Windows XP" --memory 256 --acpi on --boot1 dvd --nic1 nat
```

3. Créez un disque dur virtuel pour la VM (dans ce cas, 10Go de taille) :

```
VBoxManage createhd --filename "WinXP.vdi" --size 10000
```

4. Ajoutez un contrôleur IDE à la nouvelle VM :

```
VBoxManage storagectl "Windows XP" --name "IDE Controller" --add ide --controller PIIX4
```

5. Mettez le fichier VDI créé ci-dessus comme premier disque dur virtuel de la nouvelle VM :

```
VBoxManage storageattach "Windows XP" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium "WinXP.vdi"
```

6. Attachez le fichier ISO contenant l'installateur du système d'exploitation que vous voudrez installer plus tard sur la machine virtuelle, pour que la machine puisse démarrer dessus :

```
VBoxManage storageattach "Windows XP" --storagectl "IDE Controller" --port 0 --device 1 --type dvddrive --medium /full/path/to/iso.iso
```

7. Démarrez la machine virtuelle en utilisant VBoxHeadless:

```
VBoxHeadless --startvm "Windows XP"
```

Si tout s'est bien passé, vous devriez voir un message de copyright. Si vous êtes renvoyé en ligne de commande, quelque chose n'a pas fonctionné.

8. Sur la machine client, ouvrez le visualiseur RDP et essayez de vous connecter au serveur (voir chapitre [7.1.1, Visualiseurs RDP tiers classiques](#), page [109](#) ci-dessus pour voir comment utiliser les divers visualiseurs RDP classiques).

Vous devriez voir maintenant le processus d'installation de votre système d'exploitation invité, à distance dans le visualiseur RDP.

7.1.4 USB distant

Une fonction très spéciale du support VRDP de VirtualBox est qu'il supporte les périphériques USB distants également via le réseau. C'est-à-dire que l'invité de VirtualBox en fonction sur l'ordinateur peut accéder à des périphériques USB de l'ordinateur distant sur lequel sont affichées les données VRDP de la même manière que des périphériques USB connectés à l'hôte actuel. Cela permet d'exécuter des machines virtuelles sur un hôte VirtualBox qui agit comme serveur, où un client peut se connecter de n'importe où avec simplement un adaptateur réseau et un dispositif d'affichage capable de lancer un visualiseur RDP. Quand on branche des périphériques USB dans le client, le serveur VirtualBox distant peut y accéder.

Pour ces périphériques USB distants, les mêmes règles de filtres s'appliquent, comme pour d'autres périphériques USB, comme décrit au chapitre [3.10.1, Paramètres USB](#), page [60](#). Tout ce que vous devrez faire est de spécifier "Remote" (ou "Any") en définissant ces règles.

L'accès à des périphériques USB n'est possible que si le client RDP supporte cette extension. Sur les hôtes Linux et Solaris, l'installation de VirtualBox fournit un client VRDP convenable appelé `rdesktop-vrdp`. Les versions récentes de `utisc`, un client taillé pour être utilisé avec des clients Sun Ray thin et qui supporte l'accès aux périphériques USB distants. Les clients RDP pour d'autres plateformes seront fournis dans de futures versions de VirtualBox.

Pour rendre disponible un périphérique USB distant pour une VM, vous devriez démarrer `rdesktop-vrdp` comme suit :

```
rdesktop-vrdp -r usb -a 16 -N my.host.address
```

Remarquez que `rdesktop-vrdp` ne peut accéder à des périphériques USB que par `/proc/bus/usb`. Merci de vous reporter au chapitre [12.7.7, L'USB ne fonctionne pas](#), page [223](#) pour des détails supplémentaires sur la manière de régler correctement les droits. De plus, il est conseillé de désactiver le chargement automatique d'un pilote sur l'hôte, sur l'hôte distant, qui pourrait fonctionner sur des périphériques USB, pour vous assurer que les périphériques seront accessibles au client RDP. Si vous avez bien fait le paramétrage sur l'hôte distant, les événements de branchement/débranchement sont visibles dans le fichier `VBox.log` de la VM.

7.1.5 Authentification RDP

Pour chaque machine virtuelle accessible à distance via RDP, vous pouvez déterminer individuellement si les connexions du client sont authentifiées et comment. Pour cela, utilisez la commande `VBoxManage modifyvm` avec l'option `--vrdeauthtype` ; voir chapitre [8.8, VBoxManage modifyvm](#), page [130](#) pour une présentation générale. Trois méthodes d'authentification sont disponibles :

- La méthode "null" signifie qu'il n'y a pas d'authentification du tout ; n'importe quel client peut se connecter au serveur VRDP et, ainsi, à la machine virtuelle. C'est, bien sûr, très peu sécurisé et cela ne doit être recommandé que sur des réseaux privés.
- La méthode "external" fournit une authentification externe via une bibliothèque d'authentification spéciale. VirtualBox inclut deux bibliothèques d'authentification :
 1. La bibliothèque d'authentification par défaut, `VBoxAuth`, authentifie en fonction des droits de l'utilisateur sur l'hôte. Selon la plateforme hôte, cela signifie :
 - Sur les hôtes Linux, `VBoxAuth.so` authentifie les utilisateurs à partir du système PAM de l'hôte.
 - Sur les hôtes Windows, `VBoxAuth.dll` authentifie les utilisateurs à partir du système WinLogon de l'hôte.
 - Sur les hôtes Mac OS X, `VBoxAuth.dylib` authentifie les utilisateurs à partir du service répertoire de l'hôte.²

²Le support pour Mac OS X a été ajouté dans la version 3.2.

En d'autres termes, la méthode "external" par défaut réalise l'authentification par les comptes utilisateurs existant sur le système hôte. Tout utilisateur ayant des autorisations d'authentification valides est accepté, c'est-à-dire que le nom d'utilisateur n'a pas besoin de correspondre à l'utilisateur qui lance la VM.

2. Une bibliothèque supplémentaire qui s'appelle `VBoxAuthSimple` réalise l'authentification à partir des autorisations configurées dans la section "extradata" du fichier XML des paramètres d'une machine virtuelle. C'est probablement la méthode la plus simple de s'authentifier, qui ne dépend pas de l'invité en fonction et supporté (voir ci-dessous). Les étapes suivantes sont nécessaires :

- a) Activer `VBoxAuthSimple` avec la commande suivante :

```
VBoxManage setproperty vrdeauthlibrary "VBoxAuthSimple"
```

- b) Pour activer la bibliothèque pour une VM en particulier, vous devez passer en authentification externe :

```
VBoxManage modifyvm <vm> --vrdeauthtype external
```

Remplacez `<vm>` par le nom ou l'UUID de la VM.

- c) Vous devrez alors configurer les utilisateurs et les mots de passe en écrivant ces éléments dans les extradata de la machine. Comme le fichier XML des paramètres de la machine, dans lequel il faut écrire le mot de passe à la section "extradata", est un fichier en texte brut, VirtualBox utilise le hachage pour chiffrer les mots de passe. Il faut utiliser la commande suivante :

```
VBoxManage setextradata <vm> "VBoxAuthSimple/users/<utilisateur>" <hash>
```

Remplacez `<vm>` par le nom ou l'UUID de la VM, `<utilisateur>` par le nom d'utilisateur qui devrait être autorisé à se connecter et `<hash>` par le mot de passe chiffré. Par exemple, pour avoir la valeur hachée du mot de passe "secret", vous pouvez utiliser la commande suivante :

```
VBoxManage internalcommands passwordhash "secret"
```

Ceci affichera

```
2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b
```

Vous pouvez alors utiliser `VBoxManage setextradata` pour stocker cette valeur dans la section "extradata" de la machine.

Un exemple des deux ensemble, pour définir le mot de passe de l'utilisateur "john" sur la machine "Ma VM" en "secret", utilisez cette commande :

```
VBoxManage setextradata "Ma VM" "VBoxAuthSimple/users/john"
2bb80d537b1da3e38bd30361aa855686bde0eacd7162fef6a25fe97bf527a25b
```

- Enfin, la méthode d'authentification "guest" réalise l'authentification par un composant spécial fourni avec les suppléments invité ; il s'en suit que l'authentification ne s'effectue pas sur l'hôte mais via les comptes utilisateurs de l'invité.

Cette méthode est actuellement en test et pas encore supportée.

Outre les méthodes décrites ci-dessus, vous pouvez remplacer la méthode d'authentification "external" par défaut par n'importe quel autre module. Pour cela, VirtualBox offre une interface claire qui vous permet d'écrire votre propre module d'authentification. Ceci est décrit en détails dans le manuel de référence du Kit de développement logiciel de VirtualBox ; merci de voir chapitre 11, [Interfaces de programmation de VirtualBox](#), page 207 pour des détails.

7.1.6 Chiffrement RDP

Le chiffrement du flux de données des fonctions RDP se base sur le chiffrage symétrique RC4 (avec des clés jusqu'à 128bit). Les clés RC4 sont remplacées à intervalles réguliers (tous les 4096 paquets).

RDP fournit différentes méthodes d'authentification :

1. Historiquement, on utilisait l'authentification RDP4, avec laquelle le client RDP ne réalise aucun contrôle pour vérifier l'identité du serveur auquel il se connecte. Comme on peut obtenir les autorisations de l'utilisateur en utilisant une attaque "man in the middle" (MITM) (personne initiée), l'authentification RDP4 n'est pas sécurisée et vous ne devriez en général pas l'utiliser.
2. L'authentification RDP5.1 utilise un certificat serveur pour lequel le client possède la clé publique. De cette façon, vous avez la garantie que le serveur possède la clé privée correspondante. Cependant, comme cette clé privée codée en dur a été publique il y a quelques années, l'authentification RDP5.1 n'est pas sécurisée.
3. L'authentification RDP5.2 utilise la sécurité RDP améliorée, ce qui veut dire qu'un protocole de sécurité externe est utilisé pour sécuriser la connexion. RDP4 et RDP5.1 utilisent la sécurité RDP Standard. Le serveur VRDP supporte la sécurité renforcée RDP avec le protocole TLS et, faisant partie du handshake TLS, il envoie le certificat du serveur au client.

La propriété VRDE Security/Method définit la méthode de sécurité souhaitée qui est utilisée pour une connexion. Les valeurs valides sont :

- Negotiate - les connexions sécurisées améliorées (TLS) et RDP standards sont autorisées. La méthode de sécurité se négocie avec le client. C'est le réglage par défaut.
- RDP - seule la sécurité QDP Standard est acceptée.
- TLS - seule la sécurité RDP améliorée est acceptée. Le client doit supporter TLS.

Par exemple la commande suivante autorise un client à utiliser une connexion sécurisée soit par RDP Standard, soit par RDP améliorée :

```
vboxmanage modifyvm "nom VM" --vrdeproperty "Security/Method=negotiate"
```

Si la propriété Security/Method est réglée sur Negotiate ou TLS, le protocole TLS sera automatiquement utilisé par le serveur si le client supporte TLS. Cependant, pour utiliser TLS, le serveur doit posséder le certificat du serveur, la clé privée du serveur et l'autorité du certificat (Certificate Authority (CA)). L'exemple suivant montre comment générer un certificat de serveur.

a) Créez un certificat CA auto-signé :

```
openssl req -new -x509 -days 365 -extensions v3_ca \
             -keyout ca_key_private.pem -out ca_cert.pem
```

b) Générez une clé privée du serveur et une demande de signature :

```
openssl genrsa -out server_key_private.pem
openssl req -new -key server_key_private.pem -out server_req.pem
```

c) Générez le certificat du serveur :

```
openssl x509 -req -days 365 -in server_req.pem \
             -CA ca_cert.pem -CAkey ca_key_private.pem -set_serial 01 -out server_cert.pem
```

Le serveur doit être configuré pour accéder aux fichiers requis :

```
vboxmanage modifyvm "nom VM" \
                     --vrdeproperty "Security/CACertificate=path/ca_cert.pem"
```

```
vboxmanage modifyvm "nom VM" \
                     --vrdeproperty "Security/ServerCertificate=path/server_cert.pem"
```

```
vboxmanage modifyvm "nom VM" \
--vrdeproperty "Security/ServerPrivateKey=path/server_key_private.pem"
```

Comme le client qui se connecte au serveur détermine le type de chiffrement qui sera utilisé, avec rdesktop, le visualiseur RDP de Linux, utilisez lq options -4 ou -5.

7.1.7 Connexions multiples au serveur VRDP

Le serveur VRDP de VirtualBox supporte plusieurs connexions simultanées à une VM en fonction à partir de différents clients. Tous les clients connectés voient la même sortie d'écran et partagent le pointeur de souris et le focus du clavier. Cela revient à ce que plusieurs personnes utilisent le même ordinateur en même temps, chacun leur tour devant le clavier.

La commande suivante active le mode connexions multiples :

```
VBoxManage modifyvm "nom VM" --vrdemulticon on
```

7.1.8 Avoir plusieurs moniteurs distants

Pour accéder à deux ou plusieurs moniteurs de VM distants, vous devez activer le mode multi-connexion VRDP (voir la chapitre [7.1.7, Connexions multiples au serveur VRDP](#), page 116).

Le client RDP peut sélectionner le numéro du moniteur virtuel auquel se connecter en utilisant le paramètre d'identification domain (-d). Si le paramètre se termine par @ suivi d'un numéro, VirtualBox interprète ce numéro comme l'index des écrans. On sélectionne l'écran invité primaire avec @1, le premier écran secondaire avec @2, etc.

Le client Microsoft RDP6 ne nous permet pas de spécifier un nom de domaine distinct. Utilisez plutôt domaine\nom_utilisateur dans le champ Nom d'utilisateur : – par exemple, @2\name. nom doit être renseigné et il doit être le nom utilisé pour vous identifier si le serveur VRDP est configuré pour demander une autorisation. Sinon, vous pouvez utiliser n'importe quel texte comme nom d'utilisateur.

7.1.9 Redirection graphique VRDP

À partir de VirtualBox 3.2, le serveur VRDP peut rediriger les flux graphiques de l'hôte vers le client RDP. Les frames graphiques sont compressés en utilisant l'algorithme JPEG, ce qui permet un ratio de compression plus fort que les méthodes de compression bitmap RDP standards. Il est possible d'augmenter le ratio de compression en diminuant la qualité de l'image.

Le serveur VRDP détecte automatiquement les flux vidéos dans un invité au fur et à mesure que les zones rectangulaires se mettent à jour fréquemment. Il s'en suit que cette méthode fonctionne avec n'importe quel système d'exploitation invité, sans devoir installer de logiciels supplémentaires dans l'invité ; en particulier, les suppléments invité ne sont pas requis.

Par contre, côté client, seul le client de connexion Connexion bureau à distance de Windows 7 supporte actuellement cette fonctionnalité. Si un client ne supporte pas la redirection graphique, le serveur VRDP se rabat sur le rafraîchissement régulier des bitmap.

La commande suivante active la redirection graphique :

```
VBoxManage modifyvm "nom VM" --vrdevideochannel on
```

La qualité de l'image se définit par une valeur entre 10 et 100 pour cent, ce qui représente un niveau de compression JPEG (où les nombres les plus bas signifient des qualités plus faibles mais une compression plus forte). Vous pouvez modifier la qualité en utilisant la commande suivante :

```
VBoxManage modifyvm "nom VM" --vrdevideochannelquality 75
```

7.1.10 Personnalisation du VRDP

Avec VirtualBox 4.0, il est possible de désactiver l'affichage, l'entrée souris et clavier, le son, l'USB distant ou le presse-papier, de façon individuelle dans le serveur VRDP.

Les commandes suivantes modifient les paramètres correspondant du serveur :

```
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableDisplay=1
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableInput=1
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableUSB=1
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableAudio=1
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableClipboard=1
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableUpstreamAudio=1
```

Pour réactiver une fonction, utilisez une commande identique sans l'argument 1. Par exemple :

```
VBoxManage modifyvm "nom VM" --vrdeproperty Client/DisableDisplay=
```

Ces propriétés ont été ajoutées avec VirtualBox 3.2.10. Cependant, dans la série 3.2.x, il était nécessaire d'utiliser les commandes suivantes pour modifier les paramètres :

```
VBoxManage setextradata "nom VM" "VRDP/Feature/Client/DisableDisplay" 1
VBoxManage setextradata "nom VM" "VRDP/Feature/Client/DisableInput" 1
VBoxManage setextradata "nom VM" "VRDP/Feature/Client/DisableUSB" 1
VBoxManage setextradata "nom VM" "VRDP/Feature/Client/DisableAudio" 1
VBoxManage setextradata "nom VM" "VRDP/Feature/Client/DisableClipboard" 1
```

Pour réactiver une fonction, utilisez une commande identique sans l'argument 1. Par exemple :

```
VBoxManage setextradata "nom VM" "VRDP/Feature/Client/DisableDisplay"
```

7.2 Téléportation

À partir de la version 3.1, VirtualBox supporte la “téléportation” – à savoir qu'il peut déplacer une machine sur un réseau d'un hôte VirtualBox à l'autre, pendant que la machine est en fonction. Cela fonctionne indépendamment du système d'exploitation hôte : vous pouvez téléporter des machines virtuelles entre, par exemple, des hôtes Solaris et Mac.

La téléportation exige qu'une machine soit en fonction sur un hôte, qu'on appelle alors la “**source**”. L'hôte sur lequel la machine virtuelle sera téléportée s'appellera alors la “**cible**” ; la machine sur la cible est alors configurée pour attendre la source afin de contacter la cible. L'état en fonction de la machine sera transféré de la source à la cible en un temps nimumum.

La téléportation est possible sur n'importe quel réseau TCP/IP ; la source et la cible n'ont besoin que de s'entendre sur un port TCP/IP spécifié dans les paramètres de téléportation.

Cependant, pour l'instant, il y a quelques prérequis pour que cela fonctionne :

1. Sur l'hôte cible, vous devez configurer une machine virtuelle dans VirtualBox avec exactement les mêmes paramètres matériels que la machine sur la source que vous voulez téléporter. Cela ne s'applique pas aux paramètres purement descriptifs tels que le nom de la VM, mais bien sûr, pour que la téléportation fonctionne, la machine cible doit avoir la même quantité de mémoire et les autres paramètres matériels. Sans cela, la téléportation échouera avec un message d'erreur.
2. Les deux machines virtuelles sur la source et la cible doivent partager le même stockage (images de disques durs et de lecteur amovible et CD/DVD). Cela signifie qu'elles utilisent soit la même cible iSCSI, soit le stockage se trouve quelque part sur le réseau et les deux hôtes y ont accès via NFS ou SMB/CIFS.

Cela veut dire aussi que ni la machine source ni la machine cible ne peuvent avoir d'instantanés.

Puis, suivez les étapes suivantes :

1. Sur l'hôte *cible*, configurez la machine virtuelle pour attendre qu'une demande de téléportation n'arrive quand elle sera démarrée, plutôt que pour essayer de démarrer la machine. Cela se fait avec la commande `VBoxManage` suivante :

```
VBoxManage modifyvm <nomvmcible> --teleporter on --teleporterport <port>
```

où `<nomvmcible>` est le nom de la machine virtuelle de l'hôte cible et `<port>` est un numéro de port TCP/IP à utiliser sur les hôtes source et cible. Par exemple, utilisez le 6000. Pour les détails, voir chapitre [8.8.5, Paramètres de téléportation](#), page [136](#).

2. Démarrez la VM sur l'hôte cible. Vous verrez qu'au lieu de démarrer, elle affichera une boîte de dialogue de progression indiquant qu'elle attend une demande de téléportation.

3. Démarrez la machine sur l'hôte *source* comme d'habitude. Quand elle est en fonction et quand vous voulez la téléporter, lancez la commande suivante sur l'hôte source :

```
VBoxManage controlvm <nomvmsource> teleport --host <hôtecible> --port <port>
```

où `<nomvmsource>` est le nom de la machine virtuelle sur l'hôte source (la machine actuellement en fonction) et `<hôtecible>` est le nom ou l'IP de l'hôte cible où une demande de téléportation par la machine est attendue, et `<port>` doit être le même numéro que celui spécifié dans la commande sur l'hôte cible. Pour les détails, voir chapitre [8.13, VBoxManage controlvm](#), page [140](#).

Pour tester, vous pouvez aussi téléporter des machines sur le même hôte ; dans ce cas, utilisez “localhost” comme nom d'hôte sur l'hôte sourd et cible.

Note: Dans de rares cas, si les processeurs de la source et de la cible sont très différents, la téléportation peut échouer avec un message d'erreur ou la cible peut planter. Cela peut arriver surtout si la VM exécute des logiciels très optimisés pour fonctionner sur un processeur particulier sans vérifier correctement que les certaines fonctions du processeur sont bien présentes. VirtualBox filtre les capacités du processeur présenté au système d'exploitation invité. Les utilisateurs avancés peuvent essayer de restreindre les possibilités de ces processeurs virtuels avec la commande `VBoxManage --modifyvm --cpuid` ; voir chapitre [8.8.5, Paramètres de téléportation](#), page [136](#).

8 VBoxManage

8.1 Introduction

Comme brièvement indiqué au chapitre [1.16, *Interfaces alternatives*](#), page [35](#), VBoxManage est l’interface en ligne de commande de VirtualBox. Avec elle, vous pouvez contrôler complètement VirtualBox depuis la ligne de commandes de votre système d’exploitation hôte. VBoxManage supporte toutes les fonctionnalités auxquelles vous donne accès l’interface graphique, mais il supporte bien plus que ça. Il ouvre vraiment toutes les fonctions du moteur de virtualisation, même celles auxquelles on ne peut pas (encore) accéder en mode graphique.

Vous devrez utiliser la ligne de commande si vous voulez

- utiliser une autre interface que celle graphique (par exemple, VBoxSDL ou le serveur VBox-Headless) ;
- contrôler certains paramètres de configuration plus avancés et expérimentaux d’une VM.

Il faut avoir en tête deux choses en utilisant VBoxManage : D’abord, VBoxManage doit toujours être utilisé avec des “sous-commandes” spécifiques telles que “list” ou “createvm” ou “startvm”. Toutes les sous-commandes supportées par VBoxManage sont décrites en détail à la chapitre [8, *VBoxManage*](#), page [119](#).

Ensuite, la plupart de ces sous-commandes exigent que vous spécifiez une machine virtuelle en particulier après la sous-commande. Il y a deux façons de faire cela :

- Vous pouvez spécifier le nom de la VM comme il s’affiche dans l’interface graphique de VirtualBox. Remarquez que si ce nom contient des espaces, vous devez l’entourer de guillemets), comme cela est toujours le cas avec les arguments d’une ligne de commande contenant des espaces).

Par exemple :

```
VBoxManage startvm "Windows XP"
```

- Vous pouvez spécifier l’UUID, qui est l’identifiant interne unique utilisé par VirtualBox pour parler de la machine virtuelle. En supposant que la VM précitée nommée “Windows XP” porte l’UUID affiché ci-dessous, la commande suivante a le même effet que celle précédente :

```
VBoxManage startvm 670e746d-abea-4ba6-ad02-2a3b043810a5
```

Vous pouvez taper `VBoxManage list vms` pour voir toutes les VMs enregistrées ; listées avec leurs paramètres, y compris leurs noms respectifs et leurs UUIDs.

Des exemples classiques de la façon de contrôler VirtualBox depuis la ligne de commande apparaissent ci-dessous :

- Pour créer une nouvelle machine virtuelle en ligne de commandes et l’enregistrer immédiatement avec VirtualBox, utilisez `VBoxManage createvm` avec l’option `--register`,¹ comme ceci :

¹Pour des détails, voir chapitre [8.7, *VBoxManage createvm*](#), page [130](#).

```
$ VBoxManage createvm --name "SUSE 10.2" --register
VirtualBox Command Line Management Interface Version 4.2.51
(C) 2005-2013 Oracle Corporation
All rights reserved.

Virtual machine 'SUSE 10.2' is created.
UUID: c89fc351-8ec6-4f02-a048-57f4d25288e5
Settings file: '/home/nomutilisateur/.config/VirtualBox/Machines/SUSE 10.2/SUSE 10.2.xml'
```

Comme vous pouvez le voir dans la sortie ci-dessus, une nouvelle machine virtuelle a été créée avec le nouvel UUID et un nouveau fichier XML de paramètres.

- Pour afficher la configuration d'une VM en particulier, utilisez `VBoxManage showvminfo` ; voir chapitre [8.5, *VBoxManage showvminfo*](#), page [128](#) pour des détails et un exemple.
- Pour modifier les paramètres pendant qu'une VM est éteinte, utilisez `VBoxManage modifyvm`, comme ceci :

```
VBoxManage modifyvm "Windows XP" --memory "512MB"
```

Pour des détails, voir chapitre [8.8, *VBoxManage modifyvm*](#), page [130](#).

- Pour modifier la configuration du stockage (par exemple pour ajouter un contrôleur de stockage et ensuite un disque virtuel), utilisez `VBoxManage storagectl` et `VBoxManage storageattach` ; voir chapitre [8.19, *VBoxManage storagectl*](#), page [145](#) et chapitre [8.18, *VBoxManage storageattach*](#), page [142](#) pour les détails.
- Pour contrôler le comportement de la VM, utilisez :
 - Pour démarrer une VM éteinte, utilisez `VBoxManage startvm` ; voir chapitre [8.12, *VBoxManage startvm*](#), page [139](#) pour les détails.
 - Pour mettre en pause ou sauvegarder une VM en fonction, ou pour modifier certains de ses paramètres, utilisez `VBoxManage controlvm` ; voir chapitre [8.13, *VBoxManage controlvm*](#), page [140](#) pour les détails.

8.2 Aperçu des commandes

Quand vous lancez `VBoxManage` sans paramètres ou si vous entrez une ligne de commande invalide, le diagramme de syntaxe ci-dessous s'affichera. Remarquez que la sortie sera légèrement différente selon la plateforme hôte ; en cas de doute, vérifiez la sortie de `VBoxManage` pour voir les commandes disponibles sur votre hôte en particulier.

Usage:

```
VBoxManage [<general option>] <command>
```

General Options:

```
[-v|--version]           print version number and exit
[-q|--nologo]            suppress the logo
[--settingspw <pw>]      provide the settings password
[--settingspwfile <file>] provide a file containing the settings password
```

Commands:

```
list [--long|-l]          vms|runningvms|ostypes|hostdvd|hostfloppies|
                           bridgedifs|dhcpservers|hostinfo|
                           hostcpuids|hddbackends|hdds|dvds|floppies|
```

```

usbhost|usbfilters|systemproperties|extpacks|
groups

showvminfo
<uuid>|<name> [--details]
[--machinereadable]
<uuid>|<name> --log <idx>

registervm
<filename>

unregistervm
<uuid>|<name> [--delete]

createvm
--name <name>
[--groups <group>, ...]
[--ostype <ostype>]
[--register]
[--basefolder <path>]
[--uuid <uuid>]

modifyvm
<uuid|name>
[--name <name>]
[--groups <group>, ...]
[--ostype <ostype>]
[--memory <memorysize in MB>]
[--pagefusion on|off]
[--vram <vramsize in MB>]
[--acpi on|off]
[--ioapic on|off]
[--pae on|off]
[--hpet on|off]
[--hwvirtex on|off]
[--hwvirtexexcl on|off]
[--nestedpaging on|off]
[--largepages on|off]
[--vtxvpid on|off]
[--synthcpu on|off]
[--cpuidset <leaf> <eax> <ebx> <ecx> <edx>]
[--cpuidremove <leaf>]
[--cpuidremoveall]
[--hardwareuuid <uuid>]
[--cpus <number>]
[--cpuhotplug on|off]
[--plugcpu <id>]
[--unplugcpu <id>]
[--cpuexecutioncap <1-100>]
[--rtcuseutc on|off]
[--monitorcount <number>]
[--accelerate3d on|off]
[--firmware bios|efi|efi32|efi64]
[--chipset ich9|piix3]
[--bioslogofadein on|off]
[--bioslogofadeout on|off]
[--bioslogodisplaytime <msec>]
[--bioslogologopath <imagepath>]
[--biosbootmenu disabled|menuonly|messageandmenu]
[--biossystemtimeoffset <msec>]
[--biospxeedebug on|off]
[--boot<1-4> none|floppy|dvd|disk|net>]
[--nic<1-N> none|null|nat|bridged|intnet|
  generic]
[--nictype<1-N> Am79C970A|Am79C973]
[--cableconnected<1-N> on|off]
[--nictrace<1-N> on|off]
[--nictracefile<1-N> <filename>]
[--nicproperty<1-N> name=[value]]
[--nicspeed<1-N> <kbps>]
[--nicbootprio<1-N> <priority>]
[--nicpromisc<1-N> deny|allow-vms|allow-all]

```

```

[--nicbandwidthgroup<1-N> none|<name>]
[--bridgeadapter<1-N> none|<devicename>]
[--intnet<1-N> <network name>]
[--natnet<1-N> <network>|default]
[--nicgenericdrv<1-N> <driver>]
[--natsettings<1-N> [<mtu>],[<socksnd>],
    [<sockrcv>],[<tcp snd>],
    [<tcp rcv>]]
[--natpf<1-N> [<rulename>],tcp|udp,[<hostip>],
    <hostport>,[<guestip>],<guestport>]
[--natpf<1-N> delete <rulename>]
[--nattftpprefix<1-N> <prefix>]
[--nattftpfile<1-N> <file>]
[--nattftpserver<1-N> <ip>]
[--natbindip<1-N> <ip>
[--natdnspassdomain<1-N> on|off]
[--natdnsp proxy<1-N> on|off]
[--natdnshostresolver<1-N> on|off]
[--nataliasmode<1-N> default|[log],[proxyonly],
    [sameports]]
[--macaddress<1-N> auto|<mac>]
[--mouse ps2|usb|usbt ablet
[--keyboard ps2|usb
[--uart<1-N> off|<I/O base> <IRQ>]
[--uartmode<1-N> disconnected|
    server <pipe>|
    client <pipe>|
    file <file>|
    <devicename>]
[--lpt<1-N> off|<I/O base> <IRQ>]
[--lptmode<1-N> <devicename>]
[--guestmemoryballoon <balloonsize in MB>]
[--gueststatisticsinterval <seconds>]
[--audio none|null|dsound|solaudio|oss|
    oss|coreaudio]
[--audiocontroller ac97|hda|sb16]
[--clipboard disabled|hosttoguest|guesttohost|
    bidirectional]
[--draganddrop disabled|hosttoguest]
[--vrde on|off]
[--vrdeextpack default|<name>]
[--vrdeproperty <name=[value]>]
[--vrdeport <hostport>]
[--vrdeaddress <hostip>]
[--vrdeauthtype null|external|guest]
[--vrdeauthlibrary default|<name>]
[--vrdemulticon on|off]
[--vrdereusecon on|off]
[--vrdevideochannel on|off]
[--vrdevideochannelquality <percent>]
[--usb on|off]
[--usbehci on|off]
[--snapshotfolder default|<path>]
[--teleporter on|off]
[--teleporterport <port>]
[--teleporteraddress <address|empty>]
[--teleporterpassword <password>]
[--teleporterpasswordfile <file>|stdin]
[--tracing-enabled on|off]
[--tracing-config <config-string>]
[--tracing-allow-vm-access on|off]
[--autostart-enabled on|off]
[--autostart-delay <seconds>]

clonevm
<uuid>|<name>
[--snapshot <uuid>|<name>]
[--mode machine|machineandchildren|all]

```

```

[--options link|keepallmacs|keepnatmacs|
  keepdisknames]
[--name <name>]
[--groups <group>, ...]
[--basefolder <basefolder>]
[--uuid <uuid>]
[--register]

import
<ovf/ova>
[--dry-run|-n]
[--options keepallmacs|keepnatmacs]
[more options]
(run with -n to have options displayed
for a particular OVF)

export
<machines> --output|-o <name>.<ovf/ova>
[--legacy09|--ovf09|--ovf10|--ovf20]
[--manifest]
[--vsys <number of virtual system>]
  [--product <product name>]
  [--producturl <product url>]
  [--vendor <vendor name>]
  [--vendorurl <vendor url>]
  [--version <version info>]
  [--eula <license text>]
  [--eulafile <filename>]

startvm
<uuid>|<name>...
[--type gui|sdl|headless]

controlvm
<uuid>|<name>
pause|resume|reset|poweroff|savestate|
acpipowerbutton|acpisleepbutton|
keyboardputscancode <hex> [<hex> ...]|
setlinkstate<1-N> on|off |
nic<1-N> null|nat|bridged|intnet|generic
  [<devicename>] |
nictrace<1-N> on|off
nictracefile<1-N> <filename>
nicproperty<1-N> name=[value]
natpf<1-N> [<rulename>],tcp|udp,[<hostip>|,
  <hostport>,[<guestip>],<guestport>
natpf<1-N> delete <rulename>
guestmemoryballoon <balloonsize in MB>
gueststatisticsinterval <seconds>]
usbattach <uuid>|<address> |
usbdetach <uuid>|<address> |
clipboard disabled|hosttoguest|guesttohost|
  bidirectional|
draganddrop disabled|hosttoguest|
vrde on|off |
vrdeport <port> |
vrdeproperty <name=[value]> |
vrdevideochannelquality <percent>
setvideomodehint <xres> <yres> <bpp>
  [[<display>] [<enabled:yes|no>
    [<xorigin> <yorigin>]]] |
screenshotpng <file> [<display>] |
setcredentials <username>
  --passwordfile <file> | <password>
  <domain>
  [--allowlocallogon <yes|no>] |
teleport --host <name> --port <port>
  [--maxdowntime <msec>]
  [--passwordfile <file> |
  --password <password>]
plugcpu <id>

```

```

unplugcpu <id>
cpuexecutioncap <1-100>

discardstate
<uuid>|<name>

adoptstate
<uuid>|<name> <state_file>

snapshot
<uuid>|<name>
take <name> [--description <desc>] [--pause] |
delete <uuid>|<name> |
restore <uuid>|<name> |
restorecurrent |
edit <uuid>|<name>|--current
    [--name <name>]
    [--description <desc>] |
list [--details|--machinereadable]
showvminfo <uuid>|<name>

closemedium
disk|dvd|floppy <uuid>|<filename>
[--delete]

storageattach
<uuid|vmname>
--storagectl <name>
[--port <number>]
[--device <number>]
[--type dvddrive|hdd|fdd]
[--medium none|emptydrive|additions|
    <uuid>|<filename>|host:<drive>|iscsi]
[--mtype normal|writethrough|immutable|shareable|
    readonly|multiattach]
[--comment <text>]
[--setuuid <uuid>]
[--setparentuuid <uuid>]
[--passthrough on|off]
[--tempeject on|off]
[--nonrotational on|off]
[--discard on|off]
[--bandwidthgroup <name>]
[--forceunmount]
[--server <name>|<ip>]
[--target <target>]
[--tport <port>]
[--lun <lun>]
[--encodedlun <lun>]
[--username <username>]
[--password <password>]
[--initiator <initiator>]
[--intnet]

storagectl
<uuid|vmname>
--name <name>
--add ide|sata|scsi|floppy|sas
--controller LSILogic|LSILogicSAS|BusLogic|
    IntelAHCI|PIIX3|PIIX4|ICH6|I82078
[--sataportcount <1-30>]
[--hostiocache on|off]
[--bootable on|off]
[--remove]

bandwidthctl
<uuid|vmname>
add <name> --type disk|network
    --limit <megabytes per second>[k|m|g|K|M|G] |
set <name>
    --limit <megabytes per second>[k|m|g|K|M|G] |
remove <name> |
list [--machinereadable]
(limit units: k=kilobit, m=megabit, g=gigabit,

```

```

K=kilobyte, M=megabyte, G=gigabyte)

showhdinfo      <uuid>|<filename>

createhd        --filename <filename>
                [--size <megabytes>|--sizebyte <bytes>]
                [--diffparent <uuid>|<filename>]
                [--format VDI|VMDK|VHD] (default: VDI)
                [--variant Standard,Fixed,Split2G,Stream,ESX]

modifyhd        <uuid>|<filename>
                [--type normal|writethrough|immutable|shareable|
                 readonly|multiattach]
                [--autoreset on|off]
                [--compact]
                [--resize <megabytes>|--resizebyte <bytes>]

clonehd         <uuid>|<filename> <uuid>|<outputfile>
                [--format VDI|VMDK|VHD|RAW|<other>]
                [--variant Standard,Fixed,Split2G,Stream,ESX]
                [--existing]

convertfromraw   <filename> <outputfile>
                [--format VDI|VMDK|VHD]
                [--variant Standard,Fixed,Split2G,Stream,ESX]
                [--uuid <uuid>]
                stdin <outputfile> <bytes>
                [--format VDI|VMDK|VHD]
                [--variant Standard,Fixed,Split2G,Stream,ESX]
                [--uuid <uuid>]

getextradata    global|<uuid>|<name>
                <key>|enumerate

setextradata    global|<uuid>|<name>
                <key>
                [<value>] (no value deletes key)

setproperty      machinefolder default|<folder> |
                vrdeauthlibrary default|<library> |
                websrvauthlibrary default|null|<library> |
                vrdeextpack null|<library> |
                autostartdbpath null|<folder> |
                loghistorycount <value>

usbfilter        add <index,0-N>
                --target <uuid>|<name>|global
                --name <string>
                --action ignore|hold (global filters only)
                [--active yes|no] (yes)
                [--vendorid <XXXX>] (null)
                [--productid <XXXX>] (null)
                [--revision <IIFF>] (null)
                [--manufacturer <string>] (null)
                [--product <string>] (null)
                [--remote yes|no] (null, VM filters only)
                [--serialnumber <string>] (null)
                [--maskedinterfaces <XXXXXXXX>]

usbfilter        modify <index,0-N>
                --target <uuid>|<name>|global
                [--name <string>]
                [--action ignore|hold] (global filters only)
                [--active yes|no]
                [--vendorid <XXXX>|""]
                [--productid <XXXX>|""]
                [--revision <IIFF>|""]

```

```

[--manufacturer <string>|""]
[--product <string>|""]
[--remote yes|no] (null, VM filters only)
[--serialnumber <string>|""]
[--maskedinterfaces <XXXXXXXX>]

usbfilter
remove <index,0-N>
--target <uuid>|<name>|global

sharedfolder
add <vmname>|<uuid>
--name <name> --hostpath <hostpath>
[--transient] [--readonly] [--automount]

sharedfolder
remove <vmname>|<uuid>
--name <name> [--transient]

debugvm
<uuid>|<name>
dumpguestcore --filename <name> |
info <item> [args] |
injectnmi |
log [--release|--debug] <settings> ... |
logdest [--release|--debug] <settings> ... |
logflags [--release|--debug] <settings> ... |
osdetect |
osinfo |
getregisters [--cpu <id>] <reg>|all ... |
setregisters [--cpu <id>] <reg>=<value> ... |
show [--human-readable|--sh-export|--sh-eval] |
--cmd-set |
<logdbg-settings|logrel-settings>
[[opt] what ...] |
statistics [--reset] [--pattern <pattern>] |
[--descriptions]

metrics
list [*|host|<vmname> [<metric_list>]] |
(comma-separated)

metrics
setup
[--period <seconds>] (default: 1)
[--samples <count>] (default: 1)
[--list]
[*|host|<vmname> [<metric_list>]] |

metrics
query [*|host|<vmname> [<metric_list>]] |

metrics
enable
[--list]
[*|host|<vmname> [<metric_list>]] |

metrics
disable
[--list]
[*|host|<vmname> [<metric_list>]] |

metrics
collect
[--period <seconds>] (default: 1)
[--samples <count>] (default: 1)
[--list]
[--detach]
[*|host|<vmname> [<metric_list>]] |

dhcpserver
add|modify --netname <network_name> |
[--ip <ip_address>
--netmask <network_mask>
--lowerip <lower_ip>
--upperip <upper_ip>]
[--enable | --disable]

```

```

dhcpserver      remove --netname <network_name> |
extpack         install [--replace] <tarball> |
                  uninstall [--force] <name> |
                  cleanup

```

À chaque fois que vous appelez VBoxManage, une seule commande peut être exécutée. Toutefois, une commande peut supporter plusieurs sous-commandes que vous pouvez appeler en une seule fois. Les sections suivantes fournissent des informations de référence détaillées sur les différentes commandes.

8.3 Options générales

- **--version** : affiche la version de cet outil et quitte.
- **--nologo** : supprime l'affichage des informations de logo (utile pour les scripts)
- **--settingspw** : spécifie un mot de passe pour les paramètres
- **--settingspwfile** : spécifie un fichier contenant le mot de passe des paramètres

Le mot de passe des paramètres est utilisé pour certains paramètres ayant besoin d'être stockés et chiffrés pour des raisons de sécurité. Pour le moment, le seul paramètre chiffré est l'initiateur secret iSCSI (voir chapitre 8.18, *VBoxManage storageattach*, page 142 pour les détails). Tant qu'aucun mot de passe des paramètres n'a été spécifié, ces informations sont stockées en **texte brut**. Après avoir utilisé l'option **--settingspw|--settingspwfile** une fois, il faut toujours l'utiliser, sans quoi le paramètre chiffré ne peut pas être chiffré.

8.4 VBoxManage list

La commande **list** donne des informations pertinentes sur votre système et sur les paramètres actuels de VirtualBox.

Les sous-commandes suivantes sont disponibles avec **VBoxManage list** :

- **vms** liste toutes les machines virtuelles actuellement enregistrées avec VirtualBox. Par défaut, elle affiche une liste compacte avec le nom et l'UUID de chaque VM ; si vous spécifiez en plus **--long** ou **-l**, cela affichera une liste détaillée comme avec la commande **showvminfo** (voir ci-dessous).
- **runningvms** liste toutes les machines virtuelles actuellement en fonction avec leurs identifiants uniques (UUIDs) sous la même forme qu'avec **vms**.
- **ostypes** liste les systèmes d'exploitation invités connus aujourd'hui de VirtualBox, ainsi que les identifiants utilisés pour s'y référer avec la commande **modifyvm**.
- **hostdvds**, **hostfloppies**, listent respectivement les lecteurs DVD, amovibles, les interfaces réseaux bridgées et host-only sur l'hôte, ainsi que le nom utilisé pour y accéder depuis VirtualBox.
- **bridgedifs**, **hostonlyifs** et **dhcpservers**, listent respectivement les interfaces réseaux bridgées, host-only et les serveurs DHCP disponibles sur l'hôte. Merci de voir chapitre 6, *Le réseau virtuel*, page 98 pour les détails à leur sujet.
- **hostinfo** affiche des informations sur le système hôte telles que les processeurs, la taille de la mémoire et la version du système d'exploitation.

- **hostcpuids** affiche les paramètres CPUID aux processeurs de l'hôte. Cela peut servir à une analyse plus fine des possibilités de virtualisation de l'hôte.
- **hddbackends** liste les fondations de disque virtuel connues de VirtualBox. Pour chaque format (tels que VDI, VMDK ou RAW), ceci liste les possibilités de la fondation et sa configuration.
- **hdds**, **dvds** et **floppies** vous donnent des informations sur les images de disque virtuel actuellement utilisées par VirtualBox, y compris leurs paramètres, leurs identifiants uniques (UUIDs) associés par VirtualBox et tous leurs fichiers associés. C'est l'équivalent ligne de commandes du gestionnaire de médias virtuels ; voir chapitre [5.3, Le gestionnaire de médias virtuels](#), page [88](#).
- **usbhost** donne des informations sur les périphériques USB attachés à l'hôte, en particulier des informations utiles pour construire des filtres USB et s'ils sont actuellement utilisés par l'hôte.
- **usbfilters** liste les filtres USB globaux enregistrés avec VirtualBox – c'est-à-dire les filtres des périphériques accessibles à toutes les machines virtuelles – et il affiche les paramètres du filtre.
- **systemproperties** affiche des paramètres globaux de VirtualBox, tels que la RAM minimum et maximum de l'invité et la taille du disque dur virtuel, les paramètres du dossier et la bibliothèque d'authentification actuellement utilisée.
- **extpacks** affiche les packs d'extension de VirtualBox actuellement installés ; voir chapitre [1.5, Installer VirtualBox et les packs d'extension](#), page [17](#) et chapitre [8.36, VBoxManage extpack](#), page [159](#) pour plus d'informations.

8.5 VBoxManage showvminfo

La commande **showvminfo** affiche des informations sur une machine en particulier. Ce sont les mêmes informations qu'afficherait **VBoxManage list vms --long** pour toutes les machines virtuelles.

Vous aurez des informations ressemblant à ce qui suit :

```
$ VBoxManage showvminfo "Windows XP"
VirtualBox Command Line Management Interface Version 4.2.51
(C) 2005-2013 Oracle Corporation
All rights reserved.

Name:          Windows XP
Guest OS:       Other/Unknown
UUID:          1bf3464d-57c6-4d49-92a9-a5cc3816b7e7
Config file:    /home/nomutilisateur/.config/VirtualBox/Machines/Windows XP/Windows XP.xml
Memory size:   512MB
VRAM size:     12MB
Number of CPUs: 2
Synthetic Cpu: off
Boot menu mode: message and menu
Boot Device (1): DVD
Boot Device (2): HardDisk
Boot Device (3): Not Assigned
Boot Device (4): Not Assigned
ACPI:          on
IOAPIC:         on
PAE:           on
Time offset:   0 ms
Hardw. virt.ext: on
Hardw. virt.ext exclusive: on
```

```

Nested Paging: on
VT-x VPID: off
State: powered off (since 2009-10-20T14:52:19.000000000)
Monitor count: 1
3D Acceleration: off
2D Video Acceleration: off
Teleporter Enabled: off
Teleporter Port: 0
Teleporter Address:
Teleporter Password:
Storage Controller (0): IDE Controller
Storage Controller Type (0): PIIX4
Storage Controller (1): Floppy Controller 1
Storage Controller Type (1): I82078
IDE Controller (0, 0): /home/user/windows.vdi (UUID: 46f6e53a-4557-460a-9b95-68b0f17d744b)
IDE Controller (0, 1): /home/user/openbsd-cd46.iso (UUID: 4335e162-59d3-4512-91d5-b63e94eebe0b)
Floppy Controller 1 (0, 0): /home/user/floppy.img (UUID: 62ac6ccb-df36-42f2-972e-22f836368137)
NIC 1: disabled
NIC 2: disabled
NIC 3: disabled
NIC 4: disabled
NIC 5: disabled
NIC 6: disabled
NIC 7: disabled
NIC 8: disabled
UART 1: disabled
UART 2: disabled
Audio: disabled (Driver: Unknown)
Clipboard Mode: Bidirectional
VRDE: disabled
USB: disabled

USB Device Filters:
<none>

Shared folders:
<none>

Statistics update: disabled

```

8.6 VBoxManage registervm / unregistervm

La commande `registervm` vous permet d'importer la définition d'une machine virtuelle dans VirtualBox via un fichier XML. La machine ne doit pas entrer en conflit avec une autre déjà enregistrée dans VirtualBox et elle ne peut pas avoir de disques durs ou amovibles attachés. Il est recommandé de mettre le fichier de définition dans le répertoire des machines avant de l'enregistrer.

Note: Lors de la création d'une nouvelle machine virtuelle avec `VBoxManage createvm` (voir ci-dessous), vous pouvez spécifier directement l'option `--register` pour éviter de devoir l'enregistrer séparément.

La commande `unregistervm` désenregistre une machine virtuelle. Si vous spécifiez également `--delete`, les fichiers suivants seront eux aussi automatiquement effacés :

1. tous les fichiers images de disque dur, y compris les fichiers de différenciation, utilisés par la machine et non partagés avec d'autres machines ;
2. les fichiers de l'état sauvegardé créés par la machine s'il y en a (un si la machine était en état "sauvegardé" et un pour chaque instantané en ligne) ;

3. le fichier XML de la machine et ses sauvegardes ;
4. les fichiers journaux de la machine s'il y en a ;
5. le répertoire de la machine s'il est vide après l'effacement de tout ce qui précède.

8.7 VBoxManage createvm

Cette commande crée un nouveau fichier de définition de machine virtuelle XML.

Le paramètre `--name <name>` est requis et doit spécifier le nom de la machine. Ce nom étant utilisé par défaut pour nommer le fichier des paramètres (avec l'extension `.xml`) et le dossier machine (un sous-dossier du dossier `.config/VirtualBox/Machines`), il doit respecter les exigences de votre système d'exploitation hôte en matière de spécifications d'un nom de fichier. Si vous renommez plus tard la VM, le nom du fichier et du dossier seront modifiés automatiquement.

Néanmoins, si vous utilisez l'option `--basefolder <path>`, le dossier de la machine sera appelé `<path>`. Dans ce cas, les noms du fichier et du dossier ne seront pas modifiés si la machine virtuelle est renommée.

Par défaut, cette commande ne fait que créer le fichier XML, sans enregistrer automatiquement la VM avec votre installation de VirtualBox. Pour enregistrer instantanément la VM, utilisez l'option `--register`, ou lancez `VBoxManage registervm` séparément a posteriori.

8.8 VBoxManage modifyvm

Cette commande modifie les propriétés d'une machine virtuelle enregistrée et non en fonction. La plupart des propriétés disponibles avec cette commande correspondent aux paramètres de la VM affichés dans l'interface graphique de VirtualBox dans la boîte de dialogue “Paramètres” de chaque VM ; celles-ci ont été décrites en chapitre 3, [Configurer des machines virtuelles](#), page 48. Néanmoins, certains des paramètres les plus avancés ne sont disponibles que via l'interface VBoxManage.

Ces commandes exigent que la machine soit éteinte (ni en fonction ni en état “sauvegardé”). Certains paramètres de la machine peuvent être aussi modifiés pendant qu'une machine est en fonction ; ces paramètres auront alors une sous-commande correspondante avec la sous-commande `VBoxManage controlvm` (voir chapitre 8.13, [VBoxManage controlvm](#), page 140).

8.8.1 Paramètres généraux

Les paramètres généraux suivants sont disponibles via `VBoxManage modifyvm` :

- `--name <nom>` : Ceci modifie le nom de la VM et renomme éventuellement les fichiers internes virtuels, comme décrit avec `VBoxManage createvm` ci-dessus.
- `--ostype <ostype>` : Ceci spécifie le système d'exploitation supposé fonctionner dans la VM. Pour en apprendre davantage sur les différents identifiants utilisables, utilisez `VBoxManage list ostypes`.
- `--memory <taillemémoire>` : Ceci définit la quantité de RAM, en Mo, que devrait s'affecter à elle-même la machine virtuelle sur l'hôte. Voir les remarques au chapitre 1.7, [Créer votre première machine virtuelle](#), page 19 pour plus d'informations.
- `--vram <taillevram>` : Ceci définit la quantité de RAM de la carte graphique. Voir chapitre 3.5, [Paramètres d'affichage](#), page 55 pour des détails.

- **--acpi on|off ; --ioapic on|off** : Ces deux éléments déterminent si la VM devrait supporter respectivement l'ACPI et l'APIC I/O ; voir chapitre 3.4.1, *Onglet “Carte mère”*, page 52 pour des détails.
- **--hardwareuuid <uuid>** : L'UUID présentée à l'invité via les tables mémoire (DMI/SMBIOS), les propriétés matérielles et invité. Par défaut, c'est le même que l'uuid de la VM. Utile quand vous clonez une VM. La téléportation en tient compte automatiquement.
- **--cpus <nombreprocesseur>** : Ceci définit le nombre de processeurs virtuels de la machine virtuelle (voir chapitre 3.4.2, *Onglet “Processeur”*, page 54). Si le branchement à chaud des processeurs est activé (voir ci-dessous), ceci définit le nombre *maximum* de processeurs virtuels qu'on peut brancher dans les machines virtuelles.
- **--rtcuseutc on|off** : Cette option fait tourner l'horloge en temps réel (RTC) en temps UTC (voir chapitre 3.4.1, *Onglet “Carte mère”*, page 52).
- **--cpuhotplug on|off** : Ceci active le branchement à chaud du processeur. Quand vous l'activez, les processeurs virtuels peuvent être ajoutés et supprimés d'une machine virtuelle pendant qu'elle est en fonction. Voir chapitre 9.5, *Montage de processeur à chaud*, page 170 pour plus d'informations.
- **--plugcpu|unplugcpu <id>** : Si vous avez activé le branchement des processeurs à chaud (voir ci-dessus), ceci ajoute un processeur virtuel aux machines virtuelles (ou en supprime un). <id> spécifie l'index du processeur virtuel à ajouter ou supprimer et doit être un numéro entre 0 et le nombre maximum de processeurs configurés avec l'option --cpus. Le processeur 0 ne peut jamais être supprimé.
- **--cpuexecutioncap <1-100>** : Ce paramètre contrôle le nombre de temps processeur utilisable par un processeur. La valeur 50 implique qu'un seul processeur virtuel peut utiliser jusqu'à 50% d'un seul processeur de l'hôte.
- **--synthcpu on|off** : Ce paramètre détermine si VirtualBox présente ou non un processeur synthétique à l'invité pour permettre une migration en direct entre les systèmes hôtes qui diffèrent de manière significative.
- **--pae on|off** : Ceci active/désactive la PAE (voir chapitre 3.4.2, *Onglet “Processeur”*, page 54).
- **--hpet on|off** : Ceci active/désactive un High Precision Event Timer (HPET) qui peut remplacer les timers du système primitif. C'est désactivé par défaut. Remarquez que Windows ne supporte un HPET qu'à partir de Vista.
- **--hwvirtex on|off** : Ceci active ou désactive l'utilisation des extensions du matériel de virtualisation (Intel VT-x ou AMD-V) dans le processeur de votre système hôte ; voir chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201.
- **--hwvirtexexcl on|off** : Ceci spécifie si VirtualBox utilisera exclusivement les extensions de virtualisation matérielle (Intel VT-x ou AMD-V) dans le processeur de votre système hôte ; voir chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201. Si vous souhaitez partager en même temps ces extensions avec d'autres hyperviseurs, vous devez désactiver ce paramètre. Ceci a des implications négatives en performance.
- **--nestedpaging on|off** : Si la virtualisation matérielle est activée, ce paramètre supplémentaire active ou désactive l'utilisation de la fonction de pagination nested du processeur de votre système hôte ; voir chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201.

- **--largepages on|off** : Si la virtualisation matérielle et la pagination nested sont activées, pour Intel VT-x seulement, vous pouvez obtenir un gain de performances jusqu'à 5% en activant ce paramètre. Il fait utiliser à l'hyperviseur les grandes pages pour réduire l'utilisation du TLB et l'overhead.
- **--vtxvpid on|off** : Si la virtualisation matérielle est activée, pour Intel VT-x seulement, ce paramètre supplémentaire active ou désactive l'utilisation de la fonction TLB taggée (VPID) du processeur de votre système hôte ; voir see chapitre [10.3, Virtualisation matérielle vs. logicielle](#), page [201](#).
- **--accelerate3d on|off** : Ceci active, si vous avez installé les suppléments invité, la disponibilité de l'accélération 3D ; voir chapitre [4.4.1, Accélération 3D matérielle \(OpenGL and Direct3D 8/9\)](#), page [77](#).
- Vous pouvez influencer le logo du BIOS affiché quand une machine virtuelle démarre avec un certain nombre de paramètres. Par défaut, un logo VirtualBox est affiché.
Avec **--bioslogofadein on|off** et **--bioslogofadeout on|off**, vous pouvez déterminer si le logo devrait fade in and out.
Avec **--bioslogodisplaytime <msec>** vous pouvez définir la durée d'affichage du logo en millisecondes.
Avec **--bioslogoimagepath <cheminimage>** vous pouvez, si vous le souhaitez, remplacer l'image affichée, par votre propre logo. L'image doit être un fichier BMP décompressé 256 couleurs.
- **--biosbootmenu disabled|menuonly|messageandmenu** : Ceci spécifie si le BIOS permet à l'utilisateur de sélectionner un périphérique de démarrage temporaire. **menuonly** supprime le message, mais l'utilisateur peut encore appuyer sur F12 pour sélectionner un périphérique de démarrage temporaire.
- **--nicbootprio<1-N> <priorité>** : Ceci spécifie l'ordre dans lequel les NICs sont essayés pour démarrer par le réseau (en utilisant PXE). La priorité est un entier entre 0 à 4. La priorité 1 est la plus élevée, la priorité 4 est la plus faible. La priorité 0, qui est celle par défaut sauf autrement spécifié, est la plus faible.
Remarquez que cette option n'a d'effet que quand le démarrage Intel PXE ROM est utilisé.
- **--boot<1-4> none|floppy|dvd|disk|net** : Ceci spécifie l'ordre d'amorçage pour la machine virtuelle. Il y a quatre "slots", auxquels essaiera d'accéder la VM de 1 à 4, et pour chacun d'eux, vous pouvez définir un périphérique sur lequel la VM devrait essayer de démarrer.
- **--snapshotfolder default|<path>** : Ceci vous permet de spécifier le dossier dans lequel seront stockés les instantanés d'une machine virtuelle.
- **--firmware efi|bios** : Spécifie le firmware utilisé pour démarrer une machine virtuelle en particulier : EFI ou BIOS. N'utilisez EFI que si vous comprenez bien ce que vous faites.
- **--guestmemoryballoon <taille>** définit la taille par défaut de la mémoire du ballon de l'invité c'est-à-dire la mémoire affectée par les suppléments invité de VirtualBox à partir du système d'exploitation invité et rendue à l'hyperviseur pour être réutilisée par d'autres machines virtuelles. <taille> doit être spécifiée en mégaoctets. La taille par défaut est de 0 mégaoctets. Pour les détails, voir chapitre [4.8.1, Faire du ballon avec la mémoire](#), page [83](#).
- **--lptmode<1-N> <Périphérique>** Spécifie le nom du périphérique du port parallèle utilisé par la fonction Port parallèle. Utilisez ceci avant **--lpt**. Cette fonction est spécifique au système d'exploitation.

- **--lpt<1-N> <base E/S> <IRQ>** Spécifie l'adresse E/S du port parallèle et le numéro IRQ utilisés par la fonction Port parallèle. Utilisez ceci après **--lptmod**. L'adresse de base E/S et le numéro IRQ sont les valeurs vues par l'invité, c'est-à-dire les valeurs disponibles dans le gestionnaire de périphériques de l'invité.

8.8.2 Paramètres réseaux

Les paramètres réseaux suivants sont disponibles avec **VBoxManage modifyvm**. Avec tous ces paramètres, le numéro décimal suivant directement le nom de l'option (“1-N” dans la liste ci-dessous) spécifie l'adaptateur réseau virtuel dont vous modifiez les paramètres.

- **--nic<1-N> none|null|nat|bridged|intnet|hostonly|generic** : Avec cela, vous pouvez définir, pour chacune des cartes réseaux virtuelles de la VM, le type de réseau qui devrait être disponible. Ils peuvent ne pas être présents (none), non connectés à l'hôte (null), utiliser le network address translation (nat), le réseau bridgé (bridged) ou communiquer avec d'autres machines virtuelles en utilisant le réseau internal (intnet), le réseau host-only (hostonly), ou accéder à des sous-modes rarement utilisés (generic). Ces options correspondent aux modes décrits en détail au chapitre [6.2, Introduction aux modes réseaux](#), page [99](#).
- **--nictype<1-N> Am79C970A|Am79C973|82540EM|82543GC|82545EM|virtio** : Cela vous permet, pour chacune des cartes réseaux virtuelles de la VM, de spécifier le matériel réseau que VirtualBox présente à l'invité ; voir chapitre [6.1, Matériel réseau virtuel](#), page [98](#).
- **--cableconnected<1-N> on|off** : Cela vous permet de déconnecter temporairement une interface réseau virtuelle comme si vous débranchez un câble réseau d'une vraie carte réseau. Cela pourrait être utile pour reparamétrer certains composants logiciels dans la VM.
- Avec l'option “nictrace”, vous pouvez éventuellement tracer le trafic réseau en l'envoyant vers un fichier pour faire du débogage.
Avec **--nictrace<1-N> on|off**, vous pouvez activer un traçage réseau pour des cartes réseaux en particulier.
Si vous l'activez, vous devez spécifier avec **--nictracefile<1-N> <nomfichier>** le fichier où devrait être journalisée la trace.
- **--bridgeadapter<1-N> none|<nompériphérique>** : Si le réseau bridgé est activé pour une carte réseau virtuelle (voir l'option **--nic** ci-dessus ; sinon ce paramètre n'a aucun effet), utilisez cette option pour spécifier l'interface de l'hôte qu'utilisera l'interface réseau virtuelle donnée. Pour des détails, merci de voir chapitre [6.4, Réseau Bridged](#), page [102](#).
- **--hostonlyadapter<1-N> none|<nompériphérique>** : Si le réseau host-only a été activé pour une carte réseau virtuelle (voir l'option **--nic** ci-dessus ; sinon ce paramètre n'a aucun effet), utilisez cette option spécifier l'interface réseau host-only qu'utilisera l'interface réseau donnée. Pour des détails, merci de voir chapitre [6.6, Réseau Host-only](#), page [104](#).
- **--intnet<1-N> network** : Si le réseau interne a été activé pour une carte réseau virtuelle (voir l'option **--nic** ci-dessus ; sinon ce paramètre n'a aucun effet), utilisez cette option pour spécifier le nom du réseau interne (voir chapitre [6.5, Réseau interne](#), page [103](#)).
- **--macaddress<1-N> auto|<mac>** : Avec cette option, vous pouvez définir l'adresse MAC de la carte réseau virtuelle. Normalement, chaque carte réseau virtuel se voit affecter une adresse aléatoire par VirtualBox Lors de la création de la VM.

- `--nicgenericdrv<1-N> <pilote fondation>` : Si le réseau générique a été activée pour une carte réseau virtuelle (voir l'option `--nic` ci-dessus ; sinon ce paramètre n'a aucun effet), ce mode vous permet d'accéder à des sous-modes réseaux rarement utilisés, tels que le réseau VDE ou le tunnel UDP.
- `--nicproperty<1-N> <nomparam>="paramvaleur"` : Cette option, associée avec "nic-genericdrv", vous permet de passer des paramètres aux fondations réseaux rarement utilisées.

Ces paramètres sont des fondations spécifiques au moteur et sont différentes entre les pilotes de la fondation du tunnel UDP et du VDE. Par exemple, merci de voir chapitre 6.7, *Réseau en tunnel UDP*, page 105.

8.8.2.1 Paramètres du réseau NAT.

Les paramètres de réseau NAT suivants sont disponibles via `VBoxManage modifyvm`. Avec tous ces paramètres, le numéro décimal suivant le nom de l'option ("1-N" dans la liste ci-dessous) spécifie l'adaptateur réseau virtuel que devrait modifier ces paramètres.

- `--natpf<1-N> [<nom>],tcp|udp,[<hostip>],<porthôte>,[<ipinvité>], <portinvité>` : Cette option définit une règle de redirection de port NAT (merci de voir chapitre 6.3.1, *Configurer la redirection de ports avec NAT*, page 100 pour des détails).
- `--natpf<1-N> delete <nom>` : Cette option supprime une règle de redirection de port NAT (merci de voir chapitre 6.3.1, *Configurer la redirection de ports avec NAT*, page 100 pour les détails).
- `--nattftpprefix<1-N> <préfixe>` : Cette option définit un préfixe pour le serveur TFTP intégré, à savoir là où se trouve le fichier de démarrage (merci de voir chapitre 6.3.2, *Démarrer avec PXE avec NAT*, page 101 and chapitre 9.12.2, *Configurer le serveur d'amorçage (prochain serveur) d'une interface réseau NAT*, page 179 pour les détails).
- `--nattftpfile<1-N> <fichierdémarrage>` : Cette option définit le fichier de démarrage TFTP (merci de voir chapitre 9.12.2, *Configurer le serveur d'amorçage (prochain serveur) d'une interface réseau NAT*, page 179 pour des détails).
- `--nattftpserver<1-N> <serveurtftp>` : Cette option définit l'adresse du serveur TFTP sur lequel démarrer (voir chapitre 9.12.2, *Configurer le serveur d'amorçage (prochain serveur) d'une interface réseau NAT*, page 179 pour les détails).
- `--natdnsdomain<1-N> on|off` : Cette option spécifie si le serveur DHCP passe le nom de domaine pour la résolution du nom du réseau.
- `--natdnsproxy<1-N> on|off` : Cette option fait faire du proxy au moteur NAT pour toutes les requêtes DNS de l'invité adressées au serveur DNS de l'hôte (merci de voir chapitre 9.12.5, *Activer le proxy DNS en mode NAT*, page 180 pour des détails).
- `--natdnshostresolver<1-N> on|off` : Cette option fait utiliser par le moteur NAT les mécanismes de résolveur de l'hôte pour gérer les requêtes DNS (merci de voir chapitre 9.12.5, *Activer le proxy DNS en mode NAT*, page 180 pour les détails).
- `--natnatsettings<1-N> [<mtu>],[<socksnd>],[<sockrcv>],[<tcpsnd>], [<tcprcv>]` : Cette option contrôle plusieurs paramètres NAT (merci de voir chapitre 9.12.3, *Peaufiner les tampons TCP/IP pour NAT*, page 179 pour les détails).
- `--nataliasmode<1-N> default|[log],[proxyonly],[sameports]` : Cette option définit le comportement du cœur du moteur NAT : log - active la journalisation, proxyonly - switch les modes d'alias pour rendre NAT transparent, sameports renforce le moteur NAT

pour envoyer des paquets via le même port qu'ils arrivent, default - désactive tous les modes mentionnés ci-dessus. (merci de voir chapitre 9.12.7, *Configurer des aliases pour le moteur NAT*, page 181 pour des détails).

8.8.3 Options du port série, du son, du presse-papier, du bureau distant et de l'USB

Les autres paramètres matériels suivants sont disponibles avec `VBoxManage modifyvm` :

- `--uart<1-N> off|<I/O base> <IRQ>` : Avec cette option, vous pouvez configurer les ports série virtuels de la VM (voir chapitre 3.9, *Ports série*, page 59 pour une introduction).
- `--uartmode<1-N> <arg>` : Ce paramètre contrôle la façon dont VirtualBox se connecte à un port série virtuel donné (on la configurait jadis avec l'option `--uartX`, voir ci-dessus) sur l'hôte sur lequel fonctionne la machine virtuelle. Comme décrit en détail au chapitre 3.9, *Ports série*, page 59, pour chaque port, vous pouvez spécifier `<arg>` comme une des options suivantes :
 - `disconnected` : Même si le port série apparaît dans l'invité, il ne sert à rien – comme un vrai port COM sans câble.
 - `server <pipename>` : Sur un hôte Windows, ceci dit à VirtualBox de créer un tuyau nommé sur l'hôte, appelé `<pipename>` et de connecter le port série virtuel dessus. Remarquez que Windows exige que le nom d'un tuyau nommé commence par `\.\pipe\`.
 - `Sur un hôte Linux, plutôt qu'un tuyau nommé, une socket de domaine local est utilisée.`
 - `client <pipename>` : Ceci agit comme `server ...`, sauf que le tuyau (ou la socket de domaine local) n'est pas créée par VirtualBox, mais est supposée déjà exister.
 - `<devicename>` : Si, au lieu de celui ci-dessus, le nom du périphérique du port série matériel de l'hôte est spécifié, le port série virtuel est connecté à ce port matériel. Sur un hôte Windows, le nom du périphérique sera un port COM tel que `COM1` ; sur un hôte Linux, le nom du périphérique ressemblera à `/dev/ttys0`. Cela vous permet de "brancher" un vrai port série à une machine virtuelle.
- `--audio none|null|oss` : Avec cette option, vous pouvez définir si la VM supporte le son.
- `--clipboard disabled|hosttoguest|guesttohost|bidirectional` : Avec ce paramètre, vous pouvez sélectionner si le presse-papier du système d'exploitation invité doit être partagé avec l'hôte ; voir chapitre 3.3, *Paramètres généraux*, page 51. Cela exige que les suppléments invité soient installés dans la machine virtuelle.
- `--monitorcount <nombre>` : Cela active le support multi-moniteurs ; voir chapitre 3.5, *Paramètres d'affichage*, page 55.
- `--usb on|off` : Cette option active ou désactive le contrôleur virtuel USB de la VM ; voir chapitre 3.10.1, *Paramètres USB*, page 60 pour les détails.
- `--usbehci on|off` : Cette option active ou désactive le contrôleur USB 2.0 virtuel de la VM ; voir chapitre 3.10.1, *Paramètres USB*, page 60 pour des détails.

8.8.4 Paramètres de la machine distante

Les paramètres suivants concernant le comportement de la machine distante sont disponibles avec `VBoxManage modifyvm` :

- `--vrde on|off` : Avec l'interface graphique de VirtualBox, ceci active ou désactive le serveur VirtualBox remote desktop extension (VRDE). Remarquez que si vous utilisez `VBoxHeadless` (voir chapitre 7.1.2, *VBoxHeadless, le serveur de bureau distant*, page 110), VRDE est activé par défaut.
- `--vrdeport default|<ports>` : Un port ou une plage de ports que le serveur VRDE peut sonder ; “default” ou “0” signifient le port 3389, le port RDP standard. Vous pouvez spécifier une liste de ports ou de plages de ports séparée par des virgules. Utilisez un tiret entre de numéros de ports pour spécifier une plage. Le serveur VRDE sondera un des ports disponibles dans la liste spécifiée. Une seule machine peut utiliser un port donné en même temps. Par exemple, l'option `--vrdeport 5000,5010-5012` dira au serveur de sonder un des ports suivants : 5000, 5010, 5011 ou 5012.
- `--vrdeaddress <adresse IP>` : L'adresse IP de l'interface réseau de l'hôte que sondera le serveur VRDE. Si vous la spécifiez, le serveur n'acceptera les connexions que sur l'interface de l'hôte spécifiée.
- `--vrdeauthtype null|external|guest` : Cela vous permet de choisir s'il faut une autorisation et sous quelle forme ; voir chapitre 7.1.5, *Authentification RDP*, page 113 pour des détails.
- `--vrdemulticon on|off` : Ceci active la multiconnexion au même serveur VRDE si le serveur supporte cette fonction ; voir chapitre 7.1.7, *Connexions multiples au serveur VRDP*, page 116.
- `--vrdereusecon on|off` : Ceci spécifie le comportement du serveur VRDE quand plusieurs connexions sont désactivées. Quand on active cette option, le serveur autorisera un nouveau client à se connecter et il fermera la connexion existante. Quand cette option est désactivée (c'est le réglage par défaut), une nouvelle connexion ne sera pas acceptée si un client est déjà connecté au serveur.
- `--vrdevideochannel on|off` : Ceci active la redirection graphique, si elle est supportée par le serveur VRDE ; voir chapitre 7.1.9, *Redirection graphique VRDP*, page 116.
- `--vrdevideochannelquality <pourcent>` : Définit la qualité de l'image dans la redirection graphique ; voir chapitre 7.1.9, *Redirection graphique VRDP*, page 116.

8.8.5 Paramètres de téléportation

Avec les commandes suivantes pour `VBoxManage modifyvm`, vous pouvez configurer une machine pour qu'elle soit une cible de téléportation. Voir chapitre 7.2, *Téléportation*, page 117 pour une introduction détaillée.

- `--teleporter on|off` : Avec ce paramètre, vous activez ou désactivez l'attente par une machine d'une requête de téléportation venant du réseau lorsqu'elle est démarrée. Si elle est activée, quand on démarre la machine, cela ne démarre pas la machine virtuelle comme cela est normalement le cas, mais cela attend qu'une requête de téléportation n'arrive à l'adresse listée avec les deux options suivantes.
- `--teleporterport <port>, --teleporteraddress <adresse>` : il faut les utiliser avec `--teleporter` et elle dit à une machine virtuelle le port et l'adresse sur lesquelles elle doit écouter les demandes de téléportation d'une autre machine virtuelle. `<port>` peut

être n'importe quel numéro de port TCP/IP libre (comme 6000) ; <adresse> peut être n'importe quelle adresse IP ou nom d'hôte et il spécifie une socket TCP/IP à sonder. Par défaut, c'est "0.0.0.0", ce qui signifie toutes les adresses.

- **--teleporterpassword <mot_de_passe>** : si vous donnez cet argument facultatif, la requête de téléportation ne réussira que si la machine source spécifie le même mot de passe que celui donné avec cette commande.
- **--teleporterfichier_mot_de_passe <mot_de_passe>** : si vous donnez cet argument facultatif, la requête de téléportation ne réussira que si la machine source spécifie le même mot de passe que celui spécifié dans le fichier donné avec cette commande. Utilisez `stdin` pour lire le mot de passe à partir de l'entrée standard.
- **--cpuid <leaf> <eax> <ebx> <ecx> <edx>** : Les utilisateurs avancés peuvent utiliser cette commande avant une opération de téléportation pour restreindre les possibilités du processeur virtuel présenté par VirtualBox au système d'exploitation invité. Il faut l'exécuter à la fois sur la machine source et cible. impliquées dans la téléportation et cela modifiera ce que l'invité verra quand il exécutera l'instruction machine `CPUID`. Cel pourrait aider les applications se comportant mal à cause d'un mauvais présupposé sur la présence de certaines capacités du processeur. La signification des paramètres dépend du matériel ; merci de vous reporter aux manuels des processeurs AMD ou Intel.

8.9 VBoxManage clonevm

Cette commande crée une copie complète ou liée d'une machine virtuelle existante.

La sous-commande `clonevm` prend au moins en argument le nom de la machine virtuelle à cloner. Vous pouvez utiliser les paramètres supplémentaires suivants pour configurer davantage la procédure de clonage de la VM :

- **--snapshot <uuid>|<nom>** : Sélectionne un instantané spécifique auquel se reporte le processus de clonage. Par défaut, c'est à l'état courant qu'il se réfère.
- **--mode machine|machineandchildren|all** : Sélectionne le mode de clonage. Si `machine` est sélectionnée (comme par défaut), l'état actuel de la VM sans instantanés est cloné. En mode `machineandchildren`, l'instantané fourni par `--snapshot` et tous les instantanés fils sont clonés. Si vous sélectionnez le mode `all`, tous les instantanés de l'état actuel seront clonés.
- **--options link|keepallmacs|keepnatmacs|keepdisknames** : Permet de peaufiner encore davantage l'opération de clonage. La première dit de créer un clone lié, ce qui n'est possible que pour le clonage d'une machine à partir d'un instantané. Les deux options suivantes permettent de définir la façon dont les adresses MAC de chaque carte réseau virtuel seront gérées. Elles peuvent être soit réinitialisées (comme par défaut), soit inchangées (`keepallmacs`), soit inchangées quand le type de réseau est NAT (`keepnatmacs`). Si vous ajoutez `keepdisknames`, toutes les nouvelles images de disque dur s'appellent comme celles d'origine, sinon elles sont renommées.
- **--name <name>** : Sélectionne un nouveau nom pour la nouvelle machine virtuelle. Par défaut, il s'agit de "Nom d'origine Clone".
- **--basefolder <basefolder>** : Sélectionne le dossier où devrait être sauvegardée la configuration.
- **--uuid <uuid>** : Sélectionne l'UUID de la nouvelle VM. Cette id doit être unique dans l'instance de VirtualBox où ce clone doit être enregistré. Par défaut, un nouvel UUID est créé.

- **--register** : Enregistre automatiquement le nouveau clone dans l'installation de VirtualBox. Si vous voulez enregistrer la nouvelle VM plus tard à la main, voir chapitre 8.6, *VBoxManage registervm / unregistervm*, page 129 pour les instructions pour faire cela.

8.10 VBoxManage import

Cette commande importe un applicatif virtuel au format OVF en copiant les images de disque virtuelles et en créant les machines virtuelles dans VirtualBox. Voir chapitre 1.14, *Importer et exporter des machines virtuelles*, page 33 pour une présentation des applicatifs.

La commande `import` prend comme argument au moins le chemin d'un fichier OVF en entrée et elle attend que les images de disque, s'il en faut, soient dans le même répertoire que le fichier OVF. Beaucoup d'autres options en ligne de commande sont supportées pour contrôler en détails ce qui est importé et pour modifier les paramètres d'importation, mais les détails dépendent du contenu du fichier OVF.

Il est donc recommandé de lancer d'abord la sous-commande `import` avec l'option `--dry-run` ou `-n`. Cela affichera alors à l'écran le contenu d'un applicatif et la façon de l'importer dans VirtualBox, ainsi que les options facultatives en ligne de commandes pour influencer le comportement de l'importation.

Par exemple, voilà l'écran qui s'afficherait pour un applicatif modèle contenant un invité Windows XP :

```
VBoxManage import WindowsXp.ovf --dry-run
Interpreting WindowsXp.ovf...
OK.

Virtual system 0:
 0: Suggested OS type: "WindowsXP"
    (change with "--vsys 0 --ostype <type>"; use "list ostypes" to list all)
 1: Suggested VM name "Windows XP Professional_1"
    (change with "--vsys 0 --nomvm <name>")
 3: Number of CPUs: 1
    (change with "--vsys 0 --cpus <n>")
 4: Guest memory: 956 MB (change with "--vsys 0 --memory <MB>")
 5: Sound card (appliance expects "ensoniq1371", can change on import)
    (disable with "--vsys 0 --unit 5 --ignore")
 6: USB controller
    (disable with "--vsys 0 --unit 6 --ignore")
 7: Network adapter: orig bridged, config 2, extra type=bridged
 8: Floppy
    (disable with "--vsys 0 --unit 8 --ignore")
 9: SCSI controller, type BusLogic
    (change with "--vsys 0 --unit 9 --scsitype {BusLogic|LsiLogic}");
    (disable with "--vsys 0 --unit 9 --ignore")
10: IDE controller, type PIIX4
    (disable with "--vsys 0 --unit 10 --ignore")
11: Hard disk image: source image=WindowsXp.vmdk,
    target path=/home/user/disks/WindowsXp.vmdk, controller=9;channel=0
    (change controller with "--vsys 0 --unit 11 --controller <id>";
    (disable with "--vsys 0 --unit 11 --ignore")
```

Comme vous pouvez le voir, les éléments de configuration individuelle sont numérotés et en fonction de leur type de support, les options en ligne de commande. La sous-commande `import` peut être redirigée pour ignorer de nombreux éléments tels que ceux avec des options `--vsys X --unit Y --ignore`, où X est le numéro du système virtuel (zéro sauf s'il y a plusieurs descriptions de système virtuel dans l'applicatif) et Y le numéro de l'élément, tel qu'affiché à l'écran.

Dans l'exemple ci-dessus, Item #1 spécifie le nom de la machine cible dans VirtualBox. Items #9 et #10 spécifient respectivement les contrôleurs de disque. Item #11 décrit une image de disque dur ; dans ce cas, l'option supplémentaire `--controller` indique l'élément auquel devrait être connectée l'image de disque, avec le fichier OVF d'origine par défaut.

Vous pouvez associer plusieurs éléments pour un même système virtuel derrière une même option `--vsys`. Par exemple, pour importer une machine comme décrit dans l'OVF mais sans la carte son ni le contrôleur USB, mais avec l'image de disque connectée au contrôleur IDE au lieu du contrôleur SCSI, utilisez ceci :

```
VBoxManage import WindowsXp.ovf
  --vsys 0 --unit 5 --ignore --unit 6 --ignore --unit 11 --controller 10
```

8.11 VBoxManage export

Cette commande exporte une ou plusieurs machines virtuelles à partir de VirtualBox vers un applicatif virtuel au format OVF, ce qui inclut la copie de leurs images de disque virtuel en VMDK compressé. Voir chapitre [1.14, Importer et exporter des machines virtuelles](#), page [33](#) pour une présentation des applicatifs.

La commande `export` est simple à utiliser : affichez la machine (ou les machines) que vous voulez exporter dans un même fichier OVF et spécifiez le fichier OVF cible après l'option facultative `--output` ou `-o`. Remarquez que le répertoire du fichier cible OVF recevra également les images de disques exportés au format VNDK compressé (quel que soit le format d'origine) et vous devriez leur laisser assez d'espace disque.

Derrière une simple exportation d'une machine virtuelle donnée, vous pouvez ajouter diverses informations produit dans le fichier applicatif. Utilisez `--product`, `--producturl`, `--vendor`, `--vendorurl` et `--version` pour spécifier les informations supplémentaires. Pour des raisons légales, vous pouvez ajouter un texte de licence ou le contenu d'un fichier de licence en utilisant respectivement les options `--eula` et `--eulafile`. Comme avec l'importation OVF, vous devez utiliser l'option `--vsys X` pour adresser les options précitées à la bonne machine virtuelle.

Pour les produits de virtualisation non pleinement compatibles avec le standard OVF 1.0, vous pouvez activer un mode OVF 0.9 basique avec l'option `--legacy09`.

8.12 VBoxManage startvm

Cette commande démarre une machine virtuelle qui est en état “Éteinte” ou “Sauvegardée”.

Note: Elle n'est offerte que pour des raisons de rétro-compatibilité. Nous vous recommandons de démarrer les machines virtuelles directement via les interfaces choisies car vous pourriez sans cela manquer des messages d'erreur et des informations d'état importantes que peut afficher VirtualBox en console. C'est surtout important pour les autres interfaces que VirtualBox, notre interface graphique, car ils ne peuvent pas afficher des messages d'erreur dans une fenêtre popup. Voir chapitre [7.1.2, VBoxHeadless, le serveur de bureau distant](#), page [110](#) pour plus d'informations.

Le paramètre facultatif `--type` détermine si la machine sera démarrée dans une fenêtre (mode graphique, qui est celui par défaut) ou la sortie devrait passer par `VBoxHeadless`, en activant ou pas VRDE ; voir chapitre [7.1.2, VBoxHeadless, le serveur de bureau distant](#), page [110](#) pour plus d'informations. La liste des types peut changer, il n'est pas garanti que tous les types sont acceptés par la variante d'un produit.

Les valeurs suivantes sont autorisées :

gui Démarre une VM en affichant une fenêtre graphique. C'est le cas par défaut.

headless Démarre une VM sans fenêtre pour un affichage distant.

8.13 VBoxManage controlvm

La sous-commande `controlvm` vous permet modifier l'état d'une machine virtuelle en fonction. Vous pouvez spécifier ce qui suit :

- `VBoxManage controlvm <vm> pause` met temporairement une machine virtuelle en attente, sans vraiment changer son état. La fenêtre de la VM sera griseée pour indiquer que la VM est actuellement en pause. (Cela revient à sélectionner l'élément "Pause" dans le menu "Machine" de l'interface graphique.)
- Utilisez `VBoxManage controlvm <vm> resume` pour inverser une commande `pause` précédente. (Cela revient à sélectionner l'élément "Reprendre" du menu "Machine" de l'interface graphique.)
- `VBoxManage controlvm <vm> reset` a le même effet sur une machine virtuelle que l'appui sur le bouton "Reset" d'un vrai ordinateur : un redémarrage à froid de la machine virtuelle, ce qui redémarrera le système d'exploitation invité immédiatement. L'état de la VM n'est pas sauvegardé et des données peuvent être perdues. (Cela revient à sélectionner l'élément "Réinitialiser" du menu "Machine" de l'interface graphique.)
- `VBoxManage controlvm <vm> poweroff` a le même effet sur une machine virtuelle que de débrancher le câble d'alimentation d'un vrai ordinateur. Là encore, l'état de la VM n'est pas sauvegardé et il se peut que des données soient perdues. (Cela revient à sélectionner l'élément "Couper" de l'interface graphique ou à appuyer sur le bouton Fermer de la fenêtre puis sélectionner "Couper la machine" dans la boîte de dialogue.)

Après cela, l'état de la VM sera "éteint". À partir de là, vous pouvez la relancer ; voir chapitre [8.12, VBoxManage startvm](#), page [139](#).

- `VBoxManage controlvm <vm> savestate` sauvegardera l'état actuel de la VM sur le disque et arrêtera la VM. (Cela revient à sélectionner l'élément "Fermer" du menu "Machine" de l'interface graphique ou à appuyer sur le bouton fermer de la fenêtre et sélectionner "Sauvegarder l'état de la machine" dans la boîte de dialogue.)

Après cela, l'état de la machine sera "Sauvegardé". À partir de là, vous pouvez la relancer ; voir chapitre [8.12, VBoxManage startvm](#), page [139](#).

- `VBoxManage controlvm <vm> teleport --hostname <nom> --port <port> [--passwordfile <fichier> | --password <mot_de_passe>]` passe la machine en source de téléportation et engage une téléportation sur la cible donnée. Voir chapitre [7.2, Téléportation](#), page [117](#) pour une présentation. Si vous spécifiez le mot de passe facultatif, il doit correspondre à celui donné avec la commande `modifyvm` pour la machine cible ; voir chapitre [8.8.5, Paramètres de téléportation](#), page [136](#) pour des détails.

Quelques options supplémentaires sont disponibles avec `controlvm`, ne modifiant pas directement l'état de la VM en fonction :

- L'opération `setlinkstate<1-N>` connecte ou déconnecte les câbles réseaux virtuels à leurs interfaces réseaux.
- `nic<1-N> null|nat|bridged|intnet|hostonly|generic` : Avec ceci, vous pouvez définir, pour chaque carte réseau virtuelle de la VM, le type de réseau qui devrait être disponible. Elles peuvent ne pas être connectées à l'hôte (`null`), utiliser network address translation (`nat`), le réseau bridgé (`bridged`) ou communiquer avec d'autres machines virtuelles en utilisant le réseau interne (`intnet`) ou host-only (`hostonly`) ou accéder à des sous-modes rarement utilisés (`generic`). Ces options correspondent aux modes décrits en détails au chapitre [6.2, Introduction aux modes réseaux](#), page [99](#).

- `usbattach` et `usbdetach` rendent les périphériques USB de l'hôte, à la volée, visibles pour la machine virtuelle sans besoin de créer des filtres préalablement. Les périphériques USB peuvent être désignés par leur UUID (identifiant unique) ou leur adresse sur le système hôte.

Vous pouvez utiliser `VBoxManage list usbhost` pour trouver ces informations.

- `vrde on|off` vous permet d'activer ou de désactiver le serveur VRDP s'il est installé.
- `vrdeport default|<ports>` modifie le port ou la plage de ports que peut sonder le serveur VRDP ; "default" ou "0" signifient le port 3389, celui standard pour RDP. Pour des détails, voir la description de l'option `--vrdeport` au chapitre [8.8.3, Options du port série, du son, du presse-papier, du bureau distant et de l'USB](#), page [135](#).
- `setvideomodehint` demande que le système invité passe dans un mode graphique particulier. Cela exige l'installation des suppléments invité et ne fonctionner pas pour tous les systèmes invités.
- `screenshotpng` prend une impression écran de l'affichage invité et le sauvegarde au format PNG.
- L'opération `setcredentials` est utilisé pour les identifications distantes dans les invités Windows. Pour des détails, merci de vous reporter au chapitre [9.2, Identifications automatiques dans l'invité](#), page [163](#).
- L'opération `guestmemoryballoon` modifie la taille du ballon de la mémoire de l'invité, c'est-à-dire la mémoire affectée par les suppléments invité de VirtualBox au système d'exploitation invité et rendue à l'hyperviseur pour être réutilisée par d'autres machines virtuelles. Elle doit être spécifiée en mégaoctets. Pour des détails, voir chapitre [4.8.1, Faire du ballon avec la mémoire](#), page [83](#).
- `cpuexecutioncap <1-100>` : Cette opération contrôle la quantité de temps processeur que peut utiliser un processeur virtuel. Une valeur de 50 implique qu'un seul processeur virtuel peut utiliser jusqu'à 50% d'un seul processeur de l'hôte.

8.14 VBoxManage discardstate

Cette commande désactive l'état sauvegardé d'une machine virtuelle qui n'est pas en fonction, ce qui fera redémarrer son système d'exploitation la prochaine fois que vous démarrez. Cela revient à débrancher le câble d'alimentation d'une machine physique et vous devriez l'éviter si possible.

8.15 VBoxManage adoptstate

Si vous avez un fichier d'état sauvegardé (.sav), c'est-à-dire séparé de la configuration de la VM, vous pouvez "adopter" le fichier. Cela passera la VM en état sauvegardé et quand vous la démarrez, VirtualBox essaiera de la restaurer à partir du fichier d'état sauvegardé que vous avez indiqué. Cette commande ne devrait être utilisée que dans des configurations spéciales.

8.16 VBoxManage snapshot

Cette commande est utilisée pour contrôler les instantanés en ligne de commande. Un instantané consiste en une copie complète des paramètres de la machine virtuelle, copiés à un moment où l'instantané a été pris, et éventuellement un fichier d'état sauvegardé de la machine virtuelle si

l'instantané a été pris alors que la machine était en fonction. Après qu'un instantané a été pris, VirtualBox crée un disque dur de différenciation pour chaque disque dur normal associé à la machine pour que, quand on restaure un instantané, les disques durs virtuels de la machine virtuelle puissent être rapidement réinitialisés en abandonnant simplement les fichiers de différenciation.

L'opération `take` prend un instantané de l'état actuel de la machine virtuelle. Vous devez fournir un nom pour l'instantané et vous pouvez éventuellement entrer une description. Le nouvel instantané est inséré à l'arborescence des instantanés comme un fichier de l'instantané actuel et il devient le nouvel instantané actuel.

L'opération `delete` efface un instantané (spécifié par son nom ou son UUID). Cela peut être long à se terminer car les images de différenciation associées à l'instantané pourraient devoir être synchronisées avec leurs images de différenciation filles.

L'opération `restore` restaurera l'instantané donné (spécifié par son nom ou son UUID) en réinitialisant les paramètres de la machine virtuelle et l'état actuel à ceux de l'instantané. L'état actuel précédent de la machine sera perdu. Après cela, l'instantané donné devient le nouvel instantané "actuel" pour que les instantanés consécutifs s'insèrent sous l'instantané à partir duquel on a restauré.

L'opération `restorecurrent` est un raccourci pour restaurer l'instantané actuel (donc l'instantané d'où provient l'état actuel). Cette sous-commande revient à utiliser la sous-commande "restore" avec le nom ou l'UUID de l'instantané actuel, sauf qu'elle évite l'étape supplémentaire de déterminer ce nom ou cet UUID.

Avec l'opération `edit`, vous pouvez modifier le nom ou la description d'un instantané existant.

Avec l'opération `showvminfo`, vous pouvez voir les paramètres de la machine virtuelle stockées dans un instantané existant.

8.17 VBoxManage closemedium

Cette commande supprime un fichier image de disque dur, de DVD ou disquette, d'un registre de médias VirtualBox.²

Éventuellement, vous pouvez demander que l'image soit effacée. Vous aurez les retours circonstanciés selon lesquels la suppression a échoué, mais l'image ne sera pas enregistrée.

8.18 VBoxManage storageattach

Cette commande attache/modifie/supprime un média de stockage connecté à un contrôleur de stockage précédemment ajouté avec la commande `storagectl` command (voir la section précédente). La syntaxe est ainsi :

```
VBoxManage storageattach  <uuid|nomvm>
                      --storagectl <name>
                      [--port <number>]
                      [--device <number>]
                      [--type dvddrive|hdd|fdd]
                      [--medium none|emptydrive|
                           <uuid>|<filename>|host:<drive>|iscsi]
                      [--mtype normal|writethrough|immutable|shareable]
                      [--comment <text>]
                      [--setuuid <uuid>]
                      [--setparentuuid <uuid>]
                      [--passthrough on|off]
                      [--tempeject on|off]
```

²Avant VirtualBox 4.0, il fallait appeler `VBoxManage openmedium` avant de pouvoir attacher à média à une machine virtuelle ; cet appel "enregistrait" le média avec le registre des médias global de VirtualBox. Avec VirtualBox 4.0, ce n'est plus nécessaire ; les médias sont ajoutés aux registres des médias automatiquement. L'appel "closemedium" a cependant été conservé pour permettre de supprimer explicitement un média du registre.

```
[--nonrotational on|off]
[--discard on|off]
[--bandwidthgroup name|none]
[--forceunmount]
[--server <name>|<ip>]
[--target <target>]
[--tport <port>]
[--lun <lun>]
[--encodedlun <lun>]
[--username <username>]
[--password <password>]
[--initiator <initiator>]
[--intnet]
```

Un certain nombre de paramètres sont requis classiquement ; ceux à la fin de la liste ne sont requis que pour les cibles iSCSI (voir ci-dessous).

Les paramètres classiques sont :

uuid|nomvm Le nom ou l'UUID de la VM. Obligatoire.

storagectl Nom du contrôleur de stockage. Obligatoire. Vous pouvez obtenir la liste des contrôleurs de stockage actuellement attachés à une VM avec `VBoxManage showvminfo` ; voir chapitre 8.5, *VBoxManage showvminfo*, page 128.

port Le numéro du port du contrôleur de stockage qui doit être modifié. Obligatoire, sauf si le contrôleur de stockage n'a qu'un seul port.

device Le numéro du périphérique du port à modifier. Obligatoire, sauf si le contrôleur de stockage n'a qu'un périphérique par port.

type Définit le type de lecteur auquel doit être attaché/détaché/modifié un média. On ne peut se passer de cet argument que si le type de média peut être déterminé par le média donné avec l'argument `--medium` argument ou à partir de précédents rattachements de médias

medium Spécifie ce qu'il faut attacher. Les valeurs suivantes sont supportées :

- “none” : Tout périphérique existant serait supprimé du slot donné.
- “emptydrive” : Pour un lecteur amovible ou de DVD virtuel, cela fait se comporter le périphérique comme un lecteur amovible dans lequel on n'a inséré aucun média.
- “additions” : Seulement pour un lecteur DVD virtuel, ceci connecte l'image des *Suppléments invité de VirtualBox* au slot de périphérique donné.
- Si vous spécifiez un UUID, il doit s'agir de l'UUID d'un média de stockage déjà connu de VirtualBox (par exemple car il a été attaché à une autre machine virtuelle). Voir chapitre 8.4, *VBoxManage list*, page 127 pour la façon de lister les médias connus. Ce média est alors attaché au slot du périphérique donné.
- Si vous spécifiez un nom de fichier, il doit s'agir du chemin complet vers une image de disque existante (ISO, RAW, VDI, VMDK ou autre), qui est donc attaché au slot du périphérique donné.
- “host:<lecteur>” : Pour un lecteur amovible ou de DVD virtuel seulement, ceci connecte le slot du périphérique donné au lecteur amovible ou de DVD spécifié sur l'ordinateur hôte.
- “iscsi” : Pour les disques durs virtuels seulement, cela permet de spécifier une cible iSCSI. Dans ce cas, vous devez donner davantage de paramètres ; voir ci-dessous.

Certaines modifications ci-dessus, en particulier concernant les médias amovibles (disquettes ou CDs/DVDs), peuvent prendre effet pendant que la VM est en fonction. Les autres (changement de périphérique ou dans les slots des disques durs) exigent que la VM soit éteinte.

mtype Définit la façon dont ce média se comporte vis-à-vis des instantanés et des opérations d'écriture. Voir chapitre [5.4, Modes spéciaux d'écriture d'images](#), page [90](#) pour des détails.

comment Une description que vous souhaitez stocker avec ce média (facultative ; par exemple pour une cible iSCSI, "Gros serveur de stockage de l'étage inférieur"). C'est purement descriptif et facultatif pour que le média fonctionne correctement.

setuuid, setparentuuid Modifie l'UUID ou l'UUID parent d'un média avant de l'attacher à une VM. C'est une option pour experts. Une utilisation impropre peut rendre le média inutilisable ou donner des configurations de VM cassées si une autre VM se réfère déjà au même média. La variante la plus fréquemment utilisée est `--setuuid ""`, qui affecte un nouvel UUID (aléatoire) à une image. Cela sert à résoudre des erreurs d'UUID dupliqués si on a dupliqué une image par les outils de copie de fichiers..

passthrough Seulement pour un lecteur DVD virtuel, vous pouvez activer le support de gravure sur DVD (actuellement expérimental ; voir chapitre [5.9, Support des CD/DVD](#), page [96](#)).

tempeject Seulement pour un lecteur DVD virtuel, vous pouvez configurer le comportement pour avoir l'éjection du média synchrnmiùe avec l'invité. Si vous l'activez, l'éjection n'a qu'un effet temporaire. Si vous éteignez puis redémarrez la VM, le média configuré au départ sera toujours dans le lecteur.

nonrotational Ce paramètre permet d'activer le drapeau non-rotational pour les disques durs virtuels. Certains invités (comme Windows 7+) gèrent de tels disques comme des SSDs et ne fragmentent pas de tels médias.

bandwidthgroup Définit le groupe de bande passante à utiliser pour le périphérique donné ; voir chapitre [5.8, Limiter la bande passante des images de disque](#), page [96](#).

forceunmount Seulement pour un lecteur DVD ou amovible virtuel, cela démonte de force le DVD/CD/disquette ou monte un nouveau DVD/CD/disquette même si celui précédent est verrouillé en lecture par l'invité. De nouveau, voir chapitre [5.9, Support des CD/DVD](#), page [96](#) pour des détails.

Quand on utilise "iscsi" avec le paramètre `--medium` pour le support iSCSI – see chapitre [5.10, Serveurs iSCSI](#), page [97](#) –, vous pouvez ou devez utiliser des paramètres supplémentaires :

server Le nom de l'hôte ou l'adresse IP de la cible iSCSI ; requis.

target La chaîne du nom de la cible. Il est déterminé par la cible iSCSI et il est utilisé pour identifier la ressource de stockage ; requis.

tport Numéro de port TCP/IP du service iSCSI sur la cible (facultatif).

lun Numéro d'unité logique de la ressource cible (facultatif). Souvent, cette valeur vaut zéro.

username, password Nom d'utilisateur et mot de passe (initiator secret) pour l'authentification sur la cible si nécessaire (facultatif).

Note: Le nom d'utilisateur et le mot de passe sont stockés sans chiffrement (donc en clair) dans le fichier XML de configuration de la VM si vous ne fournissez aucun paramètre de mot de passe. Quand vous spécifiez pour la première fois un paramètre de mot de passe, le mot de passe est stocké et chiffré.

intnet Si vous le spécifiez, connecte à la cible iSCSI via le réseau interne. Cela nécessite une configuration supplémentaire décrite au chapitre [9.8.3, Accès à des cibles iSCSI via le réseau interne](#), page [176](#).

8.19 VBoxManage storagectl

Cette commande attache/modifie/supprime un contrôleur de stockage. Après elle, on peut attacher des médias virtuels au contrôleur avec la commande `storageattach` (voir la prochaine section).

La syntaxe est comme suit :

```
VBoxManage storagectl      <uuid|nomvm>
                          --name <nom>
                          [--add <ide/sata/scsi/floppy>]
                          [--controller <LsiLogic|LSILogicSAS|BusLogic|
                                         IntelAhci|PIIX3|PIIX4|ICH6|I82078>]
                          [--sataportcount <1-30>]
                          [--hostiocache on|off]
                          [--bootable on|off]
                          [--remove]
```

où les paramètres signifient :

uuid|nomvm Nom ou UUID de la VM. Obligatoire.

nom Nom du contrôleur de stockage. Obligatoire.

add Définit le type de bus système auquel doit être connecté le contrôleur de stockage.

controller Permet de choisir le type de chipset à émuler pour le contrôleur de stockage donné.

sataportcount Ceci détermine le nombre de ports que le contrôleur SATA devrait supporter.

hostiocache Configure l'utilisation du cache E/S de l'hôte pour toutes les images de disque attachées à ce contrôleur de stockage. Pour des détails, merci de voir chapitre [5.7, Mise en cache des E/S dans l'hôte](#), page [95](#).

bootable Sélectionne si ce contrôleur est amorçable.

remove Supprime le contrôleur de stockage de la config de la VM.

8.20 VBoxManage bandwidthctl

Cette commande crée/efface/modifie/affiche les groupes de bande passante de la machine virtuelle donnée :

```
VBoxManage bandwidthctl      <uuid|nomvm>
                          add <nom> --type disk|network --limit <megabytes per second>[k|m|g|K|M|G] |
                          set <nom> --limit <megabytes per second>[k|m|g|K|M|G] |
                          remove <nom> |
                          list [--machinereadable]
```

Les sous-commandes suivantes sont disponibles :

- `add`, crée un nouveau groupe de bande passante d'un type donné.
- `set`, modifie la limite d'un groupe de bande passante existante.
- `remove`, détruit un groupe de bande passante.
- `list`, affiche tous les groupes de bande passante définis pour la VM donnée.

Les paramètres signifient :

uuid|nomvm Le nom ou l'UUID de la VM. Obligatoire.

nom Nom du groupe de bande passante. Obligatoire.

type Type de groupe de bande passante. Obligatoire. Deux types sont supportés : `disk` et `network`. Voir chapitre [5.8, Limiter la bande passante des images de disque](#), page [96](#) ou chapitre [6.9, Limiter la bande passante des E/S réseaux](#), page [107](#) pour une description d'un type particulier.

limit Spécifie la limite du groupe donné. On peut le modifier pendant que la VM est en fonction. L'unité par défaut est en mégaoctet par seconde. Vous pouvez modifier l'unité en spécifiant un des suffixes suivants : `k` pour kilobits/s, `m` pour mégabits/s, `g` pour gigabits/s, `K` pour kilo octets/s, `M` pour mégaoctets/s, `G` pour giga octets/s.

Note: Les limites de bande passante du réseau ne s'appliquent qu'au trafic émis par les machines virtuelles. Le trafic reçu par les VMs est illimité.

Note: Pour supprimer un groupe de bande passante, il ne doit être référencé sur aucun disque ou adaptateur d'une VM en fonction.

8.21 VBoxManage showhdinfo

Cette commande affiche des informations sur une image de disque virtuelle, en particulier sa taille sur le disque, son type et les machines virtuelles qui l'utilisent.

Note: Pour être compatible avec des versions plus anciennes de VirtualBox la commande “showvdiinfo” est également supportée et associée en interne à la commande “showhdinfo”.

L'image de disque doit être spécifiée soit par son UUID (si le média est enregistré), soit par le nom du fichier. Les images enregistrées peuvent être listées par `VBoxManage list hdds` (voir chapitre [8.4, VBoxManage list](#), page [127](#) pour plus d'informations). Un nom de fichier doit être spécifié en tant que chemin valide, soit absolu soit relatif en partant du répertoire actuel.

8.22 VBoxManage createhd

Cette commande crée une nouvelle de disque dur virtuel. La syntaxe est comme suit :

```
VBoxManage createhd      --filename <filename>
                      --size <mégoctet>
                      [--format VDI|VMDK|VHD] (par défaut : VDI)
                      [--variant Standard,Fixed,Split2G,Stream,ESX]
```

où les paramètres signifient :

filename Permet de choisir un nom de fichier. Obligatoire.

size Permet de définir la capacité de l'image, en MiO. Obligatoire.

format Permet de choisir un format de fichier pour le fichier de sortie, différent du format du fichier d'entrée.

variant Permet de choisir une variante du format de fichier pour le fichier de sortie. C'est une liste de drapeaux séparée par des virgules. Toutes les combinaisons ne sont pas supportées, spécifier des drapeaux incohérents donne un message d'erreur.

Note: Pour être compatible avec d'anciennes versions de VirtualBox, la commande “createvdi” est également supportée et associée en interne avec la commande “createhd”.

8.23 VBoxManage modifyhd

Avec la commande `modifyhd`, vous pouvez modifier les caractéristiques d'une image de disque après l'avoir créé :

```
VBoxManage modifyhd      <uuid>|<nomfichier>
                      [--type normal|writethrough|immutable|shareable|
                           readonly|multiattach]
                      [--autoreset on|off]
                      [--compact]
                      [--resize <megabytes>|--resizebyte <octets>]
```

Note: Malgré le “hd” du nom de la sous-commande, la commande fonctionne avec toutes les images de disque et pas seulement les disques durs. Pour être compatible avec les anciennes versions de VirtualBox, la commande “modifyvdi” est également supportée et a été associée en interne à la commande “modifyhd”.

L'image de disque doit être spécifiée soit par son UUID (si le média est enregistré), soit par le nom du fichier. Les images enregistrées peuvent être listées par `VBoxManage list hdds` (voir chapitre 8.4, [VBoxManage list](#), page 127 pour plus d'informations). Un nom de fichier doit être spécifié en tant que chemin valide, soit absolu soit relatif en partant du répertoire actuel.

Les options suivantes sont disponibles :

- Avec l'argument `--type`, vous pouvez modifier le type d'une image existante entre le mode normal, immuable, write-through et autres modes ; voir chapitre 5.4, [Modes spéciaux d'écriture d'images](#), page 90 pour des détails.
- Pour les disques durs immuables seulement (de différenciation), l'option `--autoreset on|off` détermine si le disque est automatiquement réinitialisé à chaque démarrage de la VM (de nouveau, voir chapitre 5.4, [Modes spéciaux d'écriture d'images](#), page 90). Par défaut, c'est actif.
- L'option `--compact`, peut être utilisée pour compresser des images de disque (à savoir supprimer des blocs qui ne contiennent que des zéros). Cela réallègera une image à taille dynamique ; cela réduira la taille *physique* de l'image sans changer la taille logique du disque virtuel. La compression fonctionne aussi bien pour les images de base que celles de différenciation créées dans un instantané.

Pour que l'opération soit efficace, il faut d'abord libérer de l'espace dans le système invité en y mettant des zéros via le logiciel adapté. Pour les invités Windows, vous pouvez utiliser l'outil `sdelete` de Microsoft. Exécutez `sdelete -z` dans l'invité pour mettre des zéros dans l'espace libre avant de compresser l'image de disque virtuel. Pour Linux, utilisez l'outil `zerofree` qui supporte les systèmes de fichiers `s` `ext2/ext3`. Pour les invités Mac OS X, utilisez `utilisez la fonction Supprimer l'espace libre de l'outil de disque intégré`. Utilisez-y `Zero Out Data`.

Remarquez que la compression n'est actuellement disponible que pour les images VDI. Vous pouvez obtenir la même chose en mettant des zéros sur les blocs libres et clonant le disque vers un autre format dynamique. Vous pouvez utiliser ce contournement jusqu'à ce que la compression soit supportée pour les autres formats de disque que VDI.

- L'option `--resize x` (où x est le nouvel espace total désiré en **mégaoctets**) vous permet de modifier la capacité d'une image existante ; ceci ajuste la taille *logique* d'un disque virtuel sans beaucoup changer la taille physique.³ Cela ne fonctionne actuellement que pour les formats VDI et VHD et pour les variantes, à taille dynamique, et vous ne pouvez l'utiliser que pour augmenter (ou réviser) la capacité. Par exemple, si vous avez créé au départ un disque de 10G à présent plein, vous pouvez utiliser la commande `--resize 15360` pour passer la capacité à 15G (15,360Mo) sans devoir créer une nouvelle image et y copier toutes les données à partir d'une machine virtuelle. Cependant, remarquez que cela ne change que la capacité du lecteur ; vous devrez ensuite, en général, utiliser un outil de gestion des partitions dans l'invité pour ajuster la partition principale et remplir le lecteur. L'option `--resizebyte x` fait presque la même chose mais x est exprimé en octets au lieu de mégaoctets.

8.24 VBoxManage clonehd

Cette commande duplique une image de disque dur virtuel enregistrée vers un nouveau fichier image ayant un nouvel identifiant unique (UUID). La nouvelle image peut être transférée vers un autre système hôte ou réimportée dans VirtualBox en utilisant le gestionnaire de médias virtuels ; voir chapitre [5.3, Le gestionnaire de médias virtuels](#), page 88 et chapitre [5.6, Cloner des images de disque](#), page 94. La syntaxe est comme suit :

```
VBoxManage clonehd      <uuid>|<nomfichier> <outputfile>
                      [--format VDI|VMDK|VHD|RAW|<other>]
                      [--variant Standard,Fixed,Split2G,Stream,ESX]
                      [--existing]
```

L'image de disque à cloner et celle cible doivent être spécifiées soit par leur UUID (si le média est enregistré), soit par leur nom du fichier. Les images enregistrées peuvent être listées par `VBoxManage list hdds` (voir chapitre [8.4, VBoxManage list](#), page 127 pour plus d'informations). Un nom de fichier doit être spécifié en tant que chemin valide, soit absolu soit relatif en partant du répertoire actuel.

Les options suivantes sont disponibles :

format Permet de choisir un format de fichier pour le fichier de sortie, autre que celui du fichier d'entrée.

variant Permet de choisir une variante de format de fichier pour le fichier de sortie. C'est une liste de drapeaux séparée par des virgules. Toutes les combinaisons ne sont pas supportées, spécifier des drapeaux incohérents donne un message d'erreur.

existing Effectue le clonage sur un média de destination existant. Seule la partie du média source convenant au média de destination est copiée. Cela signifie que si le média de destination est plus petit que la source, seule une partie en est copiée, et si le média de destination est plus gros que la source, le reste de l'espace du média de destination reste inchangé.

Note: Pour être compatible avec les anciennes versions de VirtualBox, la commande "clonevdi" est également supportée et elle a été associée en interne à la commande "clonehd".

³Le redimensionnement d'images a été ajouté à VirtualBox 4.0.

8.25 VBoxManage convertfromraw

Cette commande convertit une image de disque brute en fichier VirtualBox Disk Image (VDI). La syntaxe est la suivante :

```
VBoxManage convertfromraw  <filename> <outputfile>
                           [--format VDI|VMDK|VHD]
                           [--variant Standard,Fixed,Split2G,Stream,ESX]
                           [--uuid <uuid>]
VBoxManage convertfromraw  stdin <fichiersortie> <octets>
                           [--format VDI|VMDK|VHD]
                           [--variant Standard,Fixed,Split2G,Stream,ESX]
                           [--uuid <uuid>]
```

où les paramètres signifient :

octets La taille du fichier image, en octets, fournie en entrée.

format Sélectionnez le format de l'image de disque à créer. Par défaut, c'est VDI.

variant Permet de choisir une variante de format de fichier pour le fichier en sortie. C'est une liste de drapeaux séparée par des virgules. Toutes les combinaisons ne sont pas supportées, spécifier des drapeaux incohérents donne un message d'erreur.

uuid Permet de spécifier l'UUID ou le fichier de sortie.

La deuxième forme oblige VBoxManage à lire le contenu de l'image de disque à partir de l'entrée standard (utile pour utiliser cette commande en ligne).

Note: Pour être compatible avec d'anciennes versions de VirtualBox, la commande "convertdd" est aussi supportée et elle est associée en interne à la commande "convertfromraw".

8.26 VBoxManage getextradata/setextradata

Ces commandes vous permet d'attacher et de récupérer des chaînes de données dans une machine virtuelle ou à une configuration de VirtualBox (en spécifiant global au lieu d'un nom de machine virtuelle). Vous pouvez spécifier une clé (comme chaîne de texte) pour associer les données que vous pourrez récupérer plus tard. Par exemple :

```
VBoxManage setextradata Fedora5 installdate 2006.01.01
VBoxManage setextradata SUSE10 installdate 2006.02.02
```

associerait la chaîne "2006.01.01" à la clé installdate pour la machine virtuelle Fedora5, et "2006.02.02" pour la machine SUSE10. Vous pourriez récupérer l'information comme suit :

```
VBoxManage getextradata Fedora5 installdate
```

qui renverrait

```
VirtualBox Command Line Management Interface Version 4.2.51
(C) 2005-2013 Oracle Corporation
All rights reserved.
```

```
Value: 2006.01.01
```

To remove a key, the setextradata command must be run without specifying data (only the key), for example:

```
VBoxManage setextradata Fedora5 installdate
```

8.27 VBoxManage setproperty

Cette commande est utilisée pour modifier des paramètres globaux qui modifient toute l'installation de VirtualBox. Certains d'entre eux correspondent aux paramètres de la boîte de dialogue “Paramètres globaux” de l'interface graphique. Les propriétés suivantes sont disponibles :

machinefolder Ceci spécifie le dossier par défaut dans lequel sont stockées les définitions de la machine virtuelle ; voir chapitre [10.1, Où VirtualBox stocke ses fichiers](#), page [196](#) pour les détails.

vrdeauthlibrary Ceci indique la bibliothèque à utiliser quand on a sélectionné l'authentification “externe” pour une machine virtuelle en particulier ; voir chapitre [7.1.5, Authentification RDP](#), page [113](#) pour des détails.

websrvauthlibrary Ceci spécifie la bibliothèque utilisée par le service web pour authentifier les utilisateurs. Pour des détails sur le service web de VirtualBox, merci de vous reporter au manuel de référence SDK distinct de VirtualBox (voir chapitre [11, Interfaces de programmation de VirtualBox](#), page [207](#)).

vrdelibrary Ceci spécifie la bibliothèque qui implémente l'extension Bureau distant de VirtualBox.

hwvirtexenabled Ceci sélectionne l'activation du support de la virtualisation matérielle par défaut.

8.28 VBoxManage usbfilter add/modify/remove

Les commandes `usbfilter` sont utilisées pour travailler avec des filtres USB dans des machines virtuelles ou avec des filtres globaux qui concernent tout le paramétrage de VirtualBox. Les filtres globaux sont appliqués avant les filtres spécifiques aux machines et ils peuvent être utilisés pour empêcher les périphériques d'être capturés par une machine virtuelle. Les filtres globaux s'appliquent toujours dans un ordre particulier, et seul le premier filtre convenant à un périphérique est appliqué. Par exemple, si le premier filtre global dit de garder (rendre disponible) une carte mémoire Kingston particulière et le deuxième d'ignorer tous les périphériques Kingston, cette carte mémoire sera disponible pour toutes les machines ayant un filtre adéquat mais pas les autres périphériques Kingston.

Lors de la création d'un filtre USB en utilisant `usbfilter add`, vous devez fournir trois ou quatre paramètres obligatoires. L'index spécifie la position du filtre dans la liste. S'il y a déjà un filtre à cette place, lui-même et les suivants seront déplacés. Sinon, le nouveau filtre sera ajouté à la fin de la liste. Le paramètre `target` sélectionne la machine virtuelle à laquelle devrait être attaché le filtre, sinon utilisez “global” pour l'appliquer à toutes les machines virtuelles. `name` est le nom du nouveau filtre et des filtres globaux, `action` autorise les machines à accéder aux périphériques correspondant à la description du filtre (“hold”) ou ne leur donne pas accès (“ignore”). En outre, vous devriez spécifier des paramètres de filtrage. Vous pouvez trouver les paramètres des périphériques attachés à votre système en utilisant `VBoxManage list usbhost`. Enfin, vous pouvez spécifier si le filtre devrait être actif, et pour les filtres globaux, s'ils concernent les périphériques locaux, à distance (par connexion RDP) ou autrement.

Quand vous modifiez un filtre USB en utilisant `usbfilter modify`, vous devez spécifier le filtre par index (voir la sortie de `VBoxManage list usbfilters` pour voir les index des filtres globaux et `VBoxManage showvminfo` pour voir les index des machines individuelles), et par cible, laquelle est soit une machine virtuelle, soit “global”. Les propriétés modifiables sont les mêmes que pour `usbfilter add`. Pour supprimer un filtre, utilisez `usbfilter remove` et spécifiez l'index et la cible.

8.29 VBoxManage sharedfolder add/remove

Cette commande vous permet de partager des dossiers de l'ordinateur hôte avec les systèmes d'exploitation invités. Pour cela, il faut que les systèmes invités aient une version installée des Suppléments invité qui supporte cette fonctionnalité.

Les dossiers partagés sont décrits en détails au chapitre [4.3, Dossiers partagés](#), page [75](#).

8.30 VBoxManage guestproperty

Les commandes “guestproperty” vous permettent de voir ou de définir les propriétés d'une machine virtuelle en fonction. Merci de voir chapitre [4.6, Propriétés invité](#), page [80](#) pour une présentation. Comme il y est expliqué, les propriétés de l'invité sont des paires de chaînes clé/valeur arbitraires qu'on peut changer et lire depuis l'hôte ou l'invité, donc on peut les utiliser comme des canaux de communication à faible volume pour des chaînes du moment qu'un invité est en fonction et a installé les suppléments invité. De plus, un certain nombre de valeurs dont les clés commencent par “/VirtualBox/” sont automatiquement définies et maintenues par les suppléments invité.

Les sous-commandes sont disponibles (où <vm>, dans chaque cas, peut être soit un nom soit un UUID d'une VM, comme pour les autres commandes de VBoxManage) :

- **enumerate <vm> [--patterns <pattern>]** : Ceci liste toutes les propriétés invité disponibles pour une VM donnée avec leur valeur. Cette liste sera très limitée si le processus service de l'invité ne peut pas être contacté, notamment car la VM n'est pas en fonction ou les suppléments invité ne sont pas installés.

Si vous spécifiez --patterns <échantillon>, il agit comme un filtre pour ne lister que les propriétés correspondant à l'échantillon donné. L'échantillon peut contenir les caractères jokers suivants :

- * (astérisque) : représente n'importe quel nombre de caractères ; par exemple, “/VirtualBox*” correspondrait à toutes les propriétés commençant par “/VirtualBox”.
- ? (point d'interrogation) : représente un seul caractère de votre choix ; par exemple, “fo?” correspondrait à “foo” et à “for”.
- | (symbole pipe) : peut être utilisé pour spécifier plusieurs échantillons alternatifs ; par exemple, “s* | t*” correspondrait à tout ce qui commence par un “s” ou un “t”.

- **get <vm>** : Ceci ne récupère la valeur que d'une propriété. Si la propriété est introuvable (par exemple car l'invité n'est pas en fonction), ceci affichera

No value set!

- **set <vm> <propriété> [<valeur> [--flags <drapeaux>]]** : Ceci vous permet de définir une propriété invité en spécifiant la clé et la valeur. Si vous ne mettez pas <valeur>, la propriété est effacée. Avec --flags, vous pouvez éventuellement spécifier un comportement supplémentaire (vous pouvez en combiner plusieurs en les séparant par des virgules) :

- TRANSIENT : la valeur ne sera pas stockée avec les données de la VM quand la VM exite ;
- TRANSRESET : la valeur sera effacée dès que la VM redémarrera et/ou existera ;
- RDONLYGUEST : la valeur ne peut être modifiée que par l'hôte, l'invité ne peut que la lire ;
- RDONLYHOST : inversement, cette valeur ne peut être modifiée que par l'invité, l'hôte ne peut que la lire ;

- **readonly** : la combinaison des deux, la valeur ne peut pas être modifiée.
- **wait <vm> <échantillon> --timeout <timeout>** : Ceci attend une valeur particulière décrite par “échantillon” à modifier ou effacer ou à créer. Les règles de l’échantillon sont les mêmes qu’avec la commande “enumerate” ci-dessus.
- **delete <vm> <propriété>** : Efface une propriété invitée définie précédemment.

8.31 VBoxManage guestcontrol

Les commandes “guestcontrol” vous permettent de contrôler certaines choses de l’invité à partir de l’hôte. Merci de voir chapitre 4.7, *Contrôle de l’invité*, page 82 pour une présentation.

En général, la syntaxe est la suivante :

```
VBoxManage guestcontrol <commande>
```

Les sous-commandes suivantes sont disponibles (où <vm>, dans tous les cas, peut être le nom ou l’UUID d’une VM comme avec les autres commandes de VBoxManage) :

- **execute**, qui permet d’exécuter un programme/script (processus) déjà installé et exécutable sur l’invité. Cette commande ne fonctionne que pendant qu’une VM est active et en fonction et sa syntaxe est la suivante :

```
VBoxManage guestcontrol <nomvm>|<uuid> exec[ute]
  --image <chemin du programme> --username <nom>
  [--passwordfile <fichier> | --password <mot de passe>]
  [--environment "<NOM>=<VALEUR> [<NOM>=<VALEUR>]"]
  [--verbose] [--timeout <msec>]
  [--wait-exit] [--wait-stdout] [--wait-stderr]
  [--dos2unix] [--unix2dos]
  -- [<argument1> ... [<argumentN>]]
```

où les paramètres signifient :

uuid|nomvm Le nom ou l’UUID de la VM. Obligatoire.

–image “<chemin du programme>“ Chemin absolu et nom du processus à exécuter dans l’invité, comme C:\Windows\System32\calc.exe

–username <nom> Nom de l’utilisateur sous lequel devrait se lancer le processus. Cet utilisateur doit exister sur l’OS invité.

–passwordfile <fichier> Mot de passe du compte utilisateur spécifié, à lire dans le fichier donné. Si vous n’en donnez pas, un mot de passe vierge est supposé.

–password <mot de passe> Mot de passe du compte utilisateur spécifié avec --username. Si vous n’en donnez pas, un mot de passe vierge est supposé.

–dos2unix Convertit les fins de lignes de la sortie des invités DOS/Windows en fins compatibles UNIX (CR + LF -> LF). Pas encore implémenté.

–environment “<NAME>=<VALUE>“ Une ou plusieurs variables d’environnement à définir ou dérégler.

Par défaut, le nouveau processus de l’invité sera créé avec l’environnement standard de l’OS invité. Cette option permet de modifier cet environnement. Pour définir/modifier une variable, une paire NOM=VALEUR doit être spécifiée ; pour dérégler une certaine variable, le nom sans valeur doit être défini, par exemple NOM=.

Les arguments contenant des espaces doivent être entourés de guillemets. Vous pouvez spécifier plus d’un --environment en même temps pour maintenir vierge la ligne de commande.

-timeout <msec> Valeur (en millisecondes) qui spécifie la durée pendant laquelle le processus démarré est autorisé à fonctionner et combien de temps VBoxManage attend une sortie de ce processus. Si vous ne spécifiez pas de timeout, VBoxManage attendra jusqu'à ce que le processus démarré se termine ou ce qu'une erreur ne survienne.

-unix2dos Convertit les fins de lignes de la sortie des invités DOS/Windows en fins compatibles UNIX (CR + LF -> LF). Pas encore implémenté.

-verbose Dit à VBoxManage d'être plus verbeux.

-wait-exit Attende la fin d'un processus et affiche son code de sortie avec la raison/les drapeaux de sa fin.

-wait-stdout Attende la fin d'un processus et affiche son code de sortie avec la raison/les drapeaux de sa fin. En attendant, VBoxManage récupère la sortie du processus sur stdout.

-wait-stderr Attende la fin d'un processus et affiche son code de sortie avec la raison/les drapeaux de sa fin. En attendant, VBoxManage récupère la sortie du processus sur stderr.

[- [<argument1s> ... [<argumentNs>]]]

Un ou plusieurs arguments à passer au processus à exécuter.

Les arguments contenant des espaces doivent être entourés de guillemets.

Note: Sur Windows, il existe certaines limites pour les applications graphiques ; merci de voir chapitre 14, *Limites connues*, page 231 pour plus d'informations.

Exemples :

```
VBoxManage --nologo guestcontrol "Ma VM" execute --image "/bin/ls"
  --username foo --passwordfile bar.txt --wait-exit --wait-stdout -- -l /usr

VBoxManage --nologo guestcontrol "My VM" execute --image "c:\\windows\\system32\\ipconfig.exe"
  --username foo --passwordfile bar.txt --wait-exit --wait-stdout
```

Remarquez que les deux barres obliques inversées du deuxième exemple ne sont requis que pour les hôtes Unix.

Note: Pour certaines commandes, il faut spécifier le nom d'utilisateur d'un compte utilisateur existant sur l'invité ; les exécutions anonymes ne sont pas supportées pour des raisons de sécurité. Néanmoins, le mot de passe d'un compte utilisateur est facultatif et dépend de la politique et des règles de l'invité. Si vous ne spécifiez aucun mot de passe pour un nom d'utilisateur donné, un mot de passe vierge sera utilisé. Sur certains OS comme Windows, il peut qu'il faille ajuster la politique de sécurité pour autoriser le paramétrage d'un compte utilisateur avec un mot de passe vierge. En outre, les règles du domaine global pourraient s'appliquer et, donc, on ne peut pas les modifier.

À partir de VirtualBox 4.1.2 l'exécution d'un processus invité par défaut se limite à 5 processus invités en même temps. Si un nouveau processus invité est lancé et dépasserait cette limite, le plus ancien dans l'invité qui ne s'exécute pas sera désactivé afin de pouvoir lancer ce nouveau processus. En outre, la récupération de la sortie de cet ancien processus invité ne sera alors plus possible. Si les 5 processus invités sont toujours actifs et en fonction, le démarrage d'un nouveau processus enverra un message d'erreur circonstancié.

Pour augmenter ou diminuer la limite d'exécution des processus invités, il faut modifier soit la propriété `/VirtualBox/GuestAdd/VBoxService/-control-procs-max-kept`, soit la

ligne de commande ee VBoxService en spécifiant `--control-procs-max-kept`. Il faut ensuite redémarrer l'OS invité. Pour gérer un nombre illimité de processus invités, il faut définir une valeur de 0 (non recommandé).

- `copyto`, qui permet de copier des fichiers de l'hôte vers l'invité (seulement si les suppléments invité 4.0 et supérieur sont installés).

```
VBoxManage guestcontrol <nomvm>|<uuid> copyto|cp
    <source invité> <hôte dest> --username <nom>
    [--passwordfile <fichier> | --password <mot de passe>]
    [--dryrun] [--follow] [--recursive] [--verbose]
```

où les paramètres signifient :

uuid|nomvm L'UUID ou le nom de la VM. Obligatoire.

source sur l'hôte Chemin absolu du/des fichier(s) source(s) de l'hôte sur l'hôte pour copier vers l'invité, comme `C:\Windows\System32\calc.exe`. Il peut s'agir d'une expression joker, comme `C:\Windows\System32*.dll`

destination sur l'invité Chemin de destination absolu sur l'invité, comme `C:\Temp`

-username <nom> Nom d'utilisateur sous lequel devrait fonctionner le processus de copie. Cet utilisateur doit exister sur l'OS invité.

-passwordfile <fichier> Mot de passe du compte utilisateur spécifié, à lire dans le fichier donné. Si vous n'en donnez pas, un mot de passe vierge est supposé.

-password <mot de passe> Mot de passe du compte utilisateur spécifié avec `--username`. Si vous n'en donnez pas, un mot de passe vierge est supposé.

-dryrun Dit à VBoxManage de ne faire qu'une exécution dry au lieu de copier vraiment les fichiers dans l'invité.

-follow Active le suivi des liens symboliques sur la source de l'hôte.

-recursive Copie récursivement les fichiers/répertoires de la source spécifiée.

-verbose Dit à VBoxManage d'être plus verbeux.

-flags <drapeaux> Drapeaux supplémentaires à définir. Ce n'est pas utilisé pour le moment.

- `copyfrom`, qui permet de copier des fichiers de l'invité vers l'hôte (seulement si les suppléments invité 4.0 et supérieur sont installés). Il prend les mêmes paramètres que `copyto` ci-dessus.

- `createdirectory`, qui permet de copier des fichiers de l'invité vers l'hôte (seulement si les suppléments invité 4.0 et supérieur sont installés). Il prend les mêmes paramètres que `copyto` ci-dessus.

```
VBoxManage guestcontrol <nomvm>|<uuid> createdir[ectomy]|mkdir|md
    <répertoire invité>... --username <nom>
    [--passwordfile <fichier> | --password <mot de passe>]
    [--parents] [--mode <mode>] [--verbose]
```

où les paramètres signifient :

uuid|nomvm L'UUID ou le nom de la VM. Obligatoire.

répertoire à créer sur l'invité Chemin absolu du/des répertoire(s) à créer sur l'invité, comme `D:\Foo\Bar`. Les répertoires Parents doivent exister (comme dans cet exemple `D:\Foo`) si vous ne mettez pas `--parents`. L'utilisateur spécifié doit avoir les droits appropriés pour créer le répertoire spécifié.

-username <nom> Nom de l'utilisateur sous lequel devrait fonctionner le processus de copie. Cet utilisateur doit exister sur l'OS invité.

- passwordfile <fichier>** Mot de passe du compte utilisateur spécifié, à lire dans le fichier donné. Si vous n'en donnez pas, un mot de passe vierge est supposé.
- password <mot de passe>** Mot de passe du compte utilisateur spécifié avec `--username`. Si vous n'en donnez pas, un mot de passe vierge est supposé.
- parents** Crée aussi les répertoires parent non encore créés pour le répertoire spécifié, par exemple si le répertoire `D:\Foo` de `D:\Foo\Bar` n'existe pas encore, il sera créé. Si on ne spécifie pas `--parent`, l'action échouerait.
- mode <mode>** Définit les droits du répertoire spécifié. Seules des valeurs octales (comme `0755`) sont supportées pour l'heure.
- verbose** Dit à VBoxManage d'être plus verbeux.

- **stat**, qui affiche l'état d'un fichier ou d'un gestionnaire de fichiers de l'invité.

```
VBoxManage guestcontrol <nomvm>|<uuid> stat
    <file>... --username <nom>
    [--passwordfile <fichier> | --password <mot de passe>]
    [--verbose]
```

où les paramètres signifient :

uuid|nomvm L'UUID ou le nom de la VM. Obligatoire.

élément(s) de fichier à vérifier sur l'invité Chemin absolu du/des répertoire(s) à vérifier sur l'invité, comme `/home/foo/a.out`. L'utilisateur spécifié doit avoir les droits adéquats d'accès à/aux élément(s) fichier unique(s).

- username <nom>** Nom de l'utilisateur sous lequel devrait s'exécuter le processus de copie. L'utilisateur doit exister sur l'OS invité.
- passwordfile <fichier>** Mot de passe du compte utilisateur spécifié, à lire dans le fichier donné. Si vous n'en donnez pas, un mot de passe vierge est supposé.
- password <mot de passe>** Mot de passe du compte utilisateur spécifié avec `--username`. Si vous n'en donnez pas, un mot de passe vierge est supposé.
- verbose** Dit à VBoxManage d'être plus verbeux.

- **updateadditions**, qui permet de mettre à jour une version des suppléments invité installée sur l'invité (sur les suppléments invité installés 4.0 et supérieur).

```
VBoxManage guestcontrol <nomvm>|<uuid> updateadditions
    [--source "<fichier des suppléments invité .ISO à utiliser>"] [--verbose]
    [--wait-start]
```

où les paramètres signifient :

uuid|nomvm L'UUID ou le nom de la VM. Obligatoire.

--source "<fichier des suppléments invité .ISO à utiliser>" Chemin complet vers un autre fichier .ISO de suppléments invité à utiliser pour mettre à jour les suppléments invité.

--verbose Dit à VBoxManage d'être plus verbeux.

--wait-start Démarrer le processus normal de mise à jour et attend jusqu'à ce que la mise à jour des suppléments invité soit effectivement démarrée dans l'invité. Cela peut être nécessaire du fait de l'interaction nécessaire avec l'OS invité pendant la phase d'installation.

Si vous ne mettez pas ce drapeau, VBoxManage attendra que toute la mise à jour des suppléments invité s'achève.

8.32 VBoxManage debugvm

Les commandes “debugvm” sont pour les experts qui veulent jouer avec les détails exacts de l’exécution d’une machine virtuelle. Comme le débogueur de la VM décrit au chapitre [12.1.3, Le débogueur de VM intégré](#), page [210](#), ces commandes ne sont utiles que si vous êtes très familier des détails l’architecture d’une VM et de la façon de déboguer un logiciel.

Les sous-commandes de “debugvm” fonctionnent sur une machine virtuelle en fonction. Celles suivantes sont disponibles :

- Avec `dumpguestcore --filename <nom>`, vous pouvez créer un fichier de renvoi système de la VM en fonction, qui sera écrit dans le fichier donné. Ce fichier aura un format ELF cœur standard (avec des sections personnalisées) ; voir chapitre [12.1.4, Format du cœur d’une VM](#), page [212](#).

Cela correspond à la commande `writecore` du débogueur.

- La commande `info` est utilisée pour afficher des éléments d’info liés à la VMM, aux émulations de périphérique et aux pilotes associés. Cette commande prend un ou deux arguments : le nom de l’élément d’info, éventuellement suivi d’une chaîne contenant les arguments spécifiques à l’élément d’info. L’élément d’info `help` fournit une liste des éléments disponibles et des astuces sur les arguments supplémentaires.

Cela correspond à la commande `info` du débogueur.

- La commande `injectnmi` provoque une interruption non masquable (non-maskable interrupt (NMI) dans l’invité, qui pourrait être utile dans certains scénarios de débogage. Ce qui se passe exactement dépend du système d’exploitation invité mais une NMI peut planter tout le système d’exploitation invité. Ne l’utilisez pas, sauf si vous savez ce que vous faites.

- La commande `osdetect` lance la fonction de (re)-détectio[n] du débogueur de VMM du système d’exploitation de l’invité.

Cela correspond à la commande `detect` du débogueur.

- La commande `osinfo` est utilisée pour afficher des infos sur le système d’exploitation (OS) détecté par le débogueur de la VM.

- La commande `getregisters` est utilisée pour afficher le registre du processeur et des périphériques. La commande prend une liste des registres, chacun ayant une des formes suivantes :

- `réglage-registre.nom-registre.sous-champ`
- `réglage-registre.nom-registre`
- `nom-registre-processeur.sous-champ`
- `nom-registre-processeur`
- `all`

La forme `all` affichera tous les registres (pas les sous-champs). Les noms des registres sont indifférents à la casse. Quand vous interrogez un registre du processeur, vous pouvez ne pas mettre le réglage du registre, il sera sélectionné en utilisant la valeur de l’option `--cpu` (0 par défaut).

- La commande `setregisters` est utilisée pour modifier les registres des périphériques et du processeur. La commande prend une liste d’affectations de registres, chacune ayant une des formes suivantes :

- `réglage-registre.nom-registre.sous-champ=valeur`
- `réglage-registre.nom-registre=valeur`

- nom-registre-processeur.sous-champ=valeur
- nom-registre-processeur=valeur

Le format de la valeur devrait être du même genre que ce qu'affiche `getregisters`, sauf qu'on peut utiliser des octets et des décimales au lieu de l'hexadécimal. Le nommage du registre et le réglage par défaut du registre processeur sont gérés de la même manière qu'avec la commande `getregisters`.

- La commande `statistics` peut être utilisée pour afficher des statistiques VMM sur la ligne de commandes. L'option `--reset` réinitialisera les statistiques. Les statistiques concernées sont filtrables avec l'option `--pattern` qui accepte des jokers en style DOS/NT (? et *).

8.33 VBoxManage metrics

Cette commande supporte la surveillance de l'utilisation des ressources système. Les ressources sont représentées par différentes métriques associées au système hôte ou à une VM en particulier. Par exemple, le système hôte a une métrique `CPU/Load/User` qui affiche le pourcentage du temps proccleur passé à tourner en mode utilisateur sur une période spécifique.

Les données de métrique sont rassemblées et stockées en interne ; vous pouvez les récupérer n'importe quand avec la sous-commande `VBoxManage metric query`. Les données sont disponibles tant que le processus en tâche de fond `VBoxSVC` process is est vivant. Ce processus ce termine peu après que toutes les VMs et les interfaces ont été fermées.

Par défaut, aucune métrique n'est enregistrée. La récupération de métriques ne commence pas avant que `VBoxManage metrics setup` ne soit appelé avec un bon intervalle modèle et que le nombre de métriques n'a été conservé. On mesure l'intervalle en secondes. Par exemple, pour activer la récupération des métriques de l'utilisation du processeur et de la mémoire hôte toutes les secondes, en gardant les 5 échantillons les plus actuels, vous pouvez utiliser la commande suivante :

```
VBoxManage metrics setup --period 1 --samples 5 host CPU/Load,RAM/Usage
```

Le recueil de métriques ne peut être activé que sur des VMs démarrees. Les données et les paramètres de récupération d'une VM en particulier disparaîtront dès que vous éteindrez la VM. Utilisez la sous-commande `VBoxManage metrics list` pour voir les métriques actuellement disponibles. Vous pouvez aussi utiliser l'option `--list` avec une sous-commande modifiant les paramètres de métrique pour chercher les métriques concernées.

Remarquez que la sous-commande `VBoxManage metrics setup` désactive tous les échantillons qui peuvent avoir été recueillis pour un jeu d'objets et de métriques spécifique.

Pour activer ou désactiver la récupération des métriques sans perdre les données, vous pouvez utiliser les sous-commandes `VBoxManage metrics enable` et `VBoxManage metrics disable`. Remarquez que ces sous-commandes attendent en paramètres des métriques, pas des sous-métriques, comme `CPU/Load` ou `RAM/Usage`. En d'autres termes, l'activation de `CPU/Load/User` tout en désactivant `CPU/Load/Kernel` n'est pas supporté.

L'hôte et les VMs ont différents ensembles de métriques associées. Vous pouvez lister les métriques disponibles avec `VBoxManage metrics list` subcommand.

Un nom de métrique complet peut inclure une fonction d'agrégat. Le nom prend la forme suivante : `Catégorie/Métrique[/SousMétrique][:aggrégé]`. Par exemple, `RAM/Usage/Free:min` signifie la quantité minimum de mémoire disponible parmi toutes les données en mémoire s, cela s'applique à l'objet hôte.

Les sous-commandes peuvent s'appliquer à tous les objets et les métriques ou se limiter à un objet et/ou une liste de métriques. Si vous ne donnez aucun objet ni métrique en paramètres, les sous-commandes s'appliqueront à toutes les métriques disponibles de tous les objets. Vous pouvez utiliser une astérisque ("*") pour spécifier explicitement que la commande devrait s'appliquer à tous les objets ou à toutes les métriques. Utilisez "host" en nom d'objet pour limiter la cible

de la commande aux métriques relatives à l'hôte. Pour limiter la cible à un sous-ensemble de métriques, utilisez une liste de noms séparés par des virgules.

Par exemple, pour interroger les données de métriques sur le temps passé par le processeur en modes utilisateur et noyau, sur la machine virtuelle nommée “test”, vous pouvez utiliser la commande suivante :

```
VBoxManage metrics query test CPU/Load/User,CPU/Load/Kernel
```

La liste suivante résume les sous-commandes disponibles :

list Cette sous-commande affiche les paramètres des métriques existant actuellement. Remarquez que les métriques spécifiques à la VM ne sont disponibles que quand une VM en particulier est en fonction.

setup Cette sous-commande définit l'intervalle entre deux prises d'échantillons des données de métriques et le nombre d'échantillons gardés en interne. Les données mémorisées sont disponibles en affichage avec la sous-commande **query**. L'option **--list** affiche les métriques qui ont été modifiées dans le résultat de l'exécution de la commande.

enable Cette sous-commande “réactive le recueil de données après qu'il a été arrêté avec la sous-commande **disable**. Remarquez que la spécification de sous-métriques comme paramètres n'activera pas les métriques sous-jacentes. Utilisez **--list** pour savoir si la commande a fait ce que vous aviez prévu.

disable Cette sous-commande “suspends” le recueil des données sans changer les paramètres de récupération ni altérer les données recueillies. Remarquez que la spécification de sous-métriques en paramètres ne désactivera pas les métriques sous-jacentes. Utilisez **--list** pour voir si la commande a fait ce que vous aviez prévu.

query Cette sous-commande récupère et affiche les données de métriques mémorisées.

Note: La sous-commande **query** subcommand ne supprime pas ni ne remet à 0 les données mémorisées. Si vous interrogez assez souvent, vous verrez la durée d'évacuation graduelle des échantillons par les nouveaux échantillons.

collect Cette sous-commande définit l'intervalle entre la prise de deux échantillons de données de métrique et le nombre d'échantillons mémorisés en interne. Les données recueillies s'affichent périodiquement jusqu'à ce que vous appuyiez sur Ctrl-C, sauf si vous spécifiez l'option **--detach**. Avec l'option **--detach**, cette sous-commande agit de la même façon que **setup**. L'option **--list** affiche les métriques correspondant au filtre spécifié.

8.34 VBoxManage hostonlyif

Avec “hostonlyif”, vous pouvez changer la configuration de l'IP d'une interface réseau host-only. Pour une description du réseau host-only, merci de vous reporter au chapitre [6.6, Réseau Host-only](#), page [104](#). Chaque interface host-only par un nom et peut soit utiliser le serveur DHCP interne, soit voir son adresse IP configurée à la main (IP4 et IP6).

8.35 VBoxManage dhcpserver

Les commandes de “dhcpserver” vous permettent de contrôler le serveur DHCP construit dans VirtualBox. Vous pourriez trouver cela utile si vous utilisez le réseau interne ou host-only. (En

théorie, vous pouvez l'activer aussi pour un réseau bridgé, mais cela provoquera probablement des conflits avec d'autres serveurs DHCP de votre réseau physique.)

Utilisez les options suivantes de la ligne de commandes :

- Si vous utilisez le réseau interne pour l'adaptateur réseau virtuel d'une machine virtuelle, utilisez `VBoxManage dhcpserver add --netname <nom_réseau>`, où `<nom_réseau>` est le même nom de réseau que celui que vous avez utilisé avec `VBoxManage modifyvm <nomvm> --intnet<X> <nom_réseau>`.
- Si vous utilisez le réseau interne pour l'adaptateur réseau virtuel d'une machine virtuelle, utilisez plutôt `VBoxManage dhcpserver add --ifname <hostonly_if_name>`, où `<hostonly_if_name>` est le même nom de l'interface host-only que vous avez utilisé avec `VBoxManage modifyvm <nomvm> --hostonlyadapter<X> <hostonly_if_name>`.

Vous pouvez aussi utiliser l'option `--netname` comme avec les réseaux internes si vous connaissez le nom du réseau host-only ; vous pouvez voir les noms avec `VBoxManage list hostonlyifs` (voir chapitre 8.4, [VBoxManage list](#), page 127 ci-dessus).

Les paramètres suivants supplémentaires sont requis quand vous ajoutez un serveur DHCP pour la première fois :

- Avec `--ip`, spécifiez l'adresse IP du serveur DHCP lui-même.
- Avec `--netmask`, spécifiez le masque du réseau.
- Avec `--lowerip` et `--upperip`, vous pouvez spécifier respectivement l'adresse la plus basse et la plus haute que le serveur DHCP attribuera aux clients.

Enfin, vous devez spécifier `--enable`, sinon le serveur DHCP sera créé en état désactivé, ce qui ne fera rien.

Après cela, VirtualBox démarrera automatiquement le serveur DHCP pour le réseau interne ou host-only donné dès que la première machine virtuelle utilisant ce réseau sera démarrée.

Inversement, utilisez `VBoxManage dhcpserver remove` avec `--netname <nom_réseau>` ou `--ifname <hostonly_if_name>` pour supprimer le serveur DHCP, de nouveau pour le réseau interne ou host-only donné.

Pour modifier les paramètres d'un serveur DHCP créé précédemment avec `VBoxManage dhcpserver add`, vous pouvez utiliser `VBoxManage dhcpserver modify` pour une interface réseau interne ou host-only donnée.

8.36 VBoxManage extpack

La commande “extpack” vous permet d'ajouter ou de supprimer des packs d'extension de VirtualBox comme décrit au chapitre 1.5, [Installer VirtualBox et les packs d'extension](#), page 17.

- `$$`

Pour ajouter un nouveau pack d'extension, utilisez `VBoxManage extpack install <.vbox-extpack>`. Cette commande échouera si une version plus ancienne du même pack d'extension est déjà installée. Vous pouvez utiliser le paramètre optionnel `--replace` pour désinstaller l'ancien paquet avant d'installer le nouveau paquet..

- Pour supprimer un pack d'extension précédemment installé, utilisez `VBoxManage extpack uninstall <nom>`. Vous pouvez utiliser `VBoxManage list extpacks` pour afficher les noms des pack d'extensions actuellement installés ; merci de voir aussi chapitre 8.4, [VBoxManage list](#), page 127. Vous pouvez utiliser le paramètre optionnel `--force` pour surpasser le refus de se désinstaller d'un pack d'extension.

- La commande `VBoxManage extpack cleanup` peut être utilisée pour supprimer temporairement les fichiers et les répertoires qui peuvent avoir été laissés après l'échec d'une précédente commande d'installation ou de désinstallation.

Les commandes suivantes montrent des exemples d'une liste des packs d'extension et de la manière d'en supprimer un :

```
$ VBoxManage list extpacks
pack d'extensions: 1
Pack no. 0: Oracle VM VirtualBox pack d'extension
Version: 4.1.12
Revision: 77218
Edition:
Description: USB 2.0 Host Controller, VirtualBox RDP, PXE ROM with E1000 support.
VRDE Module: VBoxVRDP
Usable: true
Why unusable:
$ VBoxManage extpack uninstall "Oracle VM VirtualBox pack d'extension"
0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%
Successfully uninstalled "Oracle VM VirtualBox pack d'extension".
```

9 Sujets avancés

9.1 VBoxSDL, l'afficheur simplifié de VM

9.1.1 Introduction

VBoxSDL est une interface graphique (GUI) simple qui élimine le support du clicodrome fourni par VirtualBox, notre principale GUI. VBoxSDL est utilisé actuellement d'abord pour déboguer VirtualBox, donc il n'est pas officiellement supporté. Vous pouvez quand même le trouver utile pour des environnements où les machines virtuelles ne sont pas nécessairement contrôlées par la même personne qui utilise la machine virtuelle.

Note: VBoxSDL n'est pas disponible sur la plateforme hôte Mac OS X.

Comme nous pouvons le voir sur l'impression d'écran suivante, VBoxSDL ne fournit vraiment qu'une fenêtre simple ne contenant que la machine virtuelle "pure", sans menus ni contrôleurs sur lesquels cliquer et sans indicateurs supplémentaires sur l'activité de la VM :



Pour démarrer une machine virtuelle avec VBoxSDL au lieu de l'interface graphique de VirtualBox, tapez ce qui suit sur une ligne de commandes :

```
VBoxSDL --startvm <vm>
```

où <vm> est, comme d'habitude dans les paramètres en ligne de commande de VirtualBox, le nom ou l'UUID d'une machine virtuelle existante.

9.1.2 Étiquetage sécurisé avec VBoxSDL

Quand vous lancez des systèmes d'exploitation invités en mode plein écran, le système d'exploitation invité a en général le contrôle de tout l'écran. Cela pourrait représenter un risque

de sécurité car le système d'exploitation invité pourrait, pour l'utilisateur, lui faire croire qu'il est vraiment dans un autre système (qui pourrait avoir un haut niveau de sécurité), ou lui faire assimiler des messages à l'écran comme provenant du système d'exploitation hôte.

Afin de protéger l'utilisateur contre les risques de sécurité précédents, on a développé la fonction d'étiquetage de sécurité. L'étiquetage de sécurité n'est actuellement disponible que pour VBoxSDL. S'il est activé, une partie de la zone d'affichage est réservée à une étiquette où est affiché un message défini par l'utilisateur. La hauteur de l'étiquette est définie à 20 pixels dans VBoxSDL. La couleur de la police et de l'arrière-plan de l'étiquette peuvent éventuellement être définies en valeurs de couleurs RGB hexadécimales. On utilise la syntaxe suivante pour activer l'étiquetage de sécurité :

```
VBoxSDL --startvm "nom VM"
  --securelabel --seclabelfnt ~/fonts/arial.ttf
  --seclabelsiz 14 --seclabelfgcol 00FF00 --seclabelbgcol 00FFFF
```

Outre l'activation de l'étiquette de sécurité, il faut fournir une police TrueType. Pour utiliser une autre taille de police que 12 points, utilisez le paramètre `--seclabelsiz`.

Vous pouvez définir le texte de l'étiquette avec

```
VBoxManage setextradata "nom VM" "VBoxSDL/SecureLabel" "L étiquette"
```

Une modification de cette étiquette prendra effet immédiatement.

En général, les résolutions du plein écran sont limitées à certaines géométries "standards" telles que 1024 x 768. Une augmentation de vingt lignes n'est en général pas faisable, donc dans la plupart des cas, VBoxSDL choisira la résolution suivante la plus élevée comme 1280 x 1024 et l'écran de l'invité ne couvrira pas toute la zone d'affichage. Si VBoxSDL ne peut pas choisir de résolution plus élevée, l'étiquette de sécurité sera dessinée en haut de la zone de l'écran de l'invité. Pour surmonter le problème selon lequel le bas de l'écran de l'invité est caché, VBoxSDL peut fournir des modes graphiques personnalisés à l'invité, réduits par la hauteur de l'étiquette. Pour les invités Windows et ceux Solaris et Linux récents, les suppléments invité de VirtualBox fournissent automatiquement les modes graphiques réduits. De plus, le BIOS VESA a été ajusté pour dupliquer sa table en mode standard avec des résolutions ajustées. Les IDs du mode ajusté se calculent en utilisant la formule suivante :

```
reduced_modeid = modeid + 0x30
```

Par exemple, pour démarrer Linux avec 1024 x 748 x 16, le mode standard 0x117 (1024 x 768 x 16) est utilisé de base. Le paramètre du noyau Linux du mode graphique se calcule alors en faisant :

```
vga = 0x200 | 0x117 + 0x30
vga = 839
```

On duplique les modes standards au lieu de ne fournir que les modes ajustés car la plupart des systèmes d'exploitation invités ont besoin des modes VESA standards figés et ils refusent de démarrer avec d'autres modes.

Quand vous utilisez le pilote VESA de X.org, il faut calculer les modes personnalisés et les ajouter à la main à la configuration (en général, dans /etc/X11/xorg.conf). Vous pouvez trouver un outil à la main pour déterminer les entrées des modes sur <http://www.tkk.fi/Misc/Electronics/faq/vga2rgb/calc.html>.)

9.1.3 Libérer les modificateurs avec VBoxSDL sur Linux

Quand vous basculez d'un terminal virtuel (VT) X à un autre en utilisant Ctrl-Alt-Fx pendant que la fenêtre VBoxSDL contient le focus d'entrée, l'invité recevra les événements d'appui sur Ctrl et Alt sans recevoir les événements de relâchement correspondant de la touche. C'est une limite liée à l'architecture de Linux. Pour réinitialiser les touches modificaterices, il est possible

d'envoyer SIGUSR1 au fil principal de VBoxSDL (première entrée de la liste ps). Par exemple, quand vous basculez vers un autre VT et quand vous enregistrez la machine virtuelle à partir de ce terminal, la séquence suivante peut être utilisée pour s'assurer que la VM sauvegardée avec des modificateurs bloqué :

```
kill -usr1 <pid>
VBoxManage controlvm "Windows 2000" savestate
```

9.2 Identifications automatiques dans l'invité

VirtualBox fournit des modules invité supplémentaires pour Windows, Linux et Solaris pour activer l'identification automatique dans l'invité.

Quand on lance un système d'exploitation dans une machine virtuelle, il pourrait être souhaitable d'effectuer des identifications automatiques et coordonnées en utilisant des autorisations issues d'un système d'identification maître. (Avec les “autorisations”, on se réfère aux informations d'identification qui consistent dans le nom d'utilisateur, le mot de passe et le nom de domaine, où chaque valeur pourrait être vide.)

9.2.1 Identification automatique dans un invité Windows

Depuis Windows NT, Windows fourni un sous-système d'identification modulaire (“Winlogon”) qu'on peut utiliser et étendre par ce qu'on appelle des modules GINA (Graphical Identification and Authentication). Avec Windows Vista et Windows 7, les modules GINA ont été remplacés par un nouveau mécanisme appelé “fournisseurs d'autorisations”. Les suppléments invité de VirtualBox pour Windows sont fournis à la fois avec un module GINA et un fournisseur d'autorisations, ils permettent donc à n'importe quel invité Windows d'effectuer des identifications automatiques.

Pour activer le module GINA ou fournisseur d'autorisations des suppléments invité de VirtualBox, installez les suppléments invité en utilisant le paramètre `/with_autologon` en ligne de commande. Toutes les étapes manuelles suivantes exigés pour installer ces modules se feront via l'installeur.

Pour installer à la main le module GINA de VirtualBox, extrayez les suppléments invité (voir chapitre [4.2.1.4, Extraction manuelle du fichier](#), page [69](#)) et copiez le fichier `VBoxGINA.dll` dans le répertoire Windows `SYSTEM32`. Puis, dans le registre, créez la clé suivante :

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\GinaDLL
```

avec la valeur `VBoxGINA.dll`.

Note: Le module GINA de VirtualBox est implémenté sur le module GINA standard de Windows (`MSGINA.DLL`). En conséquence, il ne fonctionnera vraisemblablement pas avec des modules GINA tiers.

Pour installer à la main le module fournisseur d'autorisation de VirtualBox, extrayez les suppléments invité (voir chapitre [4.2.1.4, Extraction manuelle du fichier](#), page [69](#)) et copiez le fichier `VBoxCredProv.dll` dans le répertoire Windows `SYSTEM32`. Puis, dans le registre, créez les clés suivantes :

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}
HKEY_CLASSES_ROOT\CLSID\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}
HKEY_CLASSES_ROOT\CLSID\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}\InprocServer32
```

avec pour valeurs celles par défaut (la clé nommée (Default) dans chaque clé) définies sur VBoxCredProv. Après quoi, il faut créer une nouvelle chaîne nommée

```
HKEY_CLASSES_ROOT\CLSID\{275D3BCC-22BB-4948-A7F6-3A3054EBA92B}\InprocServer32\ThreadingModel
```

avec une valeur de Apartment.

Pour définir les autorisations, utilisez la commande suivante sur une VM *en fonction* :

```
VBoxManage controlvm "Windows XP" setcredentials "John Doe" "secretpassword" "DOMTEST"
```

Pendant que la VM est en fonction, vous pouvez chercher les autorisations accordées par les modules d'identification de VirtualBox (GINA ou fournisseur d'autorisation) en utilisant le périphérique des suppléments invité de VirtualBox. Quand Windows est en mode "déconnecté", les modules d'identification chercheront constamment les autorisations et si elles existent, il tentera une identification. Après avoir récupéré les autorisations, les modules d'identification les écraseront pour que la commande ci-dessus doive se répéter pour les identifications consécutives.

Pour des raisons de sécurité, les autorisations ne sont pas stockées de façon permanente et vous les perdrez quand vous redémarrez la VM. En outre, les autorisations sont en "écriture seule", c'est-à-dire qu'il n'y a aucun moyen de récupérer les autorisations côté hôte. Vous pouvez réinitialiser les autorisations côté hôte en définissant des valeurs vides.

Selon la variante particulière de votre invité Windows, les restrictions suivantes s'appliquent :

1. Pour les **invités Windows XP**, le sous-système d'identification doit être configuré pour utiliser la boîte de dialogue classique d'identification car le module GINA de VirtualBox ne supporte pas la boîte de dialogue de bienvenue à la XP.
2. Pour les **invités Windows Vista, Windows 7 et Windows 8**, le sous-système d'identification ne supporte pas ce qu'on appelle la Secure Attention Sequence (CTRL+ALT+DEL). Il s'en suit que les paramètres des règles du groupe de l'invité doivent être modifiés pour ne pas utiliser la Secure Attention Sequence. De plus, le nom d'utilisateur donné n'est comparé qu'au vrai nom d'utilisateur, pas au nom convivial d'utilisateur. Cela veut dire que quand vous renommez un utilisateur, vous devez aussi fournir le nom d'utilisateur originel (en interne, Windows ne renomme jamais les comptes utilisateurs).
3. La gestion de l'identification automatique du Windows Remote Desktop Service (connu jadis sous le nom Terminal Services) est désactivée par défaut. Pour l'activer, créez la clé de registre

```
HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\VirtualBox Guest Additions\AutoLogon
```

avec une valeur DWORD de 1.

La commande suivante oblige VirtualBox à garder les autorisations après leur lecture par l'invité et au redémarrage de la VM :

```
VBoxManage setextradata "Windows XP" VBoxInternal/Devices/VMMDev/0/Config/KeepCredentials 1
```

Remarquez que c'est un risque de sécurité potentiel car une application malveillante en fonction sur l'invité pourrait solliciter ces informations en utilisant la bonne interface.

9.2.2 Identifications automatisées à un invité Linux/Unix

À partir de la version 3.2, VirtualBox fournit un module PAM personnalisé (Pluggable Authentication Module) qu'on peut utiliser pour effectuer des identifications automatiques dans l'invité sur des plateformes qui supportent cet environnement. Virtuellement, toutes les distributions Linux/Unix modernes s'appuient sur PAM.

Pour les identifications automatiques sur des distributions Ubuntu (ou des dérivées d'Ubuntu) en utilisant LightDM comme gestionnaire d'affichage, merci de voir chapitre [9.2.2.1, VirtualBox Greeter pour Ubuntu / LightDM](#), page [166](#).

Le module `pam_vbox.so` lui-même **ne fait pas** de vérification effective des autorisations passées à l'OS invité ; il s'appuie plutôt sur d'autres modules tels que `pam_unix.so` ou `pam_unix2.so` dans la pile PAM pour faire la validation effective en utilisant les autorisations récupérées par `pam_vbox.so`. Dès lors, il faut que `pam_vbox.so` soit en haut de la liste d'authentification du service PAM.

Note: `pam_vbox.so` ne supporte que le `auth` primitif. D'autres primates tels que `account`, `session` ou `password` ne sont pas supportés.

Le module `pam_vbox.so` est inclus dans les suppléments invité mais il n'est pas installé et/ou activé par défaut sur l'OS invité. Afin de l'installer, il faut le copier de `/opt/VBoxGuestAdditions-<version>/lib/VBoxGuestAdditions/` dans le répertoire des modules de sécurité, en général `/lib/security/` sur les invités Linux 32 bit ou `/lib64/security/` sur ceux 64 bits. Merci de vous reporter à la documentation de votre OS invité pour le bon répertoire du module PAM.

Par exemple, pour utiliser `pam_vbox.so` avec un OS invité Linux Ubuntu et GDM (le GNOME Desktop Manager) pour identifier les utilisateurs automatiquement avec les droits passés par l'hôte, l'OS invité doit être configuré comme ce qui suit :

1. Le module `pam_vbox.so` doit être copié dans le répertoire des modules de sécurité, dans ce cas, c'est `/lib/security`.
2. Éditez le fichier de configuration de PAM avec GDM qui se trouve dans `/etc/pam.d/gdm`, en ajoutant la ligne `auth requisite pam_vbox.so` au début. En outre, dans la plupart des distributions Linux, il existe un fichier appelé `/etc/pam.d/common-auth`. Ce fichier est inclus dans de nombreux services (comme le fichier GDM indiqué ci-dessus). Vous devez y ajouter la ligne `auth requisite pam_vbox.so`.
3. Si vous voulez une authentification contre la base de données shadow en utilisant `pam_unix.so` ou `pam_unix2.so`, l'argument `try_first_pass` de `pam_unix.so` ou `use_first_pass` pour `pam_unix2.so` est nécessaire pour passer les autorisations du module VirtualBox au module d'authentification de la base de données shadow. Pour Ubuntu, il faut ajouter cela à `/etc/pam.d/common-auth`, à la fin de la ligne référençant `pam_unix.so`. Cet argument dit au module PAM d'utiliser les autorisations déjà présentes dans la pile, à savoir celles fournies par le module PAM de VirtualBox.

Avertissement: Une pile PAM mal configurée peut vraiment vous empêcher de vous connecter à votre système invité !

Pour faciliter le déploiement, vous pouvez passer l'argument `debug` juste après la ligne `pam_vbox.so`. La sortie du journal de débogage sera enregistrée en utilisant `syslog`.

Note: Par défaut, `pam_vbox` n'attendra pas les autorisations venant de l'hôte, en d'autres termes : quand une invite de connexion s'affiche (par exemple via GDM/KDM ou la console texte) et quand `pam_vbox` n'a pas encore les autorisations, il n'attend pas qu'elles viennent. Le module suivant de la pile PAM (selon la configuration de PAM) aura une chance d'authentification.

À partir de VirtualBox 4.1.4 pam_vbox supporte plusieurs paramètres de propriétés d'invité résidant tous dans `/VirtualBox/GuestAdd/PAM/`. Ces paramètres permettent à pam_vbox d'attendre que les autorisations soient fournies dans l'hôte et, éventuellement, il peut afficher un message tout en les attendant. Les propriétés d'invité suivantes peuvent être définies :

1. **CredsWait** : Définissez sur “1” si pam_vbox devrait commencer à attendre jusqu'à ce que les autorisations viennent de l'hôte. En attendant, aucune autre méthode d'authentification comme la connexion à la main ne sera disponible. Si cette propriété est vide ou effacée, les autorisations ne seront pas attendues et pam_vbox comme avant (voir le paragraphe ci-dessus). Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
2. **CredsWaitAbort** : Annule l'attente des autorisations si une valeur est définie. Elle peut être définie à partir de l'hôte et de l'invité.
3. **CredsWaitTimeout** : Timeout (en secondes) pendant lequel il faut laisser pam_vbox attendre les autorisations. Si aucune autorisation ne vient dans ce délai, l'authentification de pam_vbox sera définie comme échouée et le prochain module PAM de la chaîne sera appelé. Si vous ne spécifiez pas cette propriété, ou que vous la réglez sur “0” ou sur une valeur invalide, on utilisera un timeout infini. Cette propriété doit être paramétrée en lecture seule pour l'invité (RDONLYGUEST).

Pour personnaliser davantage pam_vbox, il existe les propriétés invité suivantes :

1. **CredsMsgWaiting** : message personnalisé affiché pendant que pam_vbox attend les autorisations de l'hôte. Cette propriété doit être réglée en lecture seule pour l'invité (RDONLYGUEST).
2. **CredsMsgWaitTimeout** : message personnalisé affiché pendant l'attente de la fin du timeout des autorisations de pam_vbox, par exemple si elles ne sont pas arrivées à temps. Cette propriété doit être réglée en lecture seule pour l'invité (RDONLYGUEST).

Note: Si une propriété pam_vbox est définie avec de mauvais drapeaux (RDONLYGUEST), cette propriété sera ignorée et - selon la propriété - une valeur par défaut sera utilisée. Il peut s'en suivre que pam_vbox n'attendra pas les autorisations. Consultez le fichier syslog adéquat pour plus d'informations et utilisez l'option debug.

9.2.2.1 VirtualBox Greeter pour Ubuntu / LightDM

À partir de la version 4.2.6, VirtualBox est fourni avec un module d'accueil propre qui s'appelle vbox-greeter et qu'on peut utiliser avec LightDM 1.0.1 ou supérieur. LightDM est le gestionnaire d'affichage par défaut depuis Ubuntu 10.11 et, donc, on peut aussi l'utiliser pour des identifications automatiques dans l'invité.

vbox-greeter n'a pas besoin du module pam_vbox décrit ci-dessus pour fonctionner – il est fourni avec son propre mécanisme d'authentification fourni par LightDM. Cependant, pour offrir une flexibilité maximum, vous pouvez utiliser les deux modules ensemble sur le même invité.

Comme pour le module pam_vbox, vbox-greeter fait partie des suppléments invité mais il n'est pas installé et/ou activé sur l'OS invité par défaut. Pour installer automatiquement vbox-greeter lors de l'installation des suppléments invité, utilisez le paramètre `--with-autologon` au démarrage du fichier VBoxLinuxAdditions.run :

```
# ./VBoxLinuxAdditions.run --with-autologon
```

Pour une installation manuelle ou a posteriori, il faut copier le fichier `vbox-greeter.desktop` de `/opt/VBoxGuestAdditions-/shared/VBoxGuestAdditions/` dans le répertoire `xgreeters`, généralement `/usr/share/xgreeters/`. Merci de vous reporter à la documentation de votre OS invité pour voir le bon répertoire du module d'accueil de LightDM.

Le module `vbox-greeter` lui-même a déjà été installé par l'installeur des suppléments invité de VirtualBox et il se trouve dans `/usr/sbin/`. Pour activer `vbox-greeter` comme module d'accueil standard, le fichier `/etc/lightdm/lightdm.conf` doit être modifié :

```
[SeatDefaults]
greeter-session=vbox-greeter
```

Note: Le serveur LightDM doit être totalement relancé pour que `vbox-greeter` soit utilisé comme accueil par défaut. En tant qu'administrateur, faites un service `lightdm --full-restart` sur Ubuntu, ou redémarrez simplement l'invité.

Note: `vbox-greeter` est indépendant de la session graphique choisie par l'utilisateur (comme Gnome, KDE, Unity etc). Par contre, il exige FLTK 1.3 pour afficher son interface utilisateur.

Outre les valeurs de la propriété invité du module `pam_vbox` listées ci-dessous, `vbox-greeter` inclut les propriétés invité suivantes pour mieux personnaliser son interface, lesquelles se trouvent dans `/VirtualBox/GuestAdd/Greeter/` :

1. `HideRestart` : Mettez-le sur “1” si `vbox-greeter` doit cacher le bouton de redémarrage pour redémarrer l'invité. Vous devez définir cette propriété en lecture seule pour l'invité (RDONLYGUEST).
2. `HideShutdown` : Passez-la à “1” si `vbox-greeter` devrait cacher le bouton d'extinction de l'invité. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
3. `BannerPath` : Chemin d'un fichier .PNG pour en faire une bannière en haut. La taille de l'image doit être 460 x 90 pixels, n'importe quelle profondeur de bits. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
4. `UseTheming` : Passez-la à “1” pour activer les options de thème suivantes. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
5. `Theme/BackgroundColor` : Couleurs RRGGBB hexadécimales de fond. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
6. `Theme/LogonDialog/HeaderColor` : Couleur RRGGBB hexadécimale du texte d'en-tête. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
7. `Theme/LogonDialog/BackgroundColor` : Couleur RRGGBB hexadécimale du fond des boîtes de dialogue. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).
8. `Theme/LogonDialog/ButtonColor` : Couleur RRGGBB hexadécimale du fond du bouton de la boîte de dialogue d'identification. Cette propriété doit être définie en lecture seule pour l'invité (RDONLYGUEST).

Note: Les mêmes restrictions des propriétés invité ci-dessus s'appliquent comme celles indiquées dans la section `pam_vbox`.

9.3 Configuration avancées pour les invités Windows

9.3.1 Préparation automatique du système Windows

À partir de Windows NT 4.0, Microsoft offre un outil “préparation système” (en bref : Sysprep) pour préparer un système Windows à être déployé ou redistribué. Si Windows 2000 et XP sont inclus avec Sysprep sur leur média d’installation, l’outil est également disponible en téléchargement sur le site Internet de Microsoft. Dans une installation standard de Windows Vista et 7, Sysprep est déjà inclus. Sysprep consiste principalement dans un exécutable qui s’appelle `sysprep.exe` qui est appelé par l’utilisateur pour passer l’installation Windows en mode préparation.

À partir VirtualBox 3.2.2, les suppléments invité offrent un moyen de lancer une préparation du système sur le système d’exploitation invité de manière automatisée et contrôlée depuis le système hôte. Pour faire cela, voir chapitre [4.7, Contrôle de l’invité](#), page [82](#) pour utiliser la fonction avec l’identifiant spécial `sysprep` pour que le programme s’exécute avec le nom d’utilisateur `sysprep` et le mot de passe `sysprep` des autorisations. Sysprep se lance avec les droits système requis.

Note: La spécification de l’emplacement de “`sysprep.exe`” **n’est pas possible** – les chemins suivants seront plutôt utilisés (basés sur le système d’exploitation) :

- `C:\sysprep\sysprep.exe` pour Windows NT 4.0, 2000 et XP
- `%WINDIR%\System32\Sysprep\sysprep.exe` pour Windows Vista, 2008 Server et 7

Les suppléments invité utiliseront automatiquement le chemin adapté pour exécuter l’outil de préparation système.

9.4 Configuration avancée pour les invités Linux et Solaris

9.4.1 Paramétrage manuel des services sélectionnés sur l’invité Linux

Les suppléments invité de VirtualBox contiennent plusieurs pilotes. Si, pour une raison quelconque, vous ne souhaitez pas les installer, vous pouvez installer les suppléments invité en utilisant la commande suivante :

```
sh ./VBoxLinuxAdditions.run no_setup
```

Après quoi, vous devrez au moins compiler les modules noyau en lançant la commande

```
/usr/lib/VBoxGuestAdditions/vboxadd setup
```

en tant que root (vous devrez remplacer `lib` par `lib64` sur certains invités 64 bits), et sur les anciens invités sans service udev, vous devrez ajouter le service `vboxadd` au niveau d’exécution par défaut pour vous assurer que les modules sont chargés.

Pour régler le service de synchronisation du temps, lancez la commande

```
/usr/lib/VBoxGuestAdditions/vboxadd-service setup
```

et ajoutez le service `vboxadd-service` au niveau d’exécution par défaut. Pour paramétriser la partie X11 et OpenGL des suppléments invité, lancez la commande

```
/usr/lib/VBoxGuestAdditions/vboxadd-x11 setup
```

(vous n'avez pas besoin d'activer un service).

Pour recompiler les modules noyau invité, utilisez cette commande :

```
/usr/lib/VBoxGuestAdditions/vboxadd setup
```

Après la compilation, vous devriez redémarrer votre invité pour vous assurer que les nouveaux modules sont bien utilisés.

9.4.2 Paramétrage approfondi des pilotes graphique et souris de l'invité

Cette section suppose que vous êtes familier de la configuration de votre serveur X.Org en utilisant xorg.conf et éventuellement les mécanismes récents en utilisant hal ou udev et xorg.conf.d. Sinon, vous pouvez apprendre à les utiliser en étudiant la documentation fournie avec X.Org.

Les suppléments invité de VirtualBox sont fournis avec les pilotes pour les versions X.Org

- X11R6.8/X11R6.9 et XFree86 version 4.3 (vboxvideo_drv_68.o et vboxmouse_drv_68.o)
- X11R7.0 (vboxvideo_drv_70.so and vboxmouse_drv_70.so)
- X11R7.1 (vboxvideo_drv_71.so and vboxmouse_drv_71.so)
- Serveur X.Org versions 1.3 et later (vboxvideo_drv_13.so et vboxmouse_drv_13.so et ainsi de suite).

Par défaut, vous pouvez trouver ces pilotes dans le répertoire

`/opt/VBoxGuestAdditions-<version>/lib/VBoxGuestAdditions`

et les bonnes versions du serveur X sont liées de façon symbolique aux répertoires du pilote de X.Org.

Pour que l'intégration graphique fonctionne correctement, le serveur X doit charger le pilote vboxvideo (beaucoup de versions récentes du serveur X le cherchent automatiquement si elles voient qu'elles sont sur VirtualBox) et pour une expérience utilisateur optimale, les pilotes du noyau invité doivent être chargés et l'outil des suppléments invité VBoxClient doit être en fonction en tant que client dans la session X. Pour que l'intégration de la souris fonctionne correctement, les pilotes du noyau invité doivent être chargés et, au surplus, dans les serveurs X de X.Org X11R6.8 à X11R7.1 et dans XFree86 version 4.3, le bon pilote vboxmouse doit être chargé et associé à `/dev/mouse` ou `/dev/psaux` ; dans le serveur X.Org 1.3 ou supérieur, un pilote de souris PS/2 doit être chargé et le bon pilote vboxmouse doit être associé à `/dev/vboxguest`.

Le pilote graphique invité de VirtualBox peut utiliser n'importe quelle configuration graphique pour laquelle la résolution rentre dans la mémoire graphique affectée à la machine virtuelle (moins une petite quantité utilisée par le pilote invité) comme décrit au chapitre [3.5, Paramètres d'affichage](#), page 55. Le pilote offrira une gamme de nœuds standards allant au moins jusqu'à la résolution invité par défaut pour tous les écrans invités. Dans le serveur X.Org et supérieur, le mode par défaut peut être modifié en définissant la propriété de sortie VBOX_MODE sur "`<width>x<height>`" pour tout écran invité. Quand VBoxClient et les pilotes du noyau sont actifs, cela se fait automatiquement quand l'hôte demande une modification du mode. Le pilote des anciennes versions ne peut recevoir de nouveaux modes qu'en demandant à l'hôte les requêtes à intervalles réguliers.

Avec les serveurs X pre-1.3, vous pouvez également ajouter vos propres modes dans le fichier de configuration du serveur X. Vous devez simplement les ajouter à la liste des "Modes" de la sous-section "Display" de la section "Screen". Par exemple, la section affichée ici a un mode de résolution personnalisé de 2048x800 :

```
Section "Screen"
    Identifier      "Default Screen"
    Device          "VirtualBox graphics card"
    Monitor         "Generic Monitor"
    DefaultDepth    24
```

```

SubSection "Display"
    Depth           24
    Modes          "2048x800" "800x600" "640x480"
EndSubSection
EndSection

```

9.5 Montage de processeur à chaud

Quand des machines virtuelles fonctionnent sur des systèmes d'exploitation serveurs modernes, VirtualBox supporte le montage à chaud de processeur.¹ Alors que, sur un ordinateur physique, cela voudrait dire qu'un processeur peut être ajouté ou supprimé pendant que la machine fonctionne, VirtualBox supporte l'ajout et le retrait de processeurs virtuels pendant que la machine virtuelle est en fonction.

Le montage à chaud de processeur ne fonctionne qu'avec les systèmes d'exploitation invités qui le supportent. Jusque-là, il ne s'applique qu'à Linux et Windows Server 2008 x64 édition Data Center. Windows ne supporte que l'ajout à chaud alors que Linux supporte l'ajout et le retrait à chaud., mais pour utiliser cette fonction avec plus de 8 processeurs, il faut un invité Linux 64 bits.

Pour l'instant, le branchement à chaud d'un processeur exige d'utiliser l'interface en ligne de commandes VBoxManage. Tout d'abord, il faut activer le branchement à chaud pour une machine virtuelle :

```
VBoxManage modifyvm "nom VM" --cpuhotplug on
```

Ensuite, l'option --cpus spécifie le nombre maximum de processeurs que peut avoir la machine virtuelle :

```
VBoxManage modifyvm "nom VM" --cpus 8
```

Quand la VM est désactivée, vous pouvez ajouter et supprimer des processeurs virtuels avec les sous-commandes modifyvm --plugcpu et --unplugcpu, qui prend le nombre de processeurs virtuels en paramètre, comme ceci :

```
VBoxManage modifyvm "nom VM" --plugcpu 3
VBoxManage modifyvm "nom VM" --unplugcpu 3
```

Remarquez que le processeur 0 ne peut jamais être supprimé.

Pendant que la VM est en fonction, les processeurs peuvent être ajoutés avec les commandes controlvm plugcpu/unplugcpu :

```
VBoxManage controlvm "nom VM" plugcpu 3
VBoxManage controlvm "nom VM" unplugcpu 3
```

Voir chapitre 8.8, [VBoxManage modifyvm](#), page 130 et chapitre 8.13, [VBoxManage controlvm](#), page 140 pour des détails.

Avec des invités Linux, ce qui suit s'applique : Pour empêcher d'éjecter alors que le processeur est utilisé, il doit être éjecté de l'invité au préalable. Les suppléments invité pour Linux contiennent un service qui reçoit les événements de retrait à chaud et ils éjectent le processeur. De plus, après qu'un processeur a été ajouté à la VM, il n'est pas utilisé automatiquement par Linux. Le service des suppléments invité pour Linux s'en chargera s'il est installé. Sinon, vous pouvez démarrer un processeur avec la commande suivante :

```
echo 1 > /sys/devices/system/cpu/cpu<id>/online
```

¹Le support du montage de processeur à chaud a été introduit avec VirtualBox 3.2.

9.6 PCI passthrough

Sur des hôtes Linux, avec un noyau assez récent (au moins la version 2.6.31), le passthrough de périphériques PCI expérimental est disponible.²

Note: Le module PCI passthrough est inclu comme un paquet d'extension de VirtualBox, qui doit être installé séparément. Voir chapitre [1.5, Installer VirtualBox et les packs d'extension](#), page [17](#) pour plus d'informations.

Cette fonction vous permettra essentiellement d'utiliser directement les périphériques PCI physiques de l'hôte sur l'invité même si l'hôte n'a pas de pilote pour ce périphérique particulier. Tant les cartes PCI normales que certaines cartes PCI express sont supportées. L'AGP et certaines cartes PCI Express ne sont pas supportées pour l'instant si elles s'appuient sur l'unité de programmation GART (Graphics Address Remapping Table) pour la gestion des textures vu qu'il fait plutôt des opérations non triviales avec l'association de pages qui s'interfacent avec IOMMU. Il se peut que cette limite soit surmontée dans les prochaines versions.

Pour être totalement opérationnel, le support PCI passthrough de VirtualBox dépend d'une unité matérielle IOMMU qui n'est pas encore trop largement disponible. Si le périphérique utilise le bus mastering (à savoir qu'il fait sa propre DMA sur la mémoire de l'OS), une IOMMU est requise, sinon de telles transactions DMA peuvent écrire sur la mauvaise adresse physique de la mémoire car le moteur DMA du périphérique est programmé pour utiliser un protocole spécifique au périphérique pour faire des transactions avec la mémoire. Les fonctions IOMMU comme traduction des unités correspondant à la mémoire physique accèdent aux requêtes du périphérique en utilisant la connaissance de l'adresse physique de la mémoire de l'invité via les règles de traduction d'adresse physique de l'hôte.

La solution d'Intel pour IOMMU est vendue sous le nom "Intel Virtualization Technology for Directed I/O" (VT-d), et celle d'AMD s'appelle AMD-Vi. Merci donc de vérifier si le modèle de votre carte mère comporte la technologie adaptée. Même si votre matériel n'a pas d'IOMMU, certaines cartes PCI peuvent fonctionner (comme des adaptateurs série PCI), mais l'invité affichera un avertissement au démarrage et l'exécution de la VM s'achèvera si le pilote invité essaie d'activer le bus mastering.

Très couramment, le BIOS ou l'OS hôte désactive par défaut l'IOMMU. Donc avant d'essayer de l'utiliser, merci de vous assurer que

1. Votre carte mère a une unité IOMMU.
2. Votre processeur supporte l'IOMMU.
3. L'IOMMU est activé dans le BIOS.
4. La VM doit fonctionner avec VT-x/AMD-V et la pagination nested doit être activée.
5. Votre noyau Linux a été compilé avec le support IOMMU (y compris la réassociation du DMA, voir l'option de compilation `CONFIG_DMAR`). Le pilote PCI stub (`CONFIG_PCI_STUB`) est requis aussi.
6. Votre noyau Linux reconnaît et utilise l'unité IOMMU (l'option (de démarrage `intel_iommu=on` pourrait être nécessaire). Cherchez DMAR et PCI-DMA dans le journal du démarrage.

Une fois que vous êtes sûre que le noyau hôte supporte l'IOMMU, la prochaine étape est de sélectionner la carte PCI et de l'attacher à l'invité. Pour visualiser la liste des périphériques PCI disponibles, utilisez la commande `lspci`. La sortie ressemblera à ceci

²Le support expérimental pour le passthrough PCI a été introduit avec VirtualBox 4.1.

```
01:00.0 VGA compatible controller: ATI Technologies Inc Cedar PRO [Radeon HD 5450]
01:00.1 Audio device: ATI Technologies Inc Manhattan HDMI Audio [Mobility Radeon HD 5000 Series]
02:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller
03:00.0 SATA controller: JMicron Technology Corp. JMB362/JMB363 Serial ATA Controller (rev 03)
03:00.1 IDE interface: JMicron Technology Corp. JMB362/JMB363 Serial ATA Controller (rev 03)
06:00.0 VGA compatible controller: nVidia Corporation G86 [GeForce 8500 GT] (rev a1)
```

La première colonne est une adresse PCI (au format `bus:device.function`). Cette adresse pourrait être utilisée pour identifier les périphériques pour aller plus loin. Par exemple, pour attacher un contrôleur réseau PCI du système listé ci-dessus, au second bus PCI de l'invité, en périphériques 5, la fonction 0, utilisez la commande suivante :

```
VBoxManage modifyvm "nom VM" --pciattech 02:00.0@01:05.0
```

Pour détacher ce même périphérique, utilisez

```
VBoxManage modifyvm "nom VM" --pcidetach 02:00.0
```

Merci de remarquer que l'hôte et l'invité pourraient librement affecter une autre adresse PCI à la carte attachée à l'exécution, donc ces adresses ne s'appliquent qu'à l'adresse de la carte au moment d'être attaché (hôte), et lors de l'initialisation du PCI de BIOS (invité).

Si la machine (virtuelle) a un périphérique PCI attaché, certaines limitations s'appliquent :

1. Seules les cartes PCI aux interruptions non partagées (telles que l'utilisation de MSI sur l'hôte) sont supportées pour le moment.
2. On ne peut pas sauvegarder/restaurer de façon fiable l'état de l'invité (car l'état interne de la carte PCI ne pourrait pas être récupéré).
3. La téléportation (migration en direct) ne fonctionne pas (pour la même raison).
4. Aucune couche d'affectation de mémoire physique. L'hôte préffectera toute la RAM nécessaire au démarrage de la VM (vu que nous ne pouvons pas relier les accès physiques au matériel à la mémoire physique).

9.7 Configuration d'affichage avancée

9.7.1 Résolutions VESA personnalisées

Outre les résolutions VESA standards, le BIOS VESA de VirtualBox vous permet d'ajouter jusqu'à 16 modes graphiques personnalisés qui seront signalés au système d'exploitation invité. Quand on utilise des invités Windows avec les suppléments invité de VirtualBox, un pilote graphique personnalisé sera utilisé à la place de la solution VESA de repli, donc ces informations ne s'appliquent pas.

Vous pouvez configurer des modes graphiques supplémentaires pour chaque VM en utilisant la fonction de données supplémentaires. La clé des données supplémentaires s'appelle `CustomVideoMode<x>` avec x étant un numéro de 1 à 16. Merci de remarquer que les modes seront lus de 1 au numéro suivant non défini ou jusqu'à 16. L'exemple suivant ajoute un mode graphique correspondant à la résolution d'affichage native de nombreux ordinateurs notebook :

```
VBoxManage setextradata "nom VM" "CustomVideoMode1" "1400x1050x16"
```

Les IDs du mode VESA pour les modes graphiques personnalisés commencent à 0x160. Afin d'utiliser le mode graphique personnalisé ci-dessus, vous devez donner à Linux la ligne de commande suivante :

```
vga = 0x200 | 0x160
vga = 864
```

Pour les systèmes d'exploitation ayant les suppléments invité, vous pouvez définir un mode graphique personnalisé en utilisant la fonction d'astuce du mode graphique.

9.7.2 Configuration de la résolution maximum des invités quand on utilise l'interface graphique

Quand on démarre des systèmes invités ayant les suppléments invité installés en utilisant l'interface graphique (l'application normale de VirtualBox), ils ne seront pas autorisés à utiliser des résolutions d'écran supérieures à la taille de l'écran de l'hôte sauf si l'utilisateur les redimensionne à la main en utilisant la fenêtre, en basculant en mode plein écran ou transparent ou en envoyant une astuce de mode graphique utilisant VBoxManage. Ce comportement est celui que la plupart des utilisateurs voudront mais si vous avez d'autres besoins, il est possible de le modifier en exécutant une des commandes suivantes sur la ligne de commandes :

```
VBoxManage setextradata global GUI/MaxGuestResolution any
```

supprimera toutes les limites des résolutions de l'invité.

```
VBoxManage setextradata global GUI/MaxGuestResolution >width,height<
```

spécifie à la main une résolution maximum.

```
VBoxManage setextradata global GUI/MaxGuestResolution auto
```

restaure les paramètres par défaut. Remarquez que ces paramètres s'appliquent globalement à tous les systèmes invités, pas seulement à une seule machine.

9.8 Configuration avancée du stockage

9.8.1 Utiliser un disque dur brut de l'hôte à partir de l'invité

À partir de la version 1.4, plutôt que d'utiliser des images de disques virtuels (comme décrit en détail au chapitre [5, Le stockage virtuel](#), page 85), VirtualBox peut aussi présenter aux machines virtuelles soit des disques durs entiers, soit des partitions sélectionnées, comme des disques virtuels.

Avec VirtualBox, ce type d'accès s'appelle "l'accès au disque dur brut" ; il permet à un système d'exploitation invité d'accéder à son disque dur virtuel sans passer par le système de fichiers de l'OS hôte. La différence de performance finale entre les fichiers images et les disques bruts varie beaucoup selon l'overhead du système de fichiers hôte et le dynamisme de la croissance des images, et enfin des stratégies de mise en cache de l'OS hôte. La mise en cache concerne aussi indirectement d'autres aspects tels que le comportement en cas d'échec, à savoir si le disque dur contient toutes les données écrites avant un OS hôte ne plante. Consultez la documentation de votre OS hôte pour les détails à ce sujet.

Avertissement: L'accès au disque dur brut est réservé aux utilisateurs experts. Une utilisation incorrecte ou d'une configuration obsolète peut provoquer une **perte totale des données** du disque physique. Surtout, *n'essayez pas* de démarrer la partition avec le système d'exploitation hôte actuellement en fonction dans un invité. Cela entraînera une grave corruption de données.

L'accès au disque dur brut – tant entiers qu'aux partitions individuelles – est implémenté comme support du format image VMDK. Il s'en suit que vous devrez créer un fichier image VMDK qui définit l'endroit où les données seront stockées. Après avoir créé une image VMDK spéciale, vous pouvez l'utiliser comme un disque virtuel normal. Par exemple, vous pouvez utiliser le gestionnaire VirtualBox (chapitre [5.3, Le gestionnaire de médias virtuels](#), page 88) ou VBoxManage pour affecter l'image à une machine virtuelle.

9.8.1.1 L'accès à un disque dur physique

Si cette variante est la plus simple à paramétrier, vous devez avoir à l'esprit que cela donnera au système d'exploitation invité un accès total et direct à *tout un disque dur*. Si votre système d'exploitation *hôte* démarre aussi sur ce disque, merci de faire particulièrement attention à ne pas accéder à la partition avec l'invité. Côté positif, le disque physique peut être repartitionné de façon arbitraire sans devoir recréer le fichier image qui donne accès au disque brut.

Pour créer une image qui représente un disque dur physique entier (qui ne contiendra pas de vraies données physiques vu qu'elles seront stockées sur le disque physique), sur un hôte Linux, utilisez la commande

```
VBoxManage internalcommands createrawvmdk -filename /chemin/vers/fichier.vmdk
    -rawdisk /dev/sda
```

Ceci crée l'image /chemin/vers/fichier.vmdk (il doit être absolu), et toutes les données seront lues et écrites à partir de /dev/sda.

Sur un hôte Windows, plutôt que de spécifier le périphérique comme ci-dessus, utilisez par exemple \\.\PhysicalDrive0. Sur un hôte Mac OS X, utilisez plutôt, par exemple, /dev/disk1. Remarquez que sur OS X, vous ne pouvez avoir d'accès à tout un disque que si aucun volume n'est monté à partir de là.

La création de l'image exige un accès en lecture/écriture au périphérique donné. L'accès en lecture/écriture sera aussi nécessaire plus tard lors de l'utilisation de l'image d'une machine virtuelle. Sur certaines plateformes hôtes (comme Windows Vista et supérieur), l'accès au disque brut peut être restreint et non autorisé par l'OS hôte dans certaines situations.

Comme avec les images de disque normales, ceci n'attache pas automatiquement l'image nouvellement créée à une machine virtuelle. Ceci peut se faire avec, par exemple,

```
VBoxManage storageattach WindowsXP --storagectl "IDE Controller"
    --port 0 --device 0 --type hdd --medium /path/to/file.vmdk
```

Quand cela se fait, la machine virtuelle démarra depuis le disque physique spécifié.

9.8.1.2 Accès aux partitions individuelles d'un disque dur physique

Ce “support de partition brut” est très semblable à l'accès au “disque dur complet” décrit ci-dessus. Cependant, dans ce cas, toutes les informations de partitionnement seront stockées dans l'image VMDK, donc vous pouvez par exemple installer un autre chargeur d'amorçage dans le disque dur virtuel sans toucher les informations de partitionnement de l'hôte. Si l'invité pourra voir toutes les partitions existantes sur le disque physique, l'accès sera filtré de sorte que la lecture des partitions pour lesquelles aucun accès n'est autorisé ne contiendra que des zéros et que toutes les écritures dessus soient ignorées.

Pour créer une image spéciale pour le support d'une partition brute (qui contiendra une petite quantité de données, comme déjà indiqué), sur un hôte Linux, utilisez la commande

```
VBoxManage internalcommands createrawvmdk -filename /chemin/vers/fichier.vmdk
    -rawdisk /dev/sda -partitions 1,5
```

Comme vous pouvez le voir, la commande est identique à celle pour l'accès “au disque dur brut”, sauf le paramètre supplémentaire *-partitions*. Cet exemple créerait l'image /chemin/vers/fichier.vmdk (qui, de nouveau, doit être absolu), et les partitions 1 et 5 de /dev/sda deviendraient accessibles à l'invité.

VirtualBox la même numérotation de partitions que votre hôte Linux. Il s'en suit que les numéros donnés dans l'exemple ci-dessus se réfèreraient respectivement à la première partition primaire et au premier lecteur logique de la partition étendue.

Sur un hôte Windows, au lieu de spécifier le périphérique comme ci-dessus, utilisez par exemple \\.\PhysicalDrive0. Sur un hôte Mac OS X, utilisez plutôt par exemple /dev/disk1. Remarquez que sur OS X, vous ne pouvez utiliser que des partitions non montées (éjectez d'abord

les volumes concernés). Les numéros de partition sont les mêmes sur les hôtes Linux, Windows et Mac OS X.

Vous pouvez prendre les numéros dans la liste des partitions dans la sortie de

```
VBoxManage internalcommands listpartitions -rawdisk /dev/sda
```

La sortie liste les types et les tailles des partitions pour donner à l'utilisateur assez d'informations pour identifier les partitions nécessaires à l'invité.

Les images donnant accès aux partitions individuelles sont spécifiques à un paramétrage de disque particulier à un hôte. Vous ne pouvez pas transposer ces images à un autre hôte ; et à chaque fois que le partitionnement de l'hôte change, l'image doit être recréée.

La création d'une image exige l'accès en écriture sur le périphérique donné. L'accès en lecture/écriture sera également nécessaire plus tard pour utiliser l'image à partir d'une machine virtuelle. Si ce n'est pas faisable, il existe une variante spéciale de l'accès à une partition brute (disponible aujourd'hui uniquement sur les hôtes Linux) qui évite de devoir donner à l'utilisateur actuel l'accès à tout le disque. Pour faire une telle image, utilisez

```
VBoxManage internalcommands createrawvmdk -filename /chemin/vers/fichier.vmdk
      -rawdisk /dev/sda -partitions 1,5 -relative
```

Utilisée depuis une machine virtuelle, l'image ne se réfèrera pas à tout le disque mais seulement aux partitions individuelles (dans l'exemple /dev/sda1 et /dev/sda5). Par conséquent, l'accès en lecture/écriture n'est requis que pour les partitions concernées, pas pour tout le disque. Mais lors de la création, un accès en lecture seule à tout le disque est nécessaire pour avoir les informations de partitionnement.

Dans certaines configurations, il peut être nécessaire de modifier le code du MBR de l'image créée, par exemple pour remplacer le chargeur de démarrage Linux utilisé sur l'hôte par un autre chargeur de démarrage. Cela permet, par exemple, à l'invité, de démarrer directement sur Windows, alors que l'hôte démarre sur Linux sur le "même" disque. Pour obtenir cela, le paramètre -mbr vous est offert. Il spécifie un nom de fichier à partir duquel il faut prendre le code du MBR. La table des partitions n'est pas modifiée, donc on peut utiliser un fichier MBR d'un système ayant un partitionnement totalement différent. Un exemple est

```
VBoxManage internalcommands createrawvmdk -filename /chemin/vers/fichier.vmdk
      -rawdisk /dev/sda -partitions 1,5 -mbr winxp.mbr
```

Le MBR modifié sera stocké dans l'image, pas sur le disque hôte.

L'image créée peut être attachée à un contrôleur de stockage dans une configuration de VM, comme d'habitude.

9.8.2 Configuration des vendor product data (VPD) du disque dur

VirtualBox signale les données liées au fabricant du produit de ses disques durs virtuels, consistant dans le numéro de série du disque dur, le numéro de révision du firmware et du modèle. Vous pouvez modifier ces données en utilisant les commandes suivantes :

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/SerialNumber" "serial"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/FirmwareRevision" "firmware"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/ModelNumber" "model"
```

Le numéro de série est une chaîne alphanumérique de 20 octets, la Révision du firmware est une chaîne alphanumérique de 8 octets et le numéro de modèle est une chaîne alphanumérique de 40 octets. Au lieu de "Port0" (qui renvoie au premier port), spécifiez le port SATA désiré du disque dur.

Les commandes ci-dessus s'appliquent aux machines virtuelles ayant un contrôleur AHCI (SATA). Les commandes pour les machines virtuelles ayant un contrôleur IDE sont :

9 Sujets avancés

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/SerialNumber" "serial"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/FirmwareRevision" "firmware"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/piix3ide/0/Config/PrimaryMaster/ModelNumber" "model"
```

Pour les disques durs, il est aussi possible de marquer le lecteur comme ayant un média non rotationnel avec :

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/NonRotational" "1"
```

Trois paramètres supplémentaires sont nécessaires pour que les lecteurs CD/DVD signalent les données produit du fabricant :

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/ATAPIVendorId" "vendor"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/ATAPIProductId" "product"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/ahci/0/Config/Port0/ATAPIRevision" "revision"
```

L'id du fabricant est une chaîne alphanumérique de 8 octets, l'id du produit est une chaîne alphanumérique de 16 octets, la révision est une chaîne alphanumérique de 4 octets. À la place de "Port0" (qui renvoie au premier port), spécifiez le port du disque dur SATA désiré.

9.8.3 Accès à des cibles iSCSI via le réseau interne

En fonctionnalité expérimentale, VirtualBox permet d'accéder à une cible iSCSI d'une machine virtuelle en fonction configurée pour utiliser le mode réseau interne. Merci de voir le chapitre [5.10, Serveurs iSCSI](#), page [97](#) ; chapitre [6.5, Réseau interne](#), page [103](#) ; et chapitre [8.18, VBoxManage storageattach](#), page [142](#) pour avoir des informations supplémentaires.

La pile IP d'accès au réseau interne doit être configurée dans la machine virtuelle qui accède à la cible iSCSI. Vous devez choisir une IP statique libre et une adresse MAC non utilisée par d'autres machines virtuelles. Dans l'exemple ci-dessous, adaptez le nom de la machine virtuelle, l'adresse MAC, la configuration IP et le nom du réseau interne ("MyIntNet") selon vos besoins. Les huit commandes suivantes doivent être d'abord lancées :

```
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/Trusted 1
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/Config/MAC 08:00:27:01:02:0f
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/Config/IP 10.0.9.1
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/Config/Netmask 255.255.255.0
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/LUN#0/Driver IntNet
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/LUN#0/Config/Network MyIntNet
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/LUN#0/Config/TrunkType 2
VBoxManage setextradata "nom VM" VBoxInternal/Devices/IntNetIP/0/LUN#0/Config/IsService 1
```

Enfin, le disque iSCSI doit être attachée avec l'option `--intnet` pour dire à l'initiateur iSCSI d'utiliser le réseau interne :

```
VBoxManage storageattach ... --medium iscsi
    --server 10.0.9.30 --target iqn.2008-12.com.sun:samplertarget --intnet
```

Par rapport à une configuration iSCSI "ordinaire", l'adresse IP de la cible *doit* être spécifiée comme un adaptateur IP numérique, vu qu'il n'y a pas de résolveur DNS pour le réseau interne.

La machine virtuelle ayant la cible iSCSI devrait être démarrée avant que la VM qui l'utilise ne soit allumée. Si vous démarrez une machine virtuelle qui utilise un disque iSCSI sans que la cible iSCSI ne soit allumée, elle peut mettre jusqu'à 200 secondes avant de détecter cette situation. La VM ne pourra pas s'allumer.

9.9 Lancer plus de 128 VMs sur des hôtes Linux

\$

Les hôtes Linux ont un nombre figé d'IDs de sémaphores IPC par processus qui empêche les utilisateurs de lancer énormément de VMs. Le nombre exact peut varier selon la distribution Linux. En essayant de lancer trop de VMs, vous verriez s'afficher une erreur “Cannot create IPC semaphore”. Pour lancer plus de VMs, vous devrez augmenter la limite d'IDs de sémaphore du processus VBoxSVC. Cherchez les limites du sémaphore imposé par le noyau en exécutant en tant que root :

```
#/sbin/sysctl kernel.sem
kernel.sem = 250 32000 32 128
```

Le paramètre “kernel.sem” rassemble 4 valeurs, celle qui nous intéresse s'appelle “SEMMNI”, le nombre maximum d'IDs de sémaphore, qui est de 128 dans l'exemple ci-dessus. Augmentez cette limite d'ID de sémaphore en exécutant en tant que rootnbsp;:

```
echo "kernel.sem = 250 32000 32 2048" >> /etc/sysctl.conf
/sbin/sysctl -p
```

Les commandes ci-dessus ajouteront les nouvelles limites au fichier de configuration, prolongeant l'effet au cours des redémarrages, et elles activeront les nouvelles limites dans le noyau en cours d'exécution.

9.10 Lancer plus de 120 VMs sur les hôtes Solaris

Les hôtes Linux ont un nombre figé d'IDs de sémaphores IPC par processus qui empêche les utilisateurs de lancer énormément de VMs. En essayant de lancer trop de VMs, vous verriez s'afficher une erreur “Cannot create IPC semaphore”. Pour lancer plus de VMs, vous devrez augmenter la limite d'IDs de sémaphore du processus VBoxSVC.

9.10.1 Solution temporaire quand VirtualBox est en fonction

Exécutez, en tant qu'administrateur, la commande prctl comme indiqué ci-dessous pour le processus VBoxSVC actuellement en fonction. Vous pouvez savoir l'ID du processus en utilisant la commande ps.

```
prctl -r -n project.max-sem-ids -v 2048 <pid-of-VBoxSVC>
```

Cela augmentera immédiatement la limite sémaphore du processus VBoxSVC actuellement en fonction et vous permettra de lancer davantage de VMs. Cependant, cette modification ne reste pas au redémarrage de VBoxSVC.

9.10.2 Solution permanente, exige que l'utilisateur se re-connecte

Si l'utilisateur qui lance VirtualBox est l'administrateur, exécutez la commande suivante :

```
prctl -n project.max-sem-ids -v 2048 -r -i project user.root
```

À partir de là, le démarrage de nouveaux processus tiendra compte de la limite de 2048. Vous pouvez alors vous reconnecter ou fermer toutes les VMs et redémarrer VBoxSVC. Vous pouvez vérifier la limite actuelle d'ID de sémaphore pour VBoxSVC en utilisant la commande suivante :

```
prctl -n project.max-sem-ids -i process <pid-of-VBoxSVC>
```

Si l'utilisateur qui exécute VirtualBox n'est pas administrateur, vous devez ajouter la propriété au projet par défaut de l'utilisateur. Créez le projet par défaut et réglez la limite en exécutant en tant qu'administrateur :

```
projadd -U <nomutilisateur> user.<nomutilisateur>
projmod -s -K "project.max-sem-ids=(priv,2048,deny)" user.<nomutilisateur>
```

Remplacez "<nomutilisateur>" avec le nom d'utilisateur exécutant VirtualBox. Puis reconnectez-vous sous le nom de cet utilisateur qui pourra exécuter plus de 120 VMs.

9.11 Commandes de base pour utiliser les ports série

À partir de la version 1.4, VirtualBox fournissait le support les ports série virtuels qui, pour l'instant, était plutôt compliqué à paramétrier avec la séquence des commandes `VBoxManage setextradata`. Depuis la version 1.5, cette façon de paramétrier les ports série n'est plus nécessaire et *obsolète*. Pour paramétrier les ports série virtuels, utilisez les méthodes décrites maintenant au chapitre [3.9, Ports série](#), page [59](#).

Note: Pour être rétro-compatible, les anciennes commandes `setextradata`, dont la description ci-dessous est issue de l'ancienne version du manuel, *restent valables* côté de la nouvelle façon de configurer les ports série. Il s'en suit que si la première méthode de configuration des ports série ne marche pas, assurez-vous que la VM en question ne contient pas d'anciennes données de configuration actives telles que érites ci-dessous.

L'ancienne séquence de configuration d'un port série utilisait les 6 commandes suivantes :

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/serial/0/Config/IRQ" 4
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/serial/0/Config/I0Base" 0x3f8
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/serial/0/LUN#0/Driver" Char
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/serial/0/LUN#0/AttachedDriver/Driver" NamedPipe
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/serial/0/LUN#0/AttachedDriver/Config/Location" "\\.\pipe\vboxCOM1"
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/serial/0/LUN#0/AttachedDriver/Config/IsServer" 1
```

Cela définit un port série dans l'invité avec les paramètres par défaut de COM1 (IRQ 4, adresse E/S 0x3f8) et le paramètre `Location` suppose que cette configuration est utilisée sur un hôte Windows, car on utilise la syntaxe de tuyau (pipe) nommé Windows. Gardez à l'esprit que sur les hôtes Windows, un tuyau nommé doit toujours commencer par `\.\pipe\`. Sur Linux, s'appliquent les mêmes paramètres de configuration, sauf que vous pouvez choisir le nom du chemin de `Location` plus librement. Les sockets du domaine local se mettent n'importe où, pourvu que l'utilisateur qui exécute VirtualBox ait le droit de créer un nouveau fichier dans le répertoire. La dernière commande ci-dessus définit que VirtualBox agit comme un serveur, c'est-à-dire qu'il crée lui-même le tuyau nommé au lieu de se connecter à un autre qui existe déjà.

9.12 Peaufiner le moteur NAT de VirtualBox

9.12.1 Configurer l'adresse d'une interface réseau NAT

En mode NAT, on affecte à l'interface réseau de l'invité une plage IPv4 `10.0.x.0/24` par défaut, où `x` correspond à l'instance d'une interface NAT +2. Donc, `x` vaut 2 quand il n'y a qu'une

instance NAT d'active. Dans ce cas, l'invité se voit affecter l'adresse 10.0.2.15, la passerelle est définie sur 10.0.2.2 et on peut trouver le serveur de noms sur 10.0.2.3.

Si, pour une raison quelconque, vous devez modifier le réseau NAT, ce qui se fait avec la commande suivante :

```
VBoxManage modifyvm "nom VM" --natnet1 "192.168/16"
```

Cette commande réservera les adresses réseaux de 192.168.0.0 à 192.168.254.254 à la première instance réseau NAT de "nom VM". On affecterait à l'invité l'IP 192.168.0.15 et on pourrait trouver la passerelle par défaut sur 192.168.0.2.

9.12.2 Configurer le serveur d'amorçage (prochain serveur) d'une interface réseau NAT

Pour un amorçage réseau en mode NAT, VirtualBox utilise par défaut le serveur TFTP inclu, qui se trouve à l'adresse 10.0.2.3. Ce comportement par défaut devrait très bien fonctionner pour les scénarii de démarrage à distance courants. Cependant, il est possible de modifier l'IP du serveur d'amorçage et l'emplacement de l'image de démarrage avec les commandes suivantes :

```
VBoxManage modifyvm "nom VM" --nattftpserver1 10.0.2.2
VBoxManage modifyvm "nom VM" --nattftpfile1 /srv/tftp/boot/MyPXEBoot.pxe
```

9.12.3 Peaufiner les tampons TCP/IP pour NAT

La performance de la pile NAT de VirtualBox est souvent déterminée par son interaction avec la pile TCP/IP de l'hôte et la taille de plusieurs tampons (SO_RCVBUF et SO_SNDBUF). Pour certaines configurations, les utilisateurs pourraient vouloir ajuster la taille des tampons pour une meilleure performance. Vous pouvez faire cela en utilisant les commandes suivantes (les valeurs s'expriment en kilo-octets peuvent varier de 8 à 1024) :

```
VBoxManage modifyvm "nom VM" --natsettings1 16000,128,128,0,0
```

Cet exemple illustre le peaufinage des paramètres NAT. Le premier paramètre est le MTU, puis la taille du tampon d'envoi de la socket et la taille du tampon de réception de la socket, la taille initiale de la fenêtre d'envoi TCP, et enfin, la taille initiale de la fenêtre de réception TCP. Remarquez que la spécification de zéro revient à se rabattre sur la valeur par défaut.

Chacun de ces tampons a une taille par défaut de 64Ko et un MTU par défaut de 1500.

9.12.4 Associer des sockets à une interface spécifique

Par défaut, le moteur NAT de VirtualBox dirigera les paquets TCP/IP via l'interface par défaut affectée par la pile TCP/IP de l'hôte. (La raison technique en est que le moteur NAT utilise des sockets pour la communication.) Si, pour une raison quelconque, vous voulez changer ce comportement, vous pouvez dire au moteur NAT d'associer à une interface en particulier une adresse IP. Utilisez la commande suivante :

```
VBoxManage modifyvm "nom VM" --natbindip1 "10.45.0.2"
```

Après cela, le trafic sortant sera envoyé par interface ayant l'adresse IP 10.45.0.2. Merci de vous assurer que cette interface est active et en fonction avant cette affectation.

9.12.5 Activer le proxy DNS en mode NAT

Le moteur NAT offre par défaut les mêmes serveurs DNS à l'invité que ceux configurés sur l'hôte. Dans certains scénarios, il peut être souhaitable de cacher les IPs du serveur DNS à l'invité, par exemple quand ces informations peuvent changer sur l'hôte après l'expiration des baux DHCP. Dans ce cas, vous pouvez dire au moteur NAT d'agir comme un proxy DNS en utilisant la commande suivante :

```
VBoxManage modifyvm "nom VM" --natdnspool1 on
```

9.12.6 Utiliser le résolveur de l'hôte comme proxy DNS en mode NAT

Pour résoudre les noms de réseau, le serveur DHCP du moteur NAT offre une liste de serveurs DNS enregistrés de l'hôte. Si pour une raison quelconque, vous devez cacher cette liste de serveurs DNS et utiliser les paramètres du serveur DNS de l'hôte, forçant ainsi le moteur NAT de VirtualBox à intercepter les requêtes DNS et à les rediriger sur le résolveur de l'hôte, utilisez la commande suivante :

```
VBoxManage modifyvm "nom VM" --natdnshostresolver1 on
```

Remarquez que ce paramètre est identique au mode proxy DNS, cependant alors que le mode proxy ne redirige que les requêtes DNS sur les serveurs appropriés, le mode résolveur interprétera les requêtes DNS et utilisera l'API DNS de l'hôte pour prendre les informations et les retourner à l'invité.

9.12.6.1 Résolution de noms d'hôte définie par l'utilisateur

Dans certains cas, il pourrait être utile d'intercepter le mécanisme de résolution de noms, en fournissant une adresse IP définie par l'utilisateur pour une requête DNS en particulier. Le mécanisme d'interception permet à l'utilisateur d'associer non seulement un hôte, mais aussi des domaines et même des conventions de nommage plus complexes si nécessaire.

La commande suivante définit la règle d'association d'un nom et d'une IP spécifiée :

```
VBoxManage setextradata "nom VM" \
    "VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/Config/HostResolverMappings/ \
    <nom uniq de la règle d'interception>/HostIP" <IPv4>
VBoxManage setextradata "nom VM" \
    "VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/Config/HostResolverMappings/ \
    <nom uniq de la règle d'interception>/HostName" <nom de vhôte>
```

La commande suivante définit une règle pour associer un échantillon de nom à une IP spécifiée :

```
VBoxManage setextradata "nom VM" \
    "VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/Config/HostResolverMappings/ \
    <nom uniq de la règle d'interception>/HostIP" <IPv4>
VBoxManage setextradata "nom VM" \
    "VBoxInternal/Devices/{pcnet,e1000}/0/LUN#0/Config/HostResolverMappings/ \
    <uniq name of interception rule>/HostNamePattern" <échantillonhôte>
```

L'échantillon hôte peut inclure "|", "?" et "*".

Cette exemple démontre la façon de demander au mécanisme du résolveur de l'hôte de résoudre tout le domaine et probablement des miroirs du site www.blocked-site.info avec l'IP 127.0.0.1:

```
VBoxManage setextradata "nom VM" \
    "VBoxInternal/Devices/e1000/0/LUN#0/Config/HostResolverMappings/ \
    all_blocked_site/HostIP" 127.0.0.1
VBoxManage setextradata "nom VM" \
    "VBoxInternal/Devices/e1000/0/LUN#0/Config/HostResolverMappings/ \
    all_blocked_site/HostNamePattern" "*.blocked-site.*|*.fb.org"
```

Note: Le mécanisme de résolution de l'hôte devrait être activé pour utiliser les règles d'association définies par l'utilisateur (merci de voir chapitre [9.12.6, Utiliser le résolveur de l'hôte comme proxy DNS en mode NAT](#), page 180 pour plus de détails).

9.12.7 Configurer des aliases pour le moteur NAT

Par défaut, le cœur de NAT utilise des alias et des ports aléatoires quand il génère un alias pour la connexion. Cela fonctionne bien pour la plupart des protocoles comme SSH, FTP et ainsi de suite. Mais certains protocoles pourraient nécessiter un comportement plus transparent ou dépendre du vrai numéro de port pour envoyer un paquet. Il est possible de modifier le mode NAT avec l'interface VBoxManage avec les commandes suivantes :

```
VBoxManage modifyvm "nom VM" --nataliasmodel proxyonly
```

and

```
VBoxManage modifyvm "Linux Guest" --nataliasmodel sameports
```

Le premier exemple désactive les alias et passe NAT en mode transparent, le deuxième exemple renforce la préservation des numéros des ports. Ces modes peuvent se combiner si nécessaire.

9.13 Configurer les informations DMI du BIOS

Vous pouvez changer les données DMI que VirtualBox fournit aux invités pour une VM spécifique. Utilisez les commandes suivantes pour configurer les informations DMI du BIOS :

9.13.1 Informations DMI du BIOS (type 0)

```
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVendor"      "fabricant BIOS"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSVersion"      "Version BIOS"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseDate"  "date publication BIOS"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMajor" 1
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSReleaseMinor" 2
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMajor" 3
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiBIOSFirmwareMinor" 4
```

9.13.2 Informations système DMI (type 1)

```
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemVendor"      "Fabricant Système"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemProduct"      "Produit système"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemVersion"      "Version système"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemSerial"      "Numéro de série système"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemSKU"        "System SKU"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemFamily"      "Famille système"
VBoxManage setextradata "nom VM"
  "VBoxInternal/Devices/pcbios/0/Config/DmiSystemUuid"        "9852bf98-b83c-49db-a8de-182c42c7226b"
```

9.13.3 Informations carte mère DMI (type 2)

VBoxManage setextradata "nom VM"	"Fabricant carte"
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardVendor"	
VBoxManage setextradata "nom VM"	"Produit carte"
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardProduct"	
VBoxManage setextradata "nom VM"	"Version carte mère"
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardVersion"	
VBoxManage setextradata "nom VM"	"Série carte"
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardSerial"	
VBoxManage setextradata "nom VM"	"Tag Board"
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardAssetTag"	
VBoxManage setextradata "nom VM"	"Emplacement carte"
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardLocInChass"	
VBoxManage setextradata "nom VM"	10
"VBoxInternal/Devices/pcbios/0/Config/DmiBoardType"	

9.13.4 Boîtier système DMI ou chassis (type 3)

VBoxManage setextradata "nom VM"	"Fabricant Chassis"
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisVendor"	
VBoxManage setextradata "nom VM"	"Version Chassis"
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisVersion"	
VBoxManage setextradata "nom VM"	"Série Chassis"
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisSerial"	
VBoxManage setextradata "nom VM"	"Tag Chassis"
"VBoxInternal/Devices/pcbios/0/Config/DmiChassisAssetTag"	

9.13.5 Informatiions DMI du processeur (type 4)

VBoxManage setextradata "nom VM"	"GenuineIntel"
"VBoxInternal/Devices/pcbios/0/Config/DmiProcManufacturer"	
VBoxManage setextradata "nom VM"	"Pentium(R) III"
"VBoxInternal/Devices/pcbios/0/Config/DmiProcVersion"	

9.13.6 Chaînes OEM DMI (type 11)

VBoxManage setextradata "nom VM"	"vboxVer_1.2.3"
"VBoxInternal/Devices/pcbios/0/Config/Dmi0EMVBoxVer"	
VBoxManage setextradata "nom VM"	"vboxRev_12345"
"VBoxInternal/Devices/pcbios/0/Config/Dmi0EMVBoxRev"	

Si une chaîne DMI n'est pas définie, la valeur par défaut de VirtualBox est utilisée. Pour définir une chaîne vide, utilisez "<EMPTY>".

Remarquez que dans la liste ci-dessus, tous les paramètres cités (DmiBIOSVendor, DmiBIOSVersion mais pas DmiBIOSReleaseMajor) sont censés être des chaînes. Si la chaîne est un nombre valide, le paramètre est traité comme un nombre et la VM refusera probablement de démarrer avec une erreur VERR_CFGM_NOT_STRING. Dans ce cas, utilisez "string:<valeur>", par exemple,

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/pcbios/0/Config/DmiSystemSerial"      "string:1234"
```

La modification de ces information cans peut \avérer nécessaire pour donner les informations DMI de l'hôte à l'invité afin d'empêcher Windows de demander une nouvelle clé du produit. Sur les hôtes Linux, vous pouvez obtenir les informations de BIOS DMI avec with

```
dmidecode -t0
```

et les informations du système DMI avec

```
dmidecode -t1
```

9.14 Configurer la table ACPI personnalisée

VirtualBox peut être configuré pour présenter à l'invité une table ACPI personnalisée. Utilisez la commande suivante pour la configurer :

```
VBoxManage setextradata "nom VM"
    "VBoxInternal/Devices/acpi/0/Config/CustomTable" "/chemin/vers/table.bin"
```

La configuration d'une table ACPI personnalisée peut empêcher Windows Vista et Windows 7 de demander une nouvelle clé du produit. Sur les hôtes Linux, on peut lire une des tables de l'hôte dans /sys/firmware/acpi/tables/.

9.15 Peaufiner les horloges et la synchronisation du temps

9.15.1 Configurer le time stamp counter (TSC) (horodateur) de l'invité pour refléter l'heure de l'exécution

Par défaut, VirtualBox synchronise toutes les sources de l'heure dans une source d'heure unique, l'heure de l'hôte monotonic. Cela reflète les suppositions de nombreux systèmes d'exploitation invités qui s'attendent à ce que toutes les sources d'heure reflètent l'heure "la pendule". Dans des circonstances spéciales, il peut être cependant utile de faire en sorte que le TSC (time stamp counter) de l'invité reflète le temps effectif passé à exécuter l'invité.

Ce mode de gestion spécial du TSC peut s'activer individuellement par VM et, pour de meilleurs résultats, il ne faut l'utiliser qu'en association avec la virtualisation matérielle. Pour activer ce mode, utilisez la commande suivante :

```
VBoxManage setextradata "nom VM" "VBoxInternal/TM/TSCTiedToExecution" 1
```

Pour inverser le mode de gestion TSC par défaut, utilisez :

```
VBoxManage setextradata "nom VM" "VBoxInternal/TM/TSCTiedToExecution" 0
```

Remarquez que si vous utilisez le mode de gestion TSC spécial avec un système d'exploitation invité qui est très strict quant à la cohérence des sources de l'heure, il se peut que vous receviez un message d'avertissement ou d'erreur lié à l'incohérence de l'heure. Cela peut aussi rendre l'heure non fiable avec certains systèmes d'exploitation invités en fonction de leur utilisation du TSC.

9.15.2 Accélérer ou ralentir l'horloge de l'invité

Pour certains objectifs, il peut être utile d'accélérer ou de ralentir l'horloge virtuelle de l'invité. Vous pouvez le faire comme suit :

```
VBoxManage setextradata "nom VM" "VBoxInternal/TM/WarpDrivePercentage" 200
```

L'exemple ci-dessus doublera la vitesse de l'horloge de l'invité alors que

```
VBoxManage setextradata "nom VM" "VBoxInternal/TM/WarpDrivePercentage" 50
```

ralentira l'horloge de l'invité. Remarquez que la modification du rythme de l'horloge virtuelle peut perturber l'invité et même provoquer un comportement anormal de l'invité. Par exemple, une vitesse plus élevée signifie des timeouts plus courts pour les périphériques virtuels, provoquant un délai de réponse légèrement accru du périphérique virtuel, à l'origine d'une augmentation de la charge de l'hôte qui peut provoquer des échecs de l'invité. Notez aussi que tous les mécanismes de synchronisation du temps essaieront souvent de resynchroniser l'heure de l'invité sur l'heure de référence (qui est celle de l'hôte si les suppléments invité de VirtualBox sont actifs). Donc, toutes les synchronisations du temps devraient être désactivées si vous modifiez la vitesse de l'horloge invité comme indiqué ci-dessus (voir chapitre [9.15.3, Peaufiner les paramètres de synchronisation du temps des suppléments invité](#), page [184](#)).

9.15.3 Peaufiner les paramètres de synchronisation du temps des suppléments invité

Les suppléments invité de VirtualBox garantissent que l'heure du système invité se synchronise avec l'heure de l'hôte. Plusieurs paramètres peuvent être personnalisés. Vous pouvez définir les paramètres pour une VM spécifique en utilisant la commande suivante :

```
VBoxManage guestproperty set "nom VM" "/VirtualBox/GuestAdd/VBoxService/PARAMETER" VALUE
```

où **PARAMETER** est un des suivants :

- timesync-interval** Spécifie l'intervalle entre deux synchronisations de l'heure invité par rapport à l'hôte. Par défaut, il est de 10000 ms (10 secondes).
- timesync-min-adjust** Valeur absolue minimum du débit mesuré en millisecondes pour faire les ajustements. Par défaut, c'est 1000 ms sur OS/2 et 100 ms ailleurs.
- timesync-latency-factor** Le multiplicateur de latence de demande de temps pour calculer le temps minimum ajusté dynamiquement. Il est par défaut de 8 fois, ce qui veut dire en détails : mesurer le temps mis pour déterminer l'heure de l'hôte (l'invité doit contacter le service hôte de la VM, ce qui peut prendre du temps), multiplier cette valeur par 8 et n'ajuster que si la différence d'heure entre l'hôte et l'invité dépasse cette valeur. Sinon, ne pas ajuster l'heure.
- timesync-max-latency** La latence de demande de l'horloge max acceptée. Par défaut, il s'agit de 250 ms.
- timesync-set-threshold** Début du débit absolu donné en millisecondes, où doit commencer le réglage de l'heure, plutôt que d'essayer de l'ajuster tout simplement. Il s'agit par défaut de 20 minutes.
- timesync-set-start** Définit l'heure à laquelle démarrer le service de syncchro du temps.
- timesync-set-on-restore 0|1** Règle l'heure après que la VM a été restaurée d'un état sauvegardé si vous mettez 1 en paramètre (par défaut). Désactivez-le en mettant 0. Dans ce dernier cas, l'heure sera ajustée tout simplement, ce qui peut mettre du temps.

Vous pouvez aussi spécifier tous ces paramètres comme options de la ligne de commandes du service BoxService.

9.15.4 Désactiver la synchronisation des suppléments invité

Une fois installés et démarrés, les suppléments invité de VirtualBox essaieront de synchroniser l'heure de l'invité avec celle de l'hôte. Vous pouvez l'empêcher en interdisant le service de l'invité de lire l'horloge de l'hôte :

```
VBoxManage setextradata "nom VM" "VBoxInternal/Devices/VMMDev/0/Config/GetHostTimeDisabled" 1
```

9.16 Installer le pilote du réseau bridgé alternatif sur les invités Solaris 11

À partir de VirtualBox 4.1, VirtualBox inclut un nouveau pilote de filtre réseau qui utilise la fonction Crossbow de Solaris 11. Par défaut, ce nouveau pilote est installé pour les hôtes Solaris 11 (construction 159 ci-dessus) qui le supportent.

Pour obliger l'installation de l'ancien pilote de filtre réseau basé sur STREAMS, exécutez en tant qu'administrateur la commande suivante avant d'installer le paquet VirtualBox :

```
touch /etc/vboxinst_vboxflt
```

Pour obliger l'installation du pilote de filtre réseau basé sur Crossbow, exécutez en tant qu'administrateur la commande suivante avant d'installer le paquet VirtualBox :

```
touch /etc/vboxinst_vboxbow
```

Pour vérifier le pilote actuellement utilisé par VirtualBox, exécutez :

```
modinfo | grep vbox
```

Si la sortie contient "vboxbow", cela indique que VirtualBox utilise le pilote de filtre réseau Crossbow, alors que le nom "vboxflt" indique que l'ancien pilote de filtre réseau STREAMS est utilisé.

9.17 Échantillons de VNIC VirtualBox pour les VLANs sur les hôtes Solaris 11

VirtualBox supporte des échantillons VNIC (Virtual Network Interface) pour configurer des VMs via des VLANs.³ Un échantillon VNIC de VirtualBox est un VNIC dont le nom commence par "vboxvnic_template".

Voici un exemple de la façon d'utiliser un échantillon VNIC pour configurer un VLAN pour des VMs. Créez un échantillon VNIC de VirtualBox en exécutant, en tant qu'administrateur :

```
dladm create-vnic -t -l nge0 -v 23 vboxvnic_template0
```

Cela créera un VNIC temporaire par l'interface "nge0" avec l'ID de VLAN 23. Pour créer des échantillons VNIC résistant aux redémarrages de l'hôte, sautez le paramètre **-t** dans la commande ci-dessus. Vous pouvez vérifier l'état actuel des liens en utilisant :

```
$ dladm show-link
LINK CLASS MTU STATE BRIDGE OVER
nge0 phys 1500 up --
nge1 phys 1500 down --
vboxvnic_template0 vnic 1500 up -- nge0

$ dladm show-vnic
LINK OVER SPEED MACADDRESS MACADDRTYPE VID
vboxvnic_template0 nge0 1000 2:8:20:25:12:75 random 23
```

Une fois que l'échantillon VNIC est créé, toutes les VMs ayant besoin de faire partie du VLAN 23 par l'interface physique "nge0" pourront utiliser le même échantillon VNIC. Cela simplifie et rend plus efficace la gestion des VMs sur des VLANs car les détails du VLAN ne sont pas stockés dans la configuration de chaque VM mais récupérés dans le modèle VNIC que vous pouvez modifier n'importe quand en utilisant **dladm**. Outre l'ID du VLAN, des traductions VNIC peuvent être créées avec des propriétés supplémentaires telles que les limites de bande passante, le fanout du processeur, etc. Reportez-vous à la documentation du réseau de votre Solaris pour savoir comment faire cela. Ces propriétés supplémentaires, s'il y en a, s'appliquent aussi aux VMs qui utilisent l'échantillon VNIC.

³Le support du réseau bridgé basé sur Crossbow a été introduit avec VirtualBox 4.1 et il exige Solaris 11 construction 159 ou supérieur.

9.18 Configurer plusieurs interfaces réseaux host-only sur les hôtes Solaris

Par défaut, VirtualBox vous offre une interface réseau host-only. L'ajout de davantage d'interfaces réseaux host-only sur les hôtes Solaris exige une configuration manuelle. Voici comment ajouter deux interfaces réseaux host-only supplémentaires.

Vous devez d'abord arrêter toutes les VMs en fonction et désactiver toutes les interfaces "vboxnet". Exécutez les commandes suivantes en tant qu'administrateur :

```
ifconfig vboxnet0 unplumb
```

Après vous être assuré que toutes les interfaces vboxnet sont désactivées, supprimez le pilote en utilisant :

```
rem_drv vboxnet
```

puis éditez le fichier `/platform/i86pc/kernel/drv/vboxnet.conf` et ajoutez une ligne pour les nouvelles interfaces :

```
name="vboxnet" parent="pseudo" instance=1;
name="vboxnet" parent="pseudo" instance=2;
```

Ajoutez autant de lignes comme celles-ci que nécessaire et assurez-vous que le nombre d'"instance" soit implémenté de façon unique. Ensuite, rechargez le pilote vboxnet en utilisant :

```
add_drv vboxnet
```

Maintenant, activez toutes les interfaces en utilisant `ifconfig vboxnetX plumb` (où X peut être 0, 1 ou 2 dans ce cas) et une fois activée, vous pouvez alors configurer l'interface comme n'importe quelle interface réseau.

Pour que les paramètres de vos nouvelles interfaces réseaux persistent entre les redémarrages, vous devrez éditer les fichiers `/etc/netmasks`, utilisez NWAM `/etc/nwam/llp` et ajoutez les entrées adéquates pour définir le masque réseau et l'IP statique de chacune de ces interfaces. L'installateur de VirtualBox ne met à jour ces fichiers de configuration que pour l'interface "vboxnet0" qu'il crée par défaut.

9.19 Configurer le CoreDumper sur les hôtes Solaris

VirtualBox est capable de produire ses propres fichiers cœur pour un débogage étendu si quelque chose ne va pas. Cela n'est actuellement disponible que sur les hôtes Solaris.

On peut activer le CoreDumper en utilisant la commande suivante :

```
VBoxManage setextradata "nom VM" VBoxInternal2/CoreDumpEnabled 1
```

Vous pouvez spécifier le répertoire à utiliser pour y mettre les fichiers cœur avec cette commande :

```
VBoxManage setextradata "nom VM" VBoxInternal2/CoreDumpDir <chemin-du-répertoire>
```

Assurez-vous que le répertoire que vous spécifiez se trouve sur un volume ayant un espace disque suffisant et où le processus VirtualBox a assez de droits pour écrire des fichiers dans ce répertoire. Si vous sautez cette commande et si vous ne spécifiez aucun répertoire où mettre les fichiers cœur, le répertoire actuel de l'exécutable de VirtualBox sera utilisé (ce qui échouerait vraisemblablement au moment de l'écriture des coeurs car ils sont protégés par des droits administrateur). Il est recommandé que vous définissiez explicitement un répertoire d'envoi des fichiers cœur.

Vous devez spécifier le moment où les CoreDumper de VirtualBox devraient être récupérés. Cela se fait en utilisant les commandes suivantes :

```
VBoxManage setextradata "nom VM" VBoxInternal2/CoreDumpReplaceSystemDump 1
VBoxManage setextradata "nom VM" VBoxInternal2/CoreDumpLive 1
```

Vous devrez passer au moins une des deux commandes ci-dessus si vous avez activé les Core-Dumper.

Le réglage de CoreDumpReplaceSystemDump prévoit que la VM outrepasse le mécanisme cœur de l'hôte et en cas de de plantage, seul le de VirtualBox produirait le fichier cœur.

Le réglage de CoreDumpLive demande à la VM de produire des cœurs à chaque fois que le processus de la VM reçoit un signal SIGUSR2. Après avoir produit le fichier cœur, la VM ne sera pas interrompu et continuera de fonctionner. Vous pouvez ainsi récupérer des cœurs du processus de la VM en utilisant :

```
kill -s SIGUSR2 <VM-process-id>
```

Les fichiers cœur produits par le CoreDumper de VirtualBox ont la forme core.vb.<ProcessName>.<ProcessID>, par exemple core.vb.VBoxHeadless.11321.

9.20 Déverrouiller l'interface graphique du gestionnaire de VirtualBox

9.20.1 Personnalisation de l'interface graphique

Il existe plusieurs paramètres de personnalisation avancés pour déverrouiller le gestionnaire de VirtualBox, c'est-à-dire pour supprimer des fonctionnalités que l'utilisateur ne devrait pas voir.

```
VBoxManage setextradata global GUI/Customizations OPTION[,OPTION...]
```

où OPTION est un des mots-clés suivants :

noSelector N'autorise pas le démarrage du gestionnaire de VirtualBox. Ceci affichera une fenêtre contenant un vrai message d'erreur.

noMenuBar Les fenêtres de la VM ne contiendront pas de barre de menus.

noStatusBar Les fenêtres de la VM ne contiendront pas de barre d'état.

Pour désactiver toutes les personnalisations de l'interface graphique, faites

```
VBoxManage setextradata global GUI/Customizations
```

9.20.2 Personnalisation de la touche hôte

Pour désactiver toutes les combinaisons de touches de l'hôte, ouvrez les préférences et modifiez la touche hôte sur *Aucune*. Cela pourrait être utile lors de l'utilisation de VirtualBox en mode kiosk.

Pour redéfinir ou désactiver certaines actions de la touche hôte, utilisez la commande suivante :

```
VBoxManage setextradata global GUI/Input/MachineShortcuts "FullscreenMode=F,...."
```

La liste suivante montre les actions possibles avec la touche hôte ainsi que leur raccourci par défaut avec la touche hôte. Le paramétrage d'une action sur *Aucune* désactivera cette action de la touche hôte.

Action	Touche hôte par défaut	Action
SettingsDialog	S	ouvre la boîte de dialogue des préférences de la VM
TakeSnapshot	S	prend un instantané
InformationsDialog	N	affiche la boîte de dialogue d'informations de la VM
MouseIntegration	I	bascule l'intégration de la souris
TypeCAD	Del	envoie Ctrl+Alt+Del
TypeCABS	Backspace	envoie Ctrl+Alt+Backspace
Pause	P	Met en pause la VM
Reset	R	réinitialise (en dur) l'invité
Shutdown	H	appui sur le bouton d'alimentation ACPI
Close	Q	affiche la boîte de dialogue Fermer la VM
FullscreenMode	F	passe la VM en plein écran
SeamlessMode	L	passe la VM en mode transparent
ScaleMode	C	passe la VM en mode échelle
PopupMenu	Home	affiche un menu popup en mode plein-écran / transparent

Pour désactiver le mode plein-écran ainsi que le mode transparent, utilisez la commande suivante :

```
VBoxManage setextradata global GUI/Input/MachineShortcuts "FullscreenMode=None,SeamlessMode=None"
```

9.20.3 Action puand la VM s'arrête

Vous pouvez interdire certaines actions quand la VM s'arrête. Pour interdire des actions spécifiques, tapez :

```
VBoxManage setextradata "nom VM" GUI/RestrictedCloseActions OPTION[,OPTION...]
```

où OPTION est l'un des mots-clés suivants :

SaveState N'autorise pas l'utilisateur à sauvegarder l'état de la VM quand elle s'arrête.

Shutdown N'autorise pas l'utilisateur à éteindre la VM en envoyant l'événement ACPI couper à l'invité.

PowerOff N'autorise pas l'utilisateur à couper la VM.

Restore N'autorise pas l'utilisateur à revenir au dernier instantané lors de l'extinction de la VM.

Toutes les combinaison de ce qui précède est possible. Si vous spécifiez toutes les options, la VM ne pourra pas être éteinte.

9.21 Démarrer le service Web de VirtualBox automatiquement

Le service Web de VirtualBox (vboxwebsrv) est utilisé pour contrôler VirtualBox à distance. Il est documenté en détails dans le Software Development Kit (SDK) de VirtualBox ; merci de voir chapitre 11, [Interfaces de programmation de VirtualBox](#), page 207. Comme la base client qui utilise cette interface grossit, nous avons ajouté des scripts de démarrage pour les systèmes d'exploitation que nous supportons. Les sections suivantes décrivent la manière de les utiliser. Le service Web de VirtualBox ne démarre jamais automatiquement suite à une installation standard.

9.21.1 Linux : démarrer le service web via init

Sur Linux, le service web peut être démarré automatiquement au démarrage de l'hôte en ajoutant les paramètres adéquats au fichier `/etc/default/virtualbox`. Un paramètre est obligatoire, `VBOXWEB_USER`, qui doit être défini sur l'utilisateur qui démarrera alors les VMs. Les paramètres du tableau ci-dessous commencent tous par `VBOXWEB_` (`VBOXWEB_HOST`, `VBOXWEB_PORT` etc.) :

Paramètre	Description	Par défaut
USER	L'utilisateur sous lequel fonctionne le service web	
HOST	L'hôte où on doit chercher le service web	localhost
PORT	Le port où on doit chercher le service web	18083
SSL_KEYFILE	Fichier de clé et du certificat du serveur, format PEM	
SSL_PASSWORDFILE	Nom du fichier mot de passe de la clé du serveur	
SSL_CACERT	Fichier de certificat CA, format PEM	
SSL_CAPATH	Chemin du certificat CA	
SSL_DHFILE	Nom du fichier DH ou longueur de la clé DH en octets	
SSL RANDFILE	Fichier contenant seed en générateur de nombre aléatoire	
TIMEOUT	Temps de la session en secondes ; 0 désactive le timeout	300
CHECK_INTERVAL	Fréquence des vérifications des timeout en secondes	5
THREADS	Nombre maximum de session simultanées possibles	100
KEEPALIVE	Nombre maximum de requêtes avant de fermer une socket	100
LOGFILE	Nom du fichier où écrire les journaux	
ROTATE	Nombre de fichiers journaux ; 0 désactive la journalisation	10
LOGSIZE	Taille maximum d'un fichier journal en octets à récupérer	1Mo
LOGINTERVAL	Délai maximum en secondes pour ratrapper l'enregistrement des journaux	1 day

La définition du paramètre `SSL_KEYFILE` active le support SSL/TLS. L'utilisation de chiffrement est fortement recommandée, car sans cela, tout (même les mots de passe) sera transféré en clair.

9.21.2 Solaris: démarrer le service web par SMF

Sur les hôtes Solaris, le démon du service Web de VirtualBox est intégré à l'environnement SMF. Vous pouvez modifier les paramètres mais vous n'êtes pas obligé si ceux par défaut ci-dessous correspondent déjà à vos besoins :

```
svccfg -s svc:/application/virtualbox/webservice:default setprop config/host=localhost
svccfg -s svc:/application/virtualbox/webservice:default setprop config/port=18083
svccfg -s svc:/application/virtualbox/webservice:default setprop config/user=root
```

Le tableau de la section précédente montrant le nom des paramètres et leurs réglages par défaut s'applique également à Solaris. Vous devez passer le nom des paramètres en minuscules et ajouter le préfixe `config/`, par exemple `config/user` ou `config/ssl_keyfile`. Si vous avez changé quelque chose, n'oubliez pas de lancer la commande suivante pour que les changements aient un effet immédiat :

```
svcadm refresh svc:/application/virtualbox/webservice:default
```

Si vous oubliez la commande ci-dessus, les paramètres ci-dessus seront utilisés au moment de l'activation du service. Vérifiez les réglages actuelles des propriétés avec :

```
svccfg -p config svc:/application/virtualbox/webservice:default
```

Lorsque tout est bien configuré, vous pouvez démarrer le service web de VirtualBox avec la commande suivante :

```
svcadm enable svc:/application/virtualbox/webservice:default
```

Pour plus d'informations sur SMF, merci de vous reporter à la documentation de Solaris.

9.21.3 Mac OS X : démarrer le service web par launchd

Sur Mac OS X, on utilise launchd pour démarrer le service web de VirtualBox. Vous pouvez trouver un fichier exemple de configuration dans `$HOME/Library/LaunchAgents/org.virtualbox.vboxwebsrv.plist`. Vous pouvez l'activer en changeant la clé `Disabled` de `true` en `false`. Pour démarrer manuellement le service, utilisez la commande suivante :

```
launchctl load ~/Library/LaunchAgents/org.virtualbox.vboxwebsrv.plist
```

Pour des informations supplémentaires sur la façon dont vous pourriez configurer les services de launchd, voir <http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPSystemStartup/BPSystemStartup.html>.

9.22 VirtualBox Watchdog

À partir de VirtualBox 4.2, le service de ballon de mémoire, connu jadis comme le `VBoxBalloonCtrl`, a été renommé en `VBoxWatchdog`, ce qui intègre à présent plusieurs services de l'hôte qui sont conçus pour fonctionner dans un environnement serveur.

Il s'agit des services :

- Contrôle du ballon de mémoire, qui prend en charge automatiquement d'un ballon de mémoire configuré pour une VM (voir chapitre [4.8.1, Faire du ballon avec la mémoire](#), page [83](#) pour une présentation du jeu de ballon avec la méom:re). Cela est surtout utile pour les environnements serveurs où les VMs peuvent solliciter de manière dynamique plus ou moins de mémoire pendant leur fonctionnement.

Le service vérifie régulièrement que le ballon actuel d'une VM et sa RAM invitée disponible et il ajuste automatiquement le ballon de mémoire actuel en l'augmentant ou le réduisant selon le cas. Cette gestion ne s'applique qu'aux VMs en fonction ayant installé des suppléments invité écents.

- La détection d'un isolement de l'hôte, qui offre un moyen de détecter si l'hôte ne peut plus atteindre une instance en particulier du serveur VirtualBox et qui prend les mesures appropriées telles que l'extinction, la sauvegarde de l'état actuel, voire la coupure de certaines VMs.

Vous pouvez spécifier toutes les valeurs de configuration soit en ligne de commande, soit par des données supplémentaires globales, tandis que les valeurs en ligne de commandes ont toujours une priorité élevée si on les définit. Certaines des valeurs de configuration peuvent être également spécifiées sur une base individuelle par VM. Donc, l'ordre pour regarder les paramètres est : ligne de commande, données supplémentaires pour chaque VM (s'il y en a), données supplémentaires globales.

9.22.1 Contrôle du jeu de ballon de mémoire

Le contrôle des ballons de mémoire augmente ou réduit le ballon de mémoire des VMs à partir de la mémoire disponible sur les VMs et de la taille maximale désirée d'un ballon.

Pour régler le contrôle du jeu de ballons mémoires, il faut paramétriser la taille que peut atteindre une VM. Vous pouvez le faire en ligne de commande avec

```
--balloon-max <Taille en Mo>
```

, individuellement pour chaque VM avec les données supplémentaires avec

```
VBoxManage setextradata <VM-Name> VBoxInternal2/Watchdog/BalloonCtrl/BalloonSizeMax <Taille en Mo>
```

ou en utilisant une donnée supplémentaire globale avec

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonSizeMax <Taille en Mo>
```

Note: Si vous ne spécifiez pas de taille maximale du ballon avec au moins un des paramètres ci-dessus, vous ne pourrez faire aucun jeu de ballon.

Vous pouvez régler la taille incrémentale d'un ballon, en Mo, soit en ligne de commandes avec

```
--balloon-inc <Taille en Mo>
```

, soit en utilisant une donnée supplémentaire globale avec

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonIncrementMB <Taille en Mo>
```

La taille d'incrémentation par défaut est de 256 Mo si vous ne spécifiez rien.

La même chose marche pour une taille minimale incrémentée de ballon : en ligne de commande avec

```
--balloon-dec <Taille en Mo>
```

ou en utilisant une donnée supplémentaire globale avec

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonDecrementMB <Taille en Mo>
```

La taille minimale incrémentale d'un ballon par défaut est de 128 Mo si vous n'indiquez rien.

Pour définir la limite inférieure d'un ballon en Mo, c'est en ligne de commande avec

```
--balloon-lower-limit <Taille en Mo>
```

ou par une donnée supplémentaire globale avec

```
VBoxManage setextradata global VBoxInternal2/Watchdog/BalloonCtrl/BalloonLowerLimitMB <Taille en Mo>
```

. La limite inférieure par défaut est de 128 si vous n'indiquez rien.

9.22.2 Détection de l'isolement de l'hôte

Pour détecter si l'hôte va être isolé, c'est-à-dire qu'il ne va plus pouvoir atteindre la session du serveur VirtualBox, l'hôte doit régler une valeur dans une donnée supplémentaire pour une période de temps. Si cette valeur n'est pas définie dans le délai du timeout, une fois ce délai dépassé, ce qu'on appelle une réponse à l'isolement de l'hôte sera envoyée aux VMs gérées. Vous pouvez contrôler les VMs gérées en définissant des groupes de VM et en affectant des VMs à ces groupes. Par défaut, aucun groupe n'est défini, ce qui veut dire que toutes les VMs du serveur seront gérées lorsqu'aucune réponse hôte ne sera reçue dans les 30 secondes.

Pour définir en ligne de commandes les groupes gérés par la détection de l'isolement de l'hôte :

```
--apimon-groups=<string[,stringN]>
```

ou utilisez une donnée supplémentaire globale avec

```
VBoxManage setextradata global VBoxInternal2/Watchdog/APIMonitor/Groups <chaîne[,chaîneN]>
```

Pour définir le timeout d'isolement de l'hôte en ligne de commandes :

```
--apimon-isln-timeout=<ms>
```

ou utilisez une donnée supplémentaire globale avec

```
VBoxManage setextradata global VBoxInternal2/Watchdog/APIMonitor/IsolationTimeoutMS <ms>
```

Pour régler la réponse d'isolement finale en ligne de commandes :

--apimon-isln-response=<cmd>

ou en utilisant une donnée supplémentaire globale avec

VBoxManage setextradata global VBoxInternal2/Watchdog/APIMonitor/IsolationResponse <cmd>

Les commandes de réponse suivantes sont disponibles :

- none, qui ne fait rien.
- pause, qui met en pause l'exécution d'une VM.
- poweroff, qui éteint la VM en appuyant sur le bouton d'alimentation de la VM. La VM n'aura aucune chance de sauvegarder des données ou de lancer le processus d'extinction.
- save, qui enregistre l'état actuel de la machine et qui coupe ensuite la VM. Si la sauvegarde de l'état de la machine échoue, la VM sera mise en pause.
- shutdown, qui éteint la VM gentiment, en envoyant un événement ACPI d'extinction au système d'exploitation de la VM. L'OS a alors une chance de s'éteindre proprement.

9.22.3 Plus d'informations

Pour des options et des paramètres plus avancés comme la vérification de la verbosité de la journalisation, l'aide intégrée à la ligne de commande est accessible avec --help.

9.22.4 Linux : démarrer le service watchdog via init

Sur Linux, vous pouvez démarrer automatiquement le service watchdog lors du démarrage de l'hôte en ajoutant les paramètres adéquats au fichier /etc/default/virtualbox. Un paramètre est obligatoire, VBOXWATCHDOG_USER, vous devez le personnaliser avec l'utilisateur qui démarrera les VMs. Pour une rétro compatibilité, vous pouvez spécifier également VBOXBALLOONCTRL_USER Les paramètres du tableau suivant comment tous par VBOXWATCHDOG_(VBOXWATCHDOG_BALLOON_INTERVAL, VBOXWATCHDOG_LOGSIZE etc., et pour les paramètres qui existaient précédemment, vous pouvez utiliser les paramètres VBOXBALLOONCTRL_INTERVAL etc.) :

Paramètre	Description	Réglage par défaut
USER	L'utilisateur sous lequel fonctionne le service watchdog	
ROTATE	Nombre de fichiers journaux ; 0 désactive la gestion des journaux	10
LOGSIZE	Taille maximum du fichier journal, en octets, pour faire la gestion	1Mo
LOGINTERVAL	Intervalle de secondes maximum en secondes pour faire la rotation des journaux	1 day
BAL-LOON_INTERVAL	Intervalle de la vérification de la taille du ballon (msec)	30000
BAL-LOON_INCREMENT	Incrémentation de la taille du ballon (Mo)	256
BAL-LOON_DECREMENT	Diminution de la taille du ballon (Mo)	128
BAL-LOON_LOWERLIMIT	Limite la plus basse de la taille du ballon (Mo)	64
BAL-LOON_SAFETYMBR	Mémoire libre nécessaire pour diminuer la taille du ballon (Mo)	1024

9.22.5 Solaris : démarrer le service watchdog via SMF

Sur les hôtes Solaris, le démon du service watchdog de VirtualBox est intégré à l'environnement SMF. Vous pouvez modifier les paramètres, mais ce n'est pas obligatoire si ceux par défaut correspondent déjà à vos besoins :

```
svccfg -s svc:/application/virtualbox/balloonctrl:default setprop config/balloon_interval=10000
svccfg -s svc:/application/virtualbox/balloonctrl:default setprop config/balloon_safetymargin=134217728
```

Le tableau de la section précédente expliquant les noms des paramètres et les réglages par défaut s'applique également à Solaris. Vous devez passer les noms des paramètres en minuscules et ajouter un préfixe `config/`, par exemple `config/user` ou `config/balloon_safetymargin`. Si vous avez fait un changement, n'oubliez pas de lancer la commande suivante pour donner aux changements un effet immédiat :

```
svcadm refresh svc:/application/virtualbox/balloonctrl:default
```

Si vous oubliez la commande ci-dessus, les paramètres précédents seront utilisés lors de l'activation du service. Vérifiez les paramètres des propriétés actuels avec :

```
svccfg -p config svc:/application/virtualbox/balloonctrl:default
```

Quand tout est configuré correctement, vous pouvez démarrer le service watchdog de VirtualBox avec la commande suivante :

```
svcadm enable svc:/application/virtualbox/balloonctrl:default
```

Pour plus d'informations sur SMF, merci de vous reporter à la documentation de Solaris.

9.23 Autres packs d'extension

À partir de VirtualBox 4.2.0, il existe un autre pack d'extension, VNC, open source et qui remplace l'intégration précédente du protocole d'accès à distance VNC. C'est du code expérimental et il ne sera d'abord disponible que dans le paquet du code source de VirtualBox. Une grande partie du code est issue de contributions d'utilisateurs et elle n'est en aucun cas supportée par Oracle.

La gestion du clavier est très sérieusement limitée et seul la couche du clavier américain fonctionne. Les autres plans de clavier auront au moins quelques touches, qui produiront de mauvais résultats (avec des effets souvent très surprenants), et pour les plans ayant des différences significatives avec le plan de clavier américain, ils sont très probablement inutilisables.

Il est possible d'installer à la fois le pack d'extension VirtualBox d'Oracle VM et VNC, mais on ne peut activer qu'un module VRDE à la fois. La commande suivante passe en module VRDE de VNC dans VNC :

```
VBoxManage setproperty vrdeextpack VNC
```

La configuration de l'accès à distance fonctionne de la même façon que VRDP (voir chapitre 7.1, [Affichage distant \(VRDP support\)](#), page 109), avec quelques limites : VNC ne supporte pas la spécification de plusieurs numéros de ports et l'authentification se fait différemment. VNC ne peut gérer que l'authentification par mot de passe et il n'y a aucune possibilité d'utiliser le hachage de mots de passe. Il ne reste pas d'autre choix que de donner un mot de passe en clair dans la configuration de VNC, ce qu'on peut faire avec la commande suivante :

```
VBoxManage modifyvm NOMVM --vrdeproperty VNCPassword=secret
```

L'utilisateur est responsable du secret de son mot de passe et vous devriez le supprimer quand vous donnez la configuration d'une VM à quelqu'un d'autre, quelle que soit la finalité. Certains serveurs VNC prétendent qu'ils gardent le mot de passe "chiffré" dans leur configuration. Ce n'est pas du vrai chiffrement, ce ne sont que des mots de passe, ce qui est exactement aussi sécurisé que les mots de passe en clair.

La commande suivante revient à VRDP (s'il est installé) :

```
VBoxManage setproperty vrdeextpack "Oracle VM VirtualBox Extension Pack"
```

9.24 Démarrer des machines virtuelles lors de l'amorçage du système

À partir de VirtualBox 4.2.0, il est possible de démarrer des VMs automatiquement à l'amorçage du système sur Linux, Solaris et Mac OS X, pour tous les utilisateurs.

9.24.1 Linux : démarrer le service autostart par init

Sur Linux, le service autostart s'active en définissant deux variables de `/etc/default/virtualbox`. La première est `VBOXAUTOSTART_DB`, qui contient un chemin absolu vers le répertoire de la base de données existante. Tous les utilisateurs devraient avoir un accès en écriture au répertoire pour démarrer automatiquement des machines virtuelles. En outre, vous devriez donner au répertoire le bit sticky. La deuxième variable est `VBOXAUTOSTART_CONFIG`, qui fait pointer le service vers le fichier de configuration d'autostart utilisé lors du démarrage pour déterminer s'il faut autoriser des utilisateurs individuels à démarrer une VM automatiquement et les délais de démarrage de la configuration. Vous pouvez mettre le fichier de configuration dans `/etc/vbox` et il contient plusieurs options. Une s'appelle `default_policy` qui contrôle si le service autostart autorise ou non les utilisateurs non dans la liste d'exceptions à démarrer des VMs. La liste d'exceptions commence par `exception_list` et elle contient une liste de nom d'utilisateurs séparée par des virgules. De plus, vous pouvez configurer un délai de démarrage propre à chaque utilisateur pour éviter une surcharge de l'hôte. Une configuration modèle est présentée ci-dessous :

```
# La politique par défaut est d'interdire le démarrage d'une VM, l'autre
# choix étant "allow".
default_policy = deny

# Bob est autorisé à démarrer des machines virtuelles, mais chacun à intervalle
# de 10 secondes
bob = {
    allow = true
    startup_delay = 10
}

# Alice n'est pas autorisé à démarrer des machines virtuelles, utile pour
# exclure certains utilisateurs si la politique par défaut est allow.
alice = {
    allow = false
}
```

Tout utilisateur voulant activer autostart pour des machines en particulier doit définir le chemin du répertoire de la base de données autostart avec

```
VBoxManage setproperty autostartdbpath <Autostart directory>
```

9.24.2 Solaris : démarrer le service autostart par SMF

Sur les hôtes Solaris, Le démon autostart de VirtualBox est intégré à l'environnement SMF. Pour l'activer, vous devez faire pointer le service vers un fichier de configuration existant qui est au même format que sur Linux (voir chapitre [9.24.1, Linux : démarrer le service autostart par init](#), page [194](#)) :

```
svccfg -s svc:/application/virtualbox/autostart:default setprop config/config=/etc/vbox/autostart.cfg
```

Quand tout est bien configuré, vous pouvez démarrer le service autostart de VirtualBox avec la commande suivante :

```
svcadm enable svc:/application/virtualbox/autostart:default
```

Pour plus d'informations sur SMF, merci de vous reporter à la documentation de Solaris.

9.24.3 Mac OS X : démarrer le service autostart par launchd

Sur Mac OS X, on utilise launchd pour démarrer le service autostart de VirtualBox. Vous pouvez trouver un fichier de configuration exemple dans /Applications/VirtualBox.app/Contents/MacOS/org.virtualbox.plist. Pour activer le service, copiez le fichier dans /Library/LaunchDaemons et passez la clé Disabled de true à false. Par ailleurs, remplacez le deuxième paramètre par un fichier de configuration existant et qui est au même format que sur Linux (voir chapitre 9.24.1, *Linux : démarrer le service autostart par init*, page 194). Pour démarrer le service à la main, utilisez la commande suivante :

```
launchctl load /Library/LaunchDaemons/org.virtualbox.vboxautostart.plist
```

Pour avoir des informations supplémentaires sur la façon dont les services launchd pourraient se configurer, voir <http://developer.apple.com/mac/library/documentation/MacOSX/Conceptual/BPSystemStartup/BPSystemStartup.html>.

10 Sous-basements techniques

Le contenu de ce chapitre n'est pas indispensable pour utiliser VirtualBox avec succès. Nous indiquons ce qui suit à titre informatif pour ceux qui sont plus familiers de la technologie et de l'architecture informatique et qui veulent en savoir davantage sur la manière fonctionne VirtualBox "sous le capot".

10.1 Où VirtualBox stocke ses fichiers

Dans VirtualBox, une machine virtuelle et ses paramètres sont décrits dans un fichier de paramètres de la machine virtuelle, au format XML. De plus, la plupart des machines virtuelles ont un ou plusieurs disques durs qui leur sont en général présentés par des images de disque (comme au format VDI). L'endroit où sont stockés tous ces fichiers dépend de la version de VirtualBox qui a créé la machine.

10.1.1 Machines créées par VirtualBox version 4.0 ou supérieur

À partir de la version 4.0, par défaut, chaque machine virtuelle dispose d'un répertoire sur votre ordinateur hôte (où tous les fichiers de cette machine sont stockés – le fichier des paramètres XML (avec une extension de fichier .vbox) et ses images de disque).

Par défaut, ce "dossier machine" se trouve dans un dossier ordinaire appelé "VirtualBox VMs", créé par VirtualBox dans le dossier personnel de l'utilisateur du système actuel. L'emplacement de ce répertoire personnel dépend des conventions du système d'exploitation hôte :

- Sur Windows, il s'agit de %HOMEDRIVE%%HOMEPATH%; en général quelque chose comme C:\Documents and Settings\NomUtilisateur\.
- Sur Mac OS X, il s'agit de /Users/nomutilisateur.
- Sur Linux et Solaris, il s'agit de /home/nomutilisateur.

Par simplicité, nous abrègerons cela ci-dessous par \$HOME. En utilisant cette convention, le dossier ordinaire de toutes les machines virtuelles est \$HOME/VirtualBox VMs.

Par exemple, quand vous créez une machine virtuelle qui s'appelle "VM Exemple", vous verrez que VirtualBox crée

1. le dossier \$HOME/VirtualBox VMs/VM Exemple/ et, dans ce dossier,
2. le fichier des paramètres VM Exemple.vbox et
3. l'image de disque virtuel VM Example.vdi.

C'est le rangement par défaut si vous utilisez l'assistant "Créer une nouvelle machine virtuelle" comme décrit au chapitre [1.7, Créez votre première machine virtuelle](#), page [19](#). Une fois que vous commencez à travailler avec la VM, des fichiers supplémentaires apparaîtront : vous trouverez des fichiers journaux dans un sous-dossier qui s'appelle Logs, and une fois que vous aurez pris des instantanés, ils apparaîtront dans un sous-dossier Snapshots. Pour chaque VM, vous pouvez modifier l'emplacement de son dossier d'instantanés dans les paramètres de la VM.

Vous pouvez changer le dossier machine par défaut en sélectionnant "Préférences" du menu "Fichier" de la fenêtre principale de VirtualBox. Puis, dans la fenêtre qui apparaît, cliquez sur l'onglet "Général". Sinon, utilisez `VBoxManage setproperty machinefolder ;` voir le chapitre [8.27, VBoxManage setproperty](#), page [150](#).

10.1.2 Machines créées par des versions de VirtualBox antérieures à 4.0

Si vous avez mis à jour vers VirtualBox 4.0 en partant d'une ancienne version de VirtualBox, vous aurez probablement vos fichiers de paramètres et les disques selon l'organisation du système de fichiers d'alors.

Avant la version 4.0, VirtualBox séparait les fichiers des paramètres de la machine des images de disque virtuel. Les fichiers de paramétrages de la machine avaient une extension `.xml` et se trouvaient dans un dossier appelé "Machines" dans le répertoire de configuration global de VirtualBox (voir la prochaine section). Donc, par exemple, sur Linux, il s'agissait du répertoire caché `$HOME/.VirtualBox/Machines`. Le dossier par défaut des disques durs s'appelait "Hard-Disks" et se trouvait également dans le dossier `.VirtualBox`. L'utilisateur pouvait changer les deux endroits dans les préférences globales (le concept de "dossier par défaut des disques durs" a été abandonné avec VirtualBox 4.0, vu que les images de disque se trouvent désormais par défaut dans le dossier de chaque machine.)

L'ancienne organisation avait plusieurs gros inconvénients.

1. Il était très difficile de déplacer une machine virtuelle d'un hôte à l'autre car les fichiers concernés ne se trouvaient pas dans le même dossier. De plus, les médias virtuels de toutes les machines étaient enregistrés avec un registre global dans le fichier des paramètres transversaux de VirtualBox. (`$HOME/.VirtualBox/VirtualBox.xml`).

Pour déplacer une machine sur un autre hôte, il n'était donc pas suffisant de déplacer le fichier des paramètres XML et les images de disque (qui se trouvaient à des endroits différents), mais il fallait en plus copier méticuleusement les entrées du disque dur à partir du XML du registre de médias global, ce qui était presqu'impossible si la machine avait des instantanés et, donc, des images de différenciation.

2. Le stockage des images de disque virtuel, qui peuvent beaucoup grossir, sous le répertoire caché `.VirtualBox` (au moins sur les hôtés Linux et Solaris) amenait de nombreux utilisateurs à se demander ce qu'était devenu leur espace disque.

Si les nouvelles VMs créées avec VirtualBox 4.0 ou supérieur respecteront la nouvelle organisation, pour une compatibilité maximum, les anciennes VMs *ne sont pas* converties en nouvelle organisation. Sans cela, les paramètres de la machine seraient immanquablement cassés si l'utilisateur rétrogradait de la 4.0 à une version plus ancienne de VirtualBox.

10.1.3 Données globales de configuration

Outre les fichiers des machines virtuelles, VirtualBox gère des données globales de configuration. Sur Linux et Solaris, depuis as of VirtualBox 4.3 elles se trouvent dans le répertoire caché `$HOME/.config/VirtualBox` même si `$HOME/.VirtualBox` sera utilisé s'il existe pour rester compatible avec les anciennes versions ; sur un Mac, elles se trouvent dans `$HOME/Library/VirtualBox`.

VirtualBox crée automatiquement ce répertoire de configuration si nécessaire. Vous pouvez éventuellement fournir un répertoire de configuration alternatif en réglant la variable d'environnement `VBOX_USER_HOME` ou, en plus, sur Linux ou Solaris, en utilisant la variable standard `XDG_CONFIG_HOME` (car le fichier des paramètres globaux de `VirtualBox.xml` pointe vers tous les autres fichiers de configuration, ce qui permet de naviguer entre plusieurs configurations de VirtualBox).

VirtualBox stocke essentiellement dans ce répertoire son fichier de paramètres globaux, un autre fichier XML appelé `VirtualBox.xml`. Cela comprend des options de configuration globales et la liste des machines virtuelles enregistrées avec des pointeurs vers leurs fichiers de paramètres XML. Ni l'emplacement du fichier ni son répertoire n'ont changé avec VirtualBox 4.0.)

Avant VirtualBox 4.0, tous les médias virtuels (fichiers images de disque) étaient également stockés dans un registre global de ce fichier de paramètres. Par compatibilité, ce registre de

médias existe toujours si vous mettez à jour VirtualBox et s'il y a des médias issus de machines créées avec une version inférieure à 4.0. Si vous n'avez pas de telles machines, ce ne sera pas des retistres de médias globaux ; avec VirtualBox 4.0, chaque fichier XML d'une machine a son propre registre de médias.

De même, avant VirtualBox 4.0, le dossier “Machines” par défaut et le dossier “HardDisks” par défaut se trouvaient dans le répertoire de configuration de VirtualBox (par exemple, `$HOME/.VirtualBox/Machines` sur Linux). Si vous mettez à jour à partir d'une version de VirtualBox inférieure à la 4.0, les fichiers de ce répertoire ne sont pas déplacés automatiquement afin de ne pas casser la rétro compatibilité.

10.1.4 Résumé des modifications de la configuration de 4.0

	Avant 4.0	4.0 ou supérieur
Dossier par défaut des machines	<code>\$HOME/.VirtualBox/Machines</code>	<code>\$HOME/VirtualBox VMs</code>
Emplacement des images de disque	<code>\$HOME/.VirtualBox/HardDisks</code>	<code>Each machine's folder</code>
Extension des fichiers de paramètres de la machine	<code>.xml</code>	<code>.vbox</code>
Registre de médias	Fichier <code>VirtualBox.xml</code> global	Chaque fichier des paramètres d'une machine
Enregistrement des médias	Ouverture/fermeture explicite obligatoire	Automatique après la connexion

10.1.5 Fichiers XML de VirtualBox

VirtualBox utilise l'XML tant pour les fichiers des paramètres de la machine que pour le fichier de configuration global, `VirtualBox.xml`.

Tous les fichiers XML de VirtualBox sont versionnés. Quand un nouveau fichier de paramètres est créé (par exemple parce qu'on crée une nouvelle machine virtuelle), VirtualBox utilise automatiquement le format des paramètres de la version actuelle de VirtualBox. Il se peut que ces fichiers ne soient pas lus si vous rétrogradez à une version plus ancienne de VirtualBox. Cependant, quand VirtualBox rencontre un fichier de paramètres d'une ancienne version (comme après une mise à jour de VirtualBox), il essaie autant que possible de garder le format des paramètres. Il ne mettra à jour en silence les fichiers des paramètres que si les paramètres actuels ne peuvent pas être exprimés dans l'ancien format, par exemple parce que vous avez activé une fonction qui n'était pas présente dans l'ancienne version de VirtualBox.¹ Dans ces cas-là, VirtualBox sauvegarde le fichier des anciens paramètres dans le répertoire de configuration de la machine virtuelle. Si vous avez besoin de revenir à une ancienne version de VirtualBox, vous devrez recopier à la main ces fichiers de sauvegarde.

Nous ne documentons volontairement pas les spécifications des fichiers XML de VirtualBox car nous nous réservons le droit de les modifier à l'avenir. Nous vous suggérons donc fortement de ne pas éditer ces fichiers à la main. VirtualBox offre un accès complet à ses données de configuration par son outil en ligne de commande `VBoxManage` (voir le chapitre 8, [VBoxManage](#), page 119) et son API (voir le chapitre 11, [Interfaces de programmation de VirtualBox](#), page 207).

¹Par exemple, avant VirtualBox 3.1, il n'était possible que d'activer ou de désactiver un seul lecteur DVD dans une machine virtuelle. S'il a été activé, cela serait toujours possible sur le deuxième maître du contrôleur IDE. Avec VirtualBox 3.1, on peut connecter des lecteurs DVD à un slot de son choix sur un contrôleur de son choix, donc ils pourraient être sur le deuxième esclave d'un contrôleur IDE ou sur un slot SATA. Si vous avez un fichier de paramètres d'une machine d'une ancienne version et si vous mettez à jour VirtualBox vers la 3.1 et si vous déplacez le lecteur DVD de sa position par défaut, on ne peut pas l'exprimer dans l'ancien format des paramètres ; le fichier XML de la machine serait écrit dans le nouveau format et une copie de sauvegarde de l'ancien format serait gardée.

10.2 Exécutables et composants de VirtualBox

VirtualBox a été conçu pour être modulaire et flexible. Quand on ouvre l'interface graphique (GUI) de VirtualBox et qu'on démarre une VM, au moins trois processus fonctionnent :

1. **VBoxSVC**, le processus du service de VirtualBox qui fonctionne toujours en tâche de fond. Ce processus est lancé automatiquement par le processus du premier client VirtualBox (la GUI, **VBoxManage**, **VBoxHeadless**, le service web ou autres) et il s'arrête peu de temps après que le dernier client a quitté. Le service est responsable d'archiver, maintenir l'état de toutes les VMS et de la communication entre les composants de VirtualBox. Cette communication est implémentée via COM/XPCOM.

Note: Quand nous parlons de “clients” ici, nous voulons dire les clients locaux d'un processus serveur **VBoxSVC** en particulier, pas les clients sur un réseau. VirtualBox utilise son propre concept client/serveur pour permettre à ses processus de coopérer, mais tous ces processus tournent sous le même compte utilisateur du système d'exploitation hôte, et c'est entièrement transparent pour l'utilisateur.

2. Le processus de la GUI, **VirtualBox**, une application client basée sur la bibliothèque multiplateformes Qt. Lancée sans l'option `--startvm`, cette application agit comme un gestionnaire de VirtualBox, en affichant les VMs et leurs paramètres. Elle communique alors les paramètres et les changements d'état à **VBoxSVC** et elle répercute les changements subis par d'autres moyens comme **VBoxManage**.
3. Si on lance l'application client **VirtualBox** avec l'argument `--startvm`, elle charge la bibliothèque VMM qui inclut l'hyperviseur proprement dit et qui lance une machine virtuelle et offre une entrée et une sortie à l'invité.

Toutes les interfaces de VirtualBox (client) communiqueront avec le processus du service et elles peuvent contrôler et répercuter l'état actuel. Par exemple, tant le selecteur de VM que la fenêtre de VM ou **VBoxManage** peuvent être utilisés pour mettre en pause la VM en fonction, les autres composants reflèteront toujours le changement d'état.

La GUI de VirtualBox n'est qu'une des nombreuses interfaces (client) disponibles. La liste complète comprise dans VirtualBox est :

1. **VirtualBox**, l'interface Qt implémentant le gestionnaire et les VMS en fonction ;
2. **VBoxManage**, une alternative moins conviviale mais plus puissante, décrite au chapitre 8, [VBoxManage](#), page 119.
3. **VBoxSDL**, une interface graphique simple basée sur la bibliothèque SDL ; voir chapitre 9.1, [VBoxSDL, l'afficheur simplifié de VM](#), page 161.
4. **VBoxHeadless**, une interface de VM qui ne fournit pas directement de sortie graphique et d'entrée clavier/souris, mais qui permet une redirection par VirtualBox Remote Desktop Extension; voir chapitre 7.1.2, [VBoxHeadless, le serveur de bureau distant](#), page 110.
5. **vboxwebsrv**, le processus du service web de VirtualBox qui permet de contrôler un hôte VirtualBox à distance. Ceci est décrit en détails dans le manuel de référence du VirtualBox Software Development Kit (SDK) ; merci de voir le chapitre 11, [Interfaces de programmation de VirtualBox](#), page 207 pour des détails.
6. Le shell Python de VirtualBox, une alternative en Python à **VBoxManage**. Elle est aussi décrite dans le manuel de référence du SDK.

En interne, VirtualBox consiste beaucoup plus d'interfaces séparées. Vous pourriez les rencontrer en analysant les messages d'erreur internes ou les fichiers journaux. Parmi elles, on compte :

- IPRT, une bibliothèque d'exécution portable qui forme une couche d'abstraction d'accès aux fichiers, du filage (threading), la manipulation de chaînes, etc. Chaque fois que VirtualBox accède aux fonctions du système hôte, il le fait via cette bibliothèque pour une portabilité multiplateformes.
- VMM (Virtual Machine Monitor), le cœur de l'hyperviseur.
- EM (Execution Manager), contrôle l'exécution d'un code invité.
- REM (Recompiled Execution Monitor), fournit une émulation logicielle des instructions du processeur.
- TRPM (Trap Manager), intercepte et traite les traps et les exceptions de l'invité.
- HWACCM (Hardware Acceleration Manager), offre un support pour VT-x et AMD-V.
- PDM (Pluggable Device Manager), une interface abstraite entre le VMM et les périphériques émulés qui sépare les implémentations du périphérique de l'intérieur du VMM et qui facilite l'ajout de nouveaux périphériques émulés. Par PDM, des développeurs tiers peuvent ajouter de nouveaux périphériques virtuels à VirtualBox, sans devoir modifier VirtualBox lui-même.
- PGM (Page Manager), un composant contrôlant la pagination de l'invité.
- PATM (Patch Manager), corrige le code de l'invité pour améliorer et accélérer la virtualisation logicielle.
- TM (Time Manager), gère les horloges et tous les aspects de l'heure des invités.
- CFGM (Configuration Manager), fournit une structure arborescente qui garde les paramètres de configuration de la VM et tous les périphériques émulés.
- SSM (Saved State Manager), enregistre et charge l'état d'une VM.
- VUSB (Virtual USB), une couche USB qui sépare les contrôleurs USB émulés des contrôleurs de l'hôte et des périphériques USB ; ceci active également l'USB distant.
- DBGF (Debug Facility), un débogueur de VM intégré.
- VirtualBox émule un certain nombre de périphériques pour offrir l'environnement matériel dont ont besoin divers invités. La plupart de ces périphériques standards se trouvent dans beaucoup de machines compatibles PC et sont largement supportés par les systèmes d'exploitation invités. Pour les périphériques réseaux et de stockage en particulier, il existe plusieurs options pour que les périphériques émulés accèdent au matériel sous-jacent. Ces périphériques sont gérés par PDM.
- Les suppléments invité pour divers systèmes d'exploitation invités. Il s'agit de code installé dans les machines virtuelles ; voir chapitre 4, *Les suppléments invité*, page 64.
- Le composant “Main” est spécial : il croise tous les bits ci-dessus et c'est la seule API publique fournie par VirtualBox. Tous les processus clients listés ci-dessus n'utilisent que cette API et n'accèdent jamais directement aux composants de l'hyperviseur. Il s'en suit que des applications tierces utilisant l'API principale de VirtualBox peuvent s'appuyer sur le fait qu'elle est toujours bien testée et que toutes les possibilités de VirtualBox sont complètement présentées. C'est cette API qui est décrite dans le manuel de référence du SDK de VirtualBox indiqué ci-dessus (de nouveau, voir le chapitre 11, *Interfaces de programmation de VirtualBox*, page 207).

10.3 Virtualisation matérielle vs. logicielle

VirtualBox permet aux logiciels de la machine virtuelle de s'exécuter directement sur le processeur de l'hôte, mais il utilise une gamme de techniques complexes pour intercepter les opérations interférant avec votre hôte. Chaque fois que l'invité essaie de faire quelque chose de potentiellement dangereux pour votre ordinateur et ses données, VirtualBox s'interpose et rentre en action. En particulier, pour beaucoup de matériel auquel croit avoir accès l'invité, VirtualBox simule un certain environnement "virtuel" selon la façon dont vous avez configuré une machine virtuelle. Par exemple, quand l'invité cherche à accéder à un disque dur, VirtualBox redirige ces requêtes vers ce que vous avez configuré comme étant le disque dur virtuel de la machine virtuelle – en principe, un fichier image sur votre hôte.

Malheureusement, la plateforme x86 n'a jamais été conçue pour pour être virtualisée. La détection des situations où VirtualBox doit contrôler le code invité qui s'exécute, comme décrit ci-dessus, est difficile. Il existe deux façons de faire cela :

- Depuis 2006, les processeurs Intel et AMD supportent ce qu'on appelle la "**virtualisation matérielle**". Cela signifie que ces processeurs peuvent aider VirtualBox à intercepter des opérations potentiellement dangereuses que pourrait essayer de faire le système d'exploitation invité et ils facilitent la présentation de matériel virtuel à une machine virtuelle.

Ces fonctionnalités du matériel diffèrent entre les processeurs Intel et AMD. Intel a appelé sa technologie **VT-x** ; AMD a nommé la leur **AMD-V**. Le support d'Intel et d'AMD de la virtualisation est très différent dans le détail, mais pas si différent dans le principe.

Note: Sur de nombreux systèmes, les fonctions de virtualisation matérielle doivent être préalablement activées dans le BIOS avant de pouvoir être utilisées par VirtualBox.

- Contrairement aux autres logiciels de virtualisation, pour de nombreux scénarios d'utilisation, VirtualBox *n'exige pas* que les fonctions de virtualisation matérielle soient présentes. Par des techniques sophistiquées, VirtualBox virtualise beaucoup de systèmes d'exploitation invités complets de manière **logicielle**. Cela signifie que vous pouvez lancer des machines virtuelles même sur d'anciens processeurs qui ne supportent pas la virtualisation matérielle.

Même si VirtualBox n'exige pas toujours la virtualisation matérielle, son activation est *nécessaire* dans les scénarios suivants :

- Certains systèmes d'exploitation, rares, comme OS/2, utilisent des instructions processeur très ésoptiques qui ne sont pas supportées par notre virtualisation logicielle. Pour les machines virtuelles configurées pour contenir un tel système d'exploitation, la virtualisation matérielle est activée automatiquement.
- Le support des invités 64 bits de VirtualBox (ajouté avec la version 2.0) et le multiprocesseur (SMP, ajouté avec la version 3.0) exigent tous deux l'activation de la virtualisation matérielle (ce n'est tout de même pas une grosse limite vu l'immense majorité des processeurs 64 bits et multi coeurs actuels incluant la virtualisation matérielle ; les exceptions à cette règle étant par exemple les anciens processeurs Intel Celeron et AMD Opteron.)

Avertissement: Ne lancez pas d'autres hyperviseurs (produits de virtualisation open-source ou propriétaires) en même temps que VirtualBox ! Si plusieurs hyperviseurs peuvent, en principe, être *installés* en parallèle, n'essayez pas de *lancer* plusieurs machines virtuelles à partir d'hyperviseurs concurrents en même temps. VirtualBox ne peut pas savoir ce qu'un autre hyperviseur essaie de faire sur un même hôte, et surtout si plusieurs produits essaient d'utiliser la virtualisation matérielle, les fonctions telles que VT-x, cela peut planter tout l'hôte. De plus, dans VirtualBox, vous pouvez mélanger la virtualisation logicielle et matérielle quand vous lancez plusieurs VMs. Dans certains cas, une petite perte de performances sera inévitable si vous mélangez des VMs avec virtualisation VT-x et logicielle. Nous recommandons de ne pas mélanger les modes de virtualisation si la performance maximum et une faible overhead sont essentiels. Cela ne s'applique pas à AMD-V.

10.4 Détails sur la virtualisation logicielle

L'implémentation de la virtualisation sur les processeurs x86 sans le support de la virtualisation matérielle est une tâche extraordinairement complexe car l'architecture du processeur n'a pas été conçue pour être virtualisée. On peut résoudre en général les problèmes, mais au prix de performances réduites. Ainsi, il existe un conflit constant entre les performances de virtualisation et et son soin.

Le jeu d'instructions x86 a été conçu au départ dans les années 1970 et subi des modifications significatives avec l'ajout d'un mode protégé dans les années 1980s avec l'architecture du processeur 286, puis à nouveau avec l'Intel 386 et l'architecture 32 bits. Alors que le 386 avait un support de virtualisation vraiment limité pour les opérations en mode réel, (le mode V86, utilisé par la "DOS Box" de Windows 3.x et d'OS/2 2.x), aucun port n'existe pour virtualiser toute l'architecture.

En théorie, la virtualisation logicielle n'est pas complexe en soi. Outre les quatre niveaux de priviléges ("rings") fournis par le matériel (dont en général on n'utilise que deux : ring 0 pour le mode noyau et ring 3 pour le mode utilisateur), il faut faire la différence entre le "contexte hôte" et le "contexte invité".

Dans le "contexte hôte", tout est comme s'il n'y avait pas d'hyperviseur actif. Cela pourrait être le mode actif si une autre application de votre hôte consomme du temps processeur ; dans ce cas, il existe un mode ring 3 hôte et un mode ring 0 hôte. L'hyperviseur n'est pas impliqué.

Par contre, dans le "contexte invité", une machine virtuelle est active. Tant que le code invité s'exécute en ring 3, ce n'est pas très problématique vu qu'un hyperviseur peut paramétriser les tableaux des pages correctement et exécuter ce code de manière native sur le processeur. Les problèmes arrivent sur la manière d'intercepter ce que fait le noyau de l'invité.

Il y a plusieurs solutions possibles à ces problèmes. Une approche est l'émulation logicielle totale, ce qui implique généralement une recompilation. A savoir que tout le code qui doit être exécuté par l'invité est analysé, transformé sous une forme qui n'autorisera pas l'invité à modifier et à voir l'état réel du processeur, lequel l'exécutera simplement. Ce processus est bien sûr très complexe et coûteux en termes de performances. (VirtualBox contient un recompilateur basé sur QEMU qu'on peut utiliser pour une émulation logicielle pure, mais le recompilateur n'est activé que dans des situations particulières, décrites ci-dessous.)

Une autre solution possible est la paravirtualisation, où seuls les OS invités spécialement modifiés sont autorisés à s'exécuter. De cette manière, la plupart des accès matériels sont rendus abstraits et toutes les fonctions qui accèderaient normalement au matériel ou à l'état privilégié du processeur se basent plutôt sur l'hyperviseur. La paravirtualisation peut donner de bonnes fonctionnalités et de bonnes performances sur des processeurs x86 standards, mais cela ne peut marcher que si l'OS invité peut être modifié, ce qui n'est évidemment pas toujours le cas.

VirtualBox choisit une approche différente. Quand on démarre une machine virtuelle par son pilote noyau du support ring-0, VirtualBox a réglé le système hôte pour qu'il puisse lancer

nativement la plupart du code invité, mais il s'insère lui-même “en bas” de l'image. Il peut alors supposer le contrôle lorsque c'est nécessaire – si une instruction privilégiée est exécutée, l'invité plante (traps) (en particulier car un accès au registre E/S a été tenté et un périphérique doit être virtualisé) ou car des interruptions se produisent. VirtualBox peut alors gérer cela et soit acheminer une requête vers un périphérique virtuel, soit, si possible, déléguer la gestion de tels éléments à l'OS hôte ou invité. Dans le contexte invité, VirtualBox peut être donc dans un des trois états :

- Le code invité ring 3 s'exécute sans modifications, à pleine vitesse, autant que possible. Le nombre de fautes sera généralement faible (sauf si l'invité autorise l'E/S du port depuis ring 3, chose que nous ne pouvons pas faire car nous ne voulons pas que l'invité puisse accéder aux ports réels). On parle aussi de “mode brut”, car le code ring-3 de l'invité s'exécute sans modifications.
- Mour le code invité en ring 0, VirtualBox utilise une astuce savoureuse : il reconfigure l'invité pour que son code ring-0 se lance plutôt en ring 1 (ce qui n'est en principe pas utilisé sur les systèmes d'exploitation x86). Il s'en suit que lorsque le code ring-0 de l'invité (qui s'exécute en fait en ring 1) tel que le pilote d'un périphérique invité, essaie d'écrire sur un registre E/S ou d'exécuter une instruction non privilégiée, l'hyperviseur de VirtualBox en ring 0 “réel” peut prendre le dessus.
- L'hyperviseur (VMM) peut être actif. Chaque fois qu'une erreur survient, VirtualBox regarde l'instruction problématique et il peut la reléguer à un périphérique virtuel, à l'OS hôte, à l'invité ou il peut le lancer dans le recompilateur.

En particulier, on utilise le recompilateur quand le code invité désactive les interruptions et VirtualBox ne peut pas savoir quand on y reviendra (dans ces situations, VirtualBox analyse en fait le code invité en utilisant son propre désassemblleur). De plus, certaines instructions privilégiées telles que LIDT doivent être gérées à part. Enfin, tout le code en mode réel ou protégé (comme le code du BIOS, un invité DOS ou un démarrage de système d'exploitation) se lance complètement dans un recompilateur.

Malheureusement, cela ne fonctionne que dans une certaine mesure. Entre autres, les situations suivantes nécessitent une gestion spéciale :

1. L'exécution de code ring 0 en ring 1 provoque beaucoup d'erreurs d'instructions supplémentaires car ring 1 n'est pas autorisé à exécuter des instructions privilégiées (dont le ring-0 de l'invité en contient beaucoup). Avec chacune de ces erreurs, le VMM doit s'arrêter et émuler le code pour obtenir le comportement désiré. Si cela fonctionne, l'émulation de milliers d'erreurs est très coûteuse et très pénalisante en performances de l'invité virtualisé.
2. Il existe des défauts dans l'implémentation de ring 1 de l'architecture x86 qui n'ont jamais été corrigés. Certaines instructions qui *planteraient* même en ring 1 ne le font pas. Cela concerne par exemple les paires d'instructions LGDT/SGDT, LIDT/SIDT, ou POPF/PUSHF. Alors que l'opération “load” est privilégiée et peut donc planter, l'instruction “store” réussit toujours. Si l'invité est autorisé à les exécuter, il verra l'état réel du PC et pas celui virtualisé. L'instruction CPUID a également le même problème.
3. Un hyperviseur a en général besoin de réservé certaines parties de l'espace d'adresse de l'invité (tant l'espace d'adresse liénaire que les sélecteurs) pour son propre usage. Ce n'est pas complètement transparent pour l'OS invité et cela peut provoquer des conflits.
4. L'instruction SYSENTER (utilisée pour les appels système) exécutée par une application en fonction dans un OS invité transite toujours par le ring 0. Mais c'est là où l'hyperviseur se lance et pas l'OS invité. Dans ce cas, l'hyperviseur doit bloquer et émuler l'instruction même quand ce n'est pas souhaitable.

5. Les registres de segments du processeur contiennent un cache de descripteur “caché” inaccessible de manière logicielle. L'hyperviseur ne peut pas lire, enregistrer ou restaurer cet état, mais l'OS invité peut l'utiliser.
6. Certaines ressources doivent (et peuvent) être neutralisées par l'hyperviseur, mais l'accès est si fréquent que cela crée une perte significative de performances. Un exemple réside dans le registre TPR (Task Priority) en mode 32 bits. Les accès à ce registre doivent être bloqués par l'hyperviseur, mais certains systèmes d'exploitation invités (en particulier Windows et Solaris) écrivent très souvent dans ce registre, ce qui porte une atteinte certaine aux performances de virtualisation.

Pour corriger ces problèmes de performances et de sécurité, VirtualBox contient un gestionnaire d'analyse et de scan de code (Code Scanning and Analysis Manager (CSAM)), qui désassemble le code invité, et un gestionnaire de correctifs (Patch Manager (PATM)), qui peut le remplacer pendant l'exécution.

Avant d'exécuter du code ring 0, CSAM le scanne de manière récursive pour trouver des instructions problématiques. PATM le corrige *in-situ*, c'est-à-dire qu'il remplace l'instruction par un passage à la mémoire de l'hyperviseur, où un générateur intégré a mis une implémentation plus convenable. En réalité, c'est une tâche très complexe car il existe de nombreuses situations compliquées à trouver et à gérer correctement. Donc, vu son actuelle complexité, vous pourriez trouver que PATM est un recompilateur avancé *in-situ* recompiler.

De plus, à chaque fois qu'une erreur survient, VirtualBox analyse le code problématique pour déterminer s'il est possible de le corriger afin de l'empêcher de provoquer davantage futures d'erreurs. Cette approche fonctionne bien en pratique et améliore de façon drastique les performances de la virtualisation logicielle.

10.5 Détails sur la virtualisation matérielle

Avec VT-x d'Intel, il existe deux modes opératoires du processeur : le mode racine VMM et le mode non-racine.

- En mode racine, le processeur se comporte beaucoup comme les anciennes générations de processeurs sans le support VT-x. Il y a quatre niveaux de priviléges (“rings”) et le même jeu d'instructions est supporté avec, en plus, des instructions spécifiques de virtualisation. Le mode racine est ce que le système d'exploitation hôte utilise sans virtualisation, et il est aussi utilisé par l'hyperviseur quand la virtualisation est active.
- En mode non-racine, le fonctionnement du processeur est très différent. Il y a toujours quatre niveaux de priviléges et le même jeu d'instructions, mais une nouvelle structure, qui s'appelle VMCS (Virtual Machine Control Structure), contrôle désormais le fonctionnement du processeur et elle détermine la manière dont se comportent certaines instructions. Le mode non-racine est celui dans lequel les systèmes invités fonctionnent.

Le passage du mode racine au mode non racine s'appelle “l'entrée VM”, celui en sens invers s'appelle “Quitter VM”. Le VMCS inclut une zone d'état invité et hôte sauvegardée/restaurée à chaque entrée et sortie en VM. Surtout, les VMMS contrôlent les opérations de l'invité qui feront quitter la VM.

Les VMCS permettent un contrôle très fin via ce que les invités peuvent et ne peuvent pas faire. Par exemple, un hyperviseur peut autoriser un invité à écrire certains bits dans des registres de contrôle protégés, mais pas dans d'autres. Cela permet une virtualisation efficace dans des cas où les invités peuvent être autorisés à écrire des bits de contrôle sans gêner l'hyperviseur, tout en les empêchant de modifier les bits de contrôle dont l'hyperviseur a besoin pour avoir un contrôle total. Le VMMS fournit aussi un contrôle via l'affichage d'interruptions et les exceptions.

Chaque fois qu'une instruction ou un événement fait quitter une VM, le VMCS contient des informations sur les raisons de la sortie, ainsi que, souvent, des détails environnants. Par exemple,

si une écriture dans le registre CR0 fait quitter, l'instruction en cause est enregistrée, ainsi que le fait qu'un accès en écriture sur le registre de contrôle a prurqué la sortie, et des informations sur la le registre source et destination. L'hyperviseur peut ainsi gérer efficacement la condition sans avoir besoin de techniques avancées telles que CSAM et PATM décrits ci-dessus.

VT-x évite intrinsèquement plusieurs problèmes qui se posent avec la virtualisation logicielle. L'invité a son propre espace d'adresse distinct, qu'il ne partage pas avec l'hyperviseur, ce qui élimine les plantages potentiels. De plus, le code du noyau de l'OS invité se lance avec le privilège ring 0 en mode non racine VMX, rendant inopérants les problèmes d'exécution de code en ring 0 sur des niveaux moins privilégiés. Par exemple, l'instruction SYSENTER peut faire une transition vers le ring 0 sans problèmes. Naturellement, même en ring 0 en mode non-racine VMX, tous les accès E/S par le code invité amène toujours la VM à quitter, permettant l'émulation de périphérique.

La plus grosse différence entre VT-x et AMD-V est qu'AMD-V fournit en environnement de virtualisation plus complet. VT-x exige que le code non-racine VMX s'exécute en mode pagination activée, ce qui rejette la virtualisation matérielle de logiciels dont le code est en mode réel et en mode protégé non paginé. Cela n'inclut en général que les firmwares et les chargeurs d'OS, néanmoins cela complique l'implémentation d'un hyperviseur avec VT-x. AMD-V n'a pas cette restriction.

Bien entendu, la virtualisation matérielle n'est pas parfaite. Par rapport à la virtualisation logicielle, la charge des fins des VMs est relativement élevée. Cela pose des problèmes aux périphériques dont l'émulation requiert un grand nombre de blocages (trass). Par exemple, avec le périphérique VGA en mode 16 couleurs, non seulement tous les accès au port en E/S, mais aussi tous les accès à la mémoire tampon (framebuffer) doivent être bloqués.

10.6 Vagination nestée et VPIIDs

En plus de la virtualisation matérielle “brute”, votre processeur peut supporter aussi des techniques sophistiquées supplémentaires :²

- Une fonctionnalité récente, qui s'appelle la “**pagination nestée**” implémente la gestion de la mémoire dans le matériel, ce qui peut beaucoup accélérer la virtualisation matérielle puisque ces tâches n'ont plus besoin d'être accomplies par le logiciel de virtualisation.

Avec la pagination nested, le matériel fournit un autre niveau d'indirection en passant du linéaire aux adresses physiques. Les tables de page fonctionnent comme avant mais les adresses linéaires sont désormais d'abord traduites en adresses physiques de “l'invité” et pas directement en adresses physiques. Il existe maintenant un nouveau jeu de registres de pagination sous le mécanisme depagination traditionnel et qui traduit les adresses physiques invitées en adresses physiques de l'hôte, qui sont utilisées pour accéder à la mémoire.

La pagination nested élimine la charge causée par les fins de VM et les accès aux tables de pages. Par définition, avec les tables de pages nested, l'invité peut gérer la pagination sans que l'hyperviseur n'intervienne. La pagination nestée améliore ainsi substantiellement les performances de virtualisation.

Sur les processeurs AMD, la pagination nested est disponible depuis l'architecture Barcelona (K10) – on l'appelle maintenant la “rapid virtualization indexing” (RVI). Intel a ajouté le support de la pagination nested, qu'ils appellent la “extended page tables” (EPT), à leurs processeurs Core i7 (Nehalem).

Si la pagination nested est activée, l'hyperviseur de VirtualBox peut également utiliser **large pages**, pour réduire l'utilisation du TLB et la charge. Cela peut provoquer une amélioration

²VirtualBox 2.0 a ajouté le support de la pagination nestée d'AMD ; le support de l'EPT et des VPIIDs d'Intel a été ajouté à la version 2.1.

jusqu'à 5% des performances. Pour activer cette fonctionnalité pour une VM, vous avez besoin d'utiliser la commande `VBoxManage modifyvm --largepages` ; voir chapitre 8.8, [VBoxManage modifyvm](#), page 130.

- Sur les processeurs Intel, une autre fonction matérielle, qui s'appelle “**Virtual Processor Identifiers**” (VPIDs), peut beaucoup accélérer le changement de contexte en réduisant le besoin de flasher beaucoup les Translation Lookaside Buffers (TLBs) du processeur.

Pour activer ces fonctions pour une VM, vous devez utiliser les commandes `VBoxManage modifyvm --vtxvpid` and `--largepages` ; voir chapitre 8.8, [VBoxManage modifyvm](#), page 130.

11 Interfaces de programmation de VirtualBox

VirtualBox est fournie avec un support complet pour les développeurs tiers. Ce qu'on appelle “l'API principale” de VirtualBox implémente toutes les fonctionnalités du moteur de virtualisation. Elle est complètement documentée et disponible pour n'importe qui souhaitant contrôler VirtualBox en programmant.

L'API principale est disponible pour des clients C++ via COM (sur les hôtes Windows) ou XPCOM (sur les autres hôtes). Des passerelles existent aussi avec SOAP, Java et Python.

Toutes les informations sur la programmation (documentation, informations de référence, en-têtes et autres fichiers d'interface ainsi que des modèles) ont été mises dans un **Software Development Kit (SDK)**, distinct disponible en téléchargement sur <http://www.virtualbox.org>. En particulier, le SDK est fourni avec un “PGuide de programmation et de Référence” au format PDF contenant entre autres des informations qui se trouvaient jadis dans ce chapitre du manuel de l'utilisateur.

12 Dépannage

Ce chapitre apporte des réponses à des questions fréquemment posées. Afin d'améliorer votre expérience utilisateur avec VirtualBox, il est recommandé de lire cette section pour en apprendre plus sur les soucis classiques et pour avoir les recommandations sur la manière d'utiliser le produit.

12.1 Procédures et outils

12.1.1 Catégoriser et isoler des problèmes

Le plus souvent, un invité virtualisé se comporte comme un système physique. Une machine virtuelle rencontrera les mêmes problèmes que le ferait une machine physique. Si, par exemple, vous perdez la connectivité à Internet à cause de problèmes extérieurs, les machines virtuelles seront touchées exactement comme celles physiques.

Si vous rencontrez un problème vraiment lié à VirtualBox, celui-ci aide à le catégoriser et à l'isoler. Voici quelques-unes questions auxquelles vous devriez répondre avant de signaler un problème :

1. Le problème est-il spécifique à un OS invité en particulier ? À une version de l'OS invité ? Surtout avec les problèmes liés aux invités Linux, le problème peut être spécifique à une distribution et à une version de Linux.
2. Le problème est-il spécifique à un OS hôte en particulier ? Les problèmes ne sont généralement pas spécifiques à un OS hôte (car la plupart de la base du code de VirtualBox est partagée par toutes les plateformes supportées), mais, surtout en matière de réseau et de support USB, il existe d'importantes différences entre les plateformes hôtes. Certains problèmes liés à la GUI sont aussi spécifiques à l'hôte.
3. Le problème est-il spécifique à un matériel hôte particulier ? Cette catégorie de problèmes est généralement liée au processeur de l'hôte. D1 fait de différences importantes entre VT-x et AMD-V, des problèmes peuvent être spécifiques à l'une ou l'autre technologie. Le modèle exact du processeur peut également marquer une différence (même pour la virtualisation logicielle) car différents processeurs supportent différentes fonctions, ce qui peut toucher certains aspects du fonctionnement du processeur invité.
4. Le problème est-il spécifique à un mode de virtualisation en particulier ? Certains problèmes peuvent n'arriver qu'en mode virtualisation logicielle, d'autres peuvent être spécifiques à la virtualisation matérielle.
5. Le problème est-il spécifique au SMP de l'invité ? À savoir, est-il lié au nombre de processeurs virtuels (VCPUs) de l'invité ? L'utilisation de plus d'un processeur touche de façon significative le fonctionnement interne d'un OS invité.
6. Le problème est-il spécifique aux suppléments invité ? Dans certains cas, c'est écrit (par exemple un problème de dossiers partagés), dans d'autres, cela peut être moins évident (par exemple, des problèmes d'affichage). Si le problème est spécifique aux suppléments invité, est-il spécifique à une version en particulier des suppléments invité ?

7. Le problème est-il spécifique à un environnement particulier ? Certains problèmes sont liés à un environnement et externes à la VM ; cela implique en général un paramétrage du réseau. Certaines configurations de serveurs externes tels que DHCP ou PXE peuvent poser des problèmes qui ne surviennent pas avec d'autres serveurs identiques.
8. Le problème est-il une régression ? Le fait de savoir qu'un problème est une régression facilite beaucoup en général la recherche d'une solution. Dans ce cas, il est crucial de connaître la version concernée et celle qui ne l'est pas.

12.1.2 Recueillir des informations de débogage

Pour déterminer un problème, il est souvent important de recueillir des informations de débogage que l'assistance de VirtualBox peut analyser. Cette section contient des informations sur le type d'informations que vous pouvez obtenir.

À chaque fois que VirtualBox démarre une VM, ce qu'on appelle un “**release log file**” (fichier journal) est créé, contenant beaucoup d'informations sur la configuration de la VM et les événements lors de son exécution. Le fichier journal s'appelle `VBox.log` et se trouve dans le dossier du fichier journal de la VM. Il s'agira en général d'un répertoire comme celui-ci :

```
$HOME/VirtualBox VMs/{machinenename}/Logs
```

Au démarrage d'une VM, le fichier de configuration de la dernière exécution sera renommé en `.1`, jusqu'à `.3`. Parfois, quand il y a un problème, il est utile de jeter un œil dans le journal. Quand vous demandez de l'aide sur VirtualBox, le fait de fournir le fichier journal correspondant est obligatoire.

Par commodité, pour chaque machine virtuelle, la fenêtre principale de VirtualBox peut afficher ces journaux dans une fenêtre. Pour y accéder, sélectionnez une machine virtuelle dans la liste à gauche et sélectionnez “Afficher les journaux...” dans la fenêtre “Machine”.

Le fichier journal (`VBox.log`) contient une gamme d'informations de diagnostic telles que le type et la version d'OS hôte, la version de VirtualBox et l'architecture (32 ou 64 bits), un aperçu complet de la configuration de l'invité (CFGM), des informations détaillées sur le type et les fonctions supportées par le processeur, si la virtualisation matérielle est activée, des informations sur le réglage VT-x/AMD-V, l'état des transitions (création, exécution, en pause, éteint, etc.), les messages du BIOS invité, les messages des suppléments invité, les entrées du journal spécifiques aux périphériques, à la fin de l'exécution, l'état final de l'invité et des statistiques consolidées.

En cas de plantage, il est très important de recueillir **les sorties du plantage**. Ceci est vrai tant pour les plantages de l'hôte que pour ceux de l'invité. Pour des informations sur l'activation de plus de messages sur les systèmes Linux, Solaris et OS X, reportez-vous à l'article sur les messages du cœur sur le site Internet de VirtualBox.¹

Vous pouvez également utiliser `VBoxManage debugvm` pour créer un journal de toute une machine virtuelle ; voir chapitre [8.32, `VBoxManage debugvm`, page 156](#).

Pour des problèmes liés au réseau, il est souvent utile de récupérer une trace du trafic réseau. Si le trafic est acheminé par un adaptateur de l'hôte, il est possible d'utiliser Wireshark ou un outil similaire pour y récupérer le trafic. Cependant, cela inclut aussi souvent beaucoup de trafic indépendant de la VM.

VirtualBox offre la possibilité de récupérer seulement le trafic réseau de l'adaptateur réseau d'une VM spécifique. Reportez-vous à l'article sur le trafic réseau sur le site Internet de VirtualBox² pour des informations sur l'activation de cette récupération. Les fichiers de trace créés par VirtualBox sont au format `.pcap` et peuvent être facilement analysés avec Wireshark.

¹http://www.virtualbox.org/wiki/Core_dump.

²http://www.virtualbox.org/wiki/Network_tips.

12.1.3 Le débogueur de VM intégré

VirtualBox inclut un débogueur de VM intégré, qui peut servir aux utilisateurs avancés. Ce débogueur permet d'examiner et, dans une certaine mesure, de contrôler l'état de la VM.

Avertissement: L'utilisation du débogueur de VM est à vos risques et périls. Il n'existe pas d'assistance autour, la documentation suivante a été rendue disponible uniquement pour les utilisateurs avancés ayant un degré de familiarité très élevé du jeu d'instructions d'une machine x86/AMD64, ainsi que des connaissances détaillées de l'architecture PC. Une certaine familiarité avec les côtés internes de l'OS invité concerné peut aussi aider.

Le débogueur de VM est disponible dans toutes les versions ordinaires de production de VirtualBox, mais il est désactivé par défaut car l'utilisateur moyen l'utilisera très peu. Il existe deux manières d'accéder au débogueur :

- Une fenêtre de console du débogueur affichée à côté de la VM
- Via le protocole `telnet` sur le port 5000

Vous pouvez activer le débogueur de trois façons :

- Démarrer la VM directement en utilisant `VirtualBox --startvm`, avec, en plus, l'argument `--dbg`, `--debug`, ou `--debug-command-line`. Voir l'aide sur l'utilisation de VirtualBox pour des détails.
- Définir la variable d'environnement `VBOX_GUI_DBG_ENABLED` ou `VBOX_GUI_DBG_AUTO_SHOW` avec `true` avant de lancer le processus de VirtualBox. Le réglage des variables (seule leur présence est vérifiée) est effectif, même quand le premier processus de VirtualBox est la fenêtre du sélecteur de VM. Les VMs qui se lancent ensuite à partir du sélecteur auront un débogueur actif.
- Définir la donnée supplémentaire `GUI/Dbg/Enabled` sur `true` avant de lancer la VM. Vous pouvez la régler de façon globale ou sur une base individuelle à chaque VM.

Un nouveau menu 'Débogage' sera ajoutée à l'application VirtualBox. Ce menu permet à l'utilisateur d'ouvrir la console du débogueur.

La syntaxe des commandes du débogueur de VM est grossièrement sur le même modèle que les débogueurs de Microsoft et d'IBM, utilisés sur DOS, OS/2 et Windows. Les utilisateurs familiers de symdeb, CodeView, ou du débogueur du noyau de OS/2 trouveront le débogueur de VM de VirtualBox classique.

La commande la plus importante est `help`. Cela affichera un message d'aide à l'utilisation rapide de toutes les commandes du débogueur. L'ensemble des commandes supporté par le débogueur de VM change souvent et la commande `help` est toujours à jour.

Voici un résumé rapide des commandes souvent utilisées :

- `stop` – arrête l'exécution de la VM et active le mono-session (single stepping)
- `g` – continue l'exécution de la VM
- `t` – passe en mono-session (single step) une instruction
- `rg/rh/r` – affiche les registres actuels de l'invité/hyperviseur
- `kg/kh/k` – affiche la pile d'appel actuelle de l'invité/hyperviseur actuel
- `da/db/dw/dd/dq` – affiche le contenu de la mémoire sous forme d'as ASCII/octets/mots/dwords/qwords

- **u** – désassemble la mémoire
- **dg** – affiche le GDT de l'invité
- **di** – affiche le IDT de l'invité
- **dl** – affiche le LDT de l'invité
- **dt** – affiche le TSS de l'invité
- **dp*** – affiche les structures des tables de pages de l'invité
- **bp/br** – définit un point de rupture normal/recompilateur
- **bl** – liste les points de rupture
- **bc** – vide les points de rupture
- **writecore** – écrit sur le disque un fichier cœur de VM, rapprochez-vous au chapitre [12.1.4, Format du cœur d'une VM](#), page [212](#)

Voir le **help** intégré pour d'autres commandes disponibles.

Le débogueur de VM supporte le débogage symbolique de base, même si les symboles du code invité ne sont pas souvent disponibles. Pour les invités Solaris, la commande **detect** détermine automatiquement la version de l'OS invité et localise les symboles du noyau dans la mémoire de l'invité. Le débogage symbolique est alors disponible. Pour les invités Linux, les commandes **detect** déterminent également la version de l'OS invité, mais il n'y a pas de symboles dans la mémoire de l'invité. Les symboles du noyau sont disponibles dans le fichier `/proc/kallsyms` des invités Linux. Vous devez copier ce fichier dans l'hôte, en utilisant par exemple `scp`. La commande **loadmap** du débogueur peut être utilisée pour rendre les informations de symbole disponibles pour le débogueur de VM. Remarquez que le fichier `kallsyms` contient les symboles des modules actuellement chargés ; si la configuration de l'invité change, les symboles changeront aussi et doivent être mis à jour.

Pour tous les invités, une façon simple de vérifier que les bons symboles sont chargés est la commande **k**. L'invité est en principe occupé et il devrait être vidé des informations symboliques que la boucle active du système d'exploitation invité exécute.

Un autre groupe de commandes du débogueur est celui **info**. L'exécution `d info help` fournit ces informations d'utilisation complètes. Les commandes d'informations fournissent des données ad-hoc pertinentes sur divers périphériques émulés et sur les aspects de la VMM. Il n'y a pas de lignes directrices générales sur l'utilisation des commandes **info**, la bonne commande à utiliser dépend beaucoup du problème à trouver. Certaines commandes **info** sont :

- **cfgm** – affiche une branche de l'arborescence de la configuration
- **cpuid** – affiche les sorties du CPUID de l'invité
- **ioport** – affiche les E/S des plages de ports enregistrées
- **mmio** – affiche les plages MMIO enregistrées
- **mode** – affiche le mode de pagination actuel
- **pit** – affiche l'état i8254 PIT
- **pic** – affiche l'état i8259A PIC
- **ohci/ehci** – affiche un sous-ensemble de l'état du contrôleur USB OHCI/EHCI
- **pcnet0** – affiche l'état PCnet

- **vgatext** – affiche le contenu du tampon (framebuffer) VGA formaté en mode texte standard
- **timers** – affiche toutes les horloges de la VM

La sortie des commandes **info** exige généralement une connaissance approfondie du périphérique émulé et/ou des aspects internes de VirtualBox VMM. Cependant, quand on les utilise correctement, les informations fournies peuvent avoir une valeur inestimable.

12.1.4 Format du cœur d'une VM

VirtualBox utilise le format ELF 64 bits pour les fichiers cœur de la VM créés par **VBoxManage debugvm** ; voir chapitre [8.32, *VBoxManage debugvm*, page 156](#). Les fichiers cœur d'une VM contiennent les messages de la mémoire et du processeur de la VM et ils peuvent être utiles pour déboguer votre OS invité. Vous pouvez savoir les spécifications du format objet ELF 64 bits ici : <http://downloads.openwatcom.org/ftp/devel/docs/elf-64-gen.pdf>.

La présentation grosso modo du format du cœur de la VM est celle-ci :

```
[ ELF 64 Header]
[ Program Header, type PT_NOTE ]
-> offset to COREDESCRIPTOR
[ Program Header, type PT_LOAD ] - un par plage de mémoire physique contiguë
-> Memory offset of range
-> File offset
[ Note Header, type NT_VBOXCORE ]
[ COREDESCRIPTOR ]
-> Magic
-> Version du fichier cœur de la VM
-> Version de VBox
-> Nombre de vprocesseurs etc.
[ Note Header, type NT_VBOXCPU ] - one pour chaque vprocesseur
[ vCPU 1 Note Header ]
[ CPUMCTX - vCPU 1 dump ]
[ Remarques + données supplémentaires ] - Non utilisées aujourd'hui
[ Memory dump ]
```

Les descripteurs de mémoire contiennent les adresses physiques de la mémoire liées à l'invité et pas les adresses virtuelles. Les régions de la mémoire telles que les régions MMIO ne sont pas incluses dans le fichier cœur.

Vous pouvez trouver les structures de données et les définitions pertinentes dans les sources de VirtualBox sous les fichiers en-têtes suivants : `include/VBox/dbgfcorefmt.h`, `include/VBox/cpumctx.h` et `src/VBox/Runtime/include/internal/ldrELFCommon.h`.

Vous pouvez examiner le fichier cœur de la VM en utilisant `elfdump` et `GNU readelf` ou d'autres outils similaires.

12.2 Général

12.2.1 L'invité affiche des erreurs IDE/SATA pour les images basées sur un fichier sur un système de fichiers hôte lent

De temps en temps, certains systèmes de fichiers hôte offrent des performances d'écriture très faibles et, par conséquent, créent des timeout sur les commandes IDE/SATA de l'invité. C'est un comportement normal et cela ne devrait pas provoquer de vrais problèmes, car l'invité devrait répéter des commandes qui ont dépassé le timeout. Cependant, certains invités (comme certaines versions de Linux) ont de gros problèmes si l'écriture dans un fichier image met 15 secondes. Or, certains systèmes de fichiers nécessitent plus d'une minute pour effectuer une seule écriture, si le cache de l'hôte contient beaucoup de données à écrire.

12 Dépannage

Le symptôme de ce problème est que l'invité ne peut plus accéder à ses fichiers lors de grosses écritures, ce qui aboutit en général à un accroc immédiat de l'invité.

Pour contourner ce problème (la vraie correction est d'utiliser un système de fichier plus rapide qui n'excède pas de telles performances d'écriture inacceptables), il est possible de flasher le fichier image après qu'une certaine quantité de données a été écrite. Cet intervalle est en principe infini mais vous pouvez le configurer individuellement pour chaque disque d'une VM.

Pour des disques IDE, utilisez la commande suivante :

```
VBoxManage setextradata "nom VM"  
    "VBoxInternal/Devices/piix3ide/0/LUN#[x]/Config/FlushInterval" [b]
```

Pour des disques SATA, utilisez la commande suivante :

```
VBoxManage setextradata "nom VM"  
    "VBoxInternal/Devices/ahci/0/LUN#[x]/Config/FlushInterval" [b]
```

La valeur [x] qui sélectionne le disque pour l'IDE est 0 pour le périphérique maître du premier canal, 1 pour périphérique esclave du premier canal, 2 pour le périphérique maître du deuxième canal, ou 3 pour le périphérique esclave du deuxième canal. Pour SATA, utilisez des valeurs entre 0 et 29. Seuls les disques supportent cette option de configuration ; vous ne devez pas la définir pour des lecteurs CD/DVD.

L'unité d'intervalle [b] est le nombre d'octets écrits depuis le dernier flush. Sa valeur doit être sélectionnée de sorte que les longs départs d'écriture occasionnels ne se produisent pas. Comme la bonne valeur d'intervalle de flush dépend des performances de l'hôte et du système de fichiers hôte, savoir la valeur optimum qui fait disparaître le problème nécessite d'expérimenter. Des valeurs entre 1000000 et 10000000 (1 to 10 mégaoctets) sont un bon point de départ. La diminution de l'intervalle réduit la probabilité du problème et les performances d'écriture de l'invité. Le test des valeurs faibles inutilement sera coûteux en performances sans avantages. Un intervalle de 1 fera un flush toutes les opérations d'écriture et cela devrait résoudre le problème dans tous les cas, mais cela est très coûteux en performances d'écriture.

Fournir la valeur 0 à [b] revient à un intervalle de flush infini ce qui désactive de fait ce contournement. La suppression de la donnée supplémentaire en ne spécifiant aucune valeur pour [b] aboutit au même effet.

12.2.2 Réponse aux requêtes de flush IDE/SATA de l'invité

Si vous le souhaitez, les images de disque virtuelles peuvent être flashées quand l'invité lance une commande IDE FLUSH CACHE. Normalement ces requêtes sont ignorées pour des performances améliorées. Les paramètres ci-dessous sont acceptés uniquement pour les lecteurs de disque. Elles ne doivent pas être définies pour des lecteurs DVD.

Pour activer le flash des disques IDE, lancez la commande suivante :

```
VBoxManage setextradata "nom VM" "VBoxInternal/Devices/piix3ide/0/LUN#[x]/Config/IgnoreFlush" 0
```

La valeur [x] qui sélectionne le disque pour l'IDE est 0 pour le périphérique maître du premier canal, 1 pour périphérique esclave du premier canal, 2 pour le périphérique maître du deuxième canal, ou 3 pour le périphérique esclave du deuxième canal.

Pour activer le flash pour des disques SATA, lancez la commande suivante :

```
VBoxManage setextradata "nom VM" "VBoxInternal/Devices/ahci/0/LUN#[x]/Config/IgnoreFlush" 0
```

La valeur [x] qui sélectionne le disque peut être une valeur entre 0 et 29.

Remarquez que cela ne concerne pas les flash effectués selon la configuration décrite au chapitre 12.2.1, *L'invité affiche des erreurs IDE/SATA pour les images basées sur un fichier sur un système de fichiers hôte lent*, page 212. La restauration des paramètres par défaut d'ignorance des commandes est possible en paramétrant la valeur sur 1 ou en supprimant la clé.

12.2.3 Faibles performances dues à la gestion d'énergie de l'hôte

Sur certaines plateformes matériel et sur certains systèmes d'exploitation, les performances de virtualisation sont touchées de manière négative par la gestion d'énergie du processeur de l'hôte. Les symptômes peuvent être un changement de son dans l'invité ou un comportement erratique de l'horloge de l'invité.

Certains problèmes peuvent venir de bogues d'un firmware et/ou du système d'exploitation hôte. Donc, la mise à jour du firmware et l'application de correctifs au système d'exploitation est recommandée.

Pour des performances de virtualisation optimales, le support de l'état d'énergie C1E dans le BIOS du système devrait être activé si ce paramètre est disponible (tous les systèmes ne supportent pas l'état d'énergie C1E). Sur les systèmes Intel, le paramètre Intel C State devrait être désactivé. La désactivation d'autres paramètres de gestion d'énergie peut aussi améliorer les performances. Toutefois, vous devez toujours faire un bilan performance consommation d'énergie.

12.2.4 GUI : l'option d'accélération graphique est grisée

Pour utiliser l'accélération graphique 2D dans VirtualBox, la carte graphique de votre hôte devrait supporter certaines extensions d'OpenGL. Au démarrage, VirtualBox vérifie ces extensions et, si le test échoue, cette option est grisée silencieusement..

Pour savoir la raison pour laquelle il a échoué, vous pouvez exécuter à la main la commande suivante :

```
VBoxTestOGL --log "log_file_name" --test 2D
```

Elle listera les extensions OpenGL nécessaires une par une et elle vous montrera celles où le test a échoué. Cela signifie en général que vous exécutez un pilote OpenGL obsolète ou mal configuré sur votre hôte. Cela peut aussi signifier que le chipset graphique manque d'une fonctionnalité requise.

12.3 Invités Windows

12.3.1 Écrans bleus Windows après avoir changé la configuration d'une VM

La modification de certains paramètres d'une machine virtuelle peut faire échouer des invités Windows au démarrage, avec un écran bleu. Cela peut se produire si vous changez les paramètres d'une VM après avoir installé Windows ou si vous copiez une image de disque avec un Windows installé sur une VM nouvellement créée dont les paramètres diffèrent de la machine d'origine.

Cela s'applique en particulier aux paramètres suivants :

- Vous ne devriez jamais modifier les paramètres ACPI et APIC E/S après avoir installé Windows. Selon la présence de ces fonctions matérielles, le programme d'installation de Windows choisit des versions spéciales du noyau et des pilotes de périphérique et il n'arrivera pas à démarrer si on supprime ces fonctionnalités. (Leur activation pour une VM Windows installé sans elles ne présente aucun risque. Par contre, Windows n'utilisera pas ces fonctions dans ce cas.)
- La modification des contrôleurs de stockage aboutira à des échecs au démarrage. Cela pourrait aussi s'appliquer si vous copiez une image de disque d'une ancienne version de VirtualBox sur une machine virtuelle créée avec une version de VirtualBox plus récente ; le sous-type de contrôleur IDE est passé de PIIX3 à PIIX4 avec VirtualBox 2.2. Assurez-vous que ces paramètres sont identiques.

12.3.2 Écran bleu sur Windows 0x101 si SMP est activé (IPI timeout)

Si une VM est configurée pour avoir plus d'un processeur (multiprocesseurs symétrique, SMP), certaines configurations d'invités Windows plantent avec un message d'erreur 0x101 indiquant une interruption du timeout de l'inter-processeur (IPIs, Interprocessor Interrupts). Ces interruptions synchronisent la gestion de mémoire entre les processeurs.

Selon Microsoft, cela vient d'une race condition (condition conflictuelle) dans Windows. Un correctif existe.³ Si cela n'aide pas, merci de réduire le nombre de processeurs virtuels à 1.

12.3.3 Échecs d'installation de Windows 2000

En installant des invités Windows 2000, vous pourriez rencontrer un des problèmes suivants :

- L'installation redémarre, en général lors de l'enregistrement d'un composant.
- L'installation remplit tout le disque dur par des fichiers journaux vides.
- L'installation se plaint d'un échec lors de l'installation de msgina.dll.

Ces problèmes viennent tous d'un bogue du pilote de disque dur de Windows 2000. Après avoir sollicité une requête du disque dur, il arrive un conflit de condition (race condition) dans le code du pilote Windows, qui conduit à une corruption si l'opération se termine trop vite, donc si l'interruption matérielle du contrôleur IDE survient trop tôt. Avec du matériel physique, il existe un délai garanti dans la plupart des systèmes, donc le problème est généralement caché (il devrait être cependant possible de le reproduire aussi sur du matériel physique). Dans un environnement virtuel, l'opération peut se faire immédiatement (surtout sur des systèmes très rapides) avec plusieurs processeurs) et l'interruption est signalée plus tôt que sur un système physique. La solution consiste à introduire un délai artificiel avant d'envoyer de telles interruptions. Vous pouvez configurer ce délai pour une VM avec la commande suivante :

```
VBoxManage setextradata "nom VM" "VBoxInternal/Devices/piix3ide/0/Config/IRQDelay" 1
```

Ceci définit le délai sur une milliseconde. Si cela n'aide pas, passez-le à une valeur entre 1 et 5 millisecondes. Merci de remarquer que cela ralentit les performances du disque. Après l'installation, vous devriez pouvoir supprimer la clé (ou la passer à 0).

12.3.4 Comment garder les informations d'un écran bleu des invités Windows

Quand les invités Windows connaissent un plantage du noyau, ils affichent l'horrible écran bleu. Selon la façon dont est configuré Windows, les informations demeureront à l'écran jusqu'à ce que la machine ne redémarre ou ils redémarrent automatiquement. Pendant l'installation, Windows est généralement configuré pour redémarrer automatiquement. Avec le redémarrage automatique, il n'y a aucune chance d'enregistrer les informations d'un écran bleu, alors qu'elles pourraient être importantes pour déterminer le problème.

VirtualBox offre une méthode d'arrêt de l'invité quand il veut redémarrer. Pour activer cette fonction, exécutez la commande suivante :

```
VBoxManage setextradata "nom VM" "VBoxInternal/PDM/HaltOnReset" 1
```

³Voir <http://support.microsoft.com/kb/955076>.

12.3.5 Pas de réseau dans les invités Windows Vista

Avec Windows Vista, Microsoft a abandonné le support de la carte AMD PCNet utilisée par VirtualBox comme carte réseau virtuelle par défaut avant la version 1.6.0. Pour les invités Windows Vista, VirtualBox utilise maintenant par défaut une carte Intel E1000.

Si, pour une raison quelconque, vous voulez toujours utiliser la carte AMD, vous devez télécharger le pilote de PCNet sur le site Internet d'AMD (disponible seulement pour Windows 32 bits). Vous pouvez le transférer dans la machine virtuelle en utilisant un dossier partagé (voir chapitre [4.3, Dossiers partagés](#), page [75](#)).

12.3.6 Les invités Windows peuvent provoquer une forte charge du processeur

Plusieurs applications en arrière-plan des invités Windows, en particulier les anti-virus, sont connues pour augmenter considérablement la charge du processeur même si l'invité semble être inactif. Nous vous recommandons de désactiver les anti-virus des invités virtualisés si possible.

12.3.7 Temps d'accès élevés aux dossiers partagés

Les performances d'accès aux dossiers partagés depuis un invité Windows pourraient diminuer du fait des délais de résolution du service de domaine des dossiers partagés de VirtualBox. Pour corriger ces délais, ajoutez les entrées suivante au fichier `\windows\system32\drivers\etc\lmhosts` de l'invité Windows :

```
255.255.255.255      VBOXSVR #PRE
255.255.255.255      VBOXSRV #PRE
```

Après ce changement, il faut redémarrer l'invité.

12.3.8 La tablette USB coordonne mal dans les invités Windows 98

Si une VM Windows 98 est configurée pour utiliser la tablette USB émulée (périphérique de pointage absolu), il se peut que la traduction de la coordination soit incorrecte et que le pointeur soit restreint au quart en haut à gauche de l'écran de l'invité.

Les pilotes HID (Human Interface Device) USB de Windows 98 sont très vieux et ils ne gèrent pas les tablettes de la même manière que les systèmes d'exploitation récents (Windows 2000 et supérieur, Mac OS X, Solaris). Pour contourner le problème, exécutez la commande suivante :

```
VBoxManage setextradata "nom VM" "VBoxInternal/USB/HidMouse/0/Config/CoordShift" 0
```

Pour restaurer le comportement par défaut, supprimez la clé ou réglez sa valeur à 1.

12.3.9 Les invités Windows sont retirés du domaine Active Directory après la restauration d'un instantané

Si un invité Windows est membre d'un domaine Active Directory et que vous utilisez la fonction des instantanés de VirtualBox, pourraient se produire des pertes de cet état après la restauration d'un ancien instantané.

Ceci vient du changement automatique de mot de passe de la machine opéré régulièrement par Windows pour des raisons de sécurité. Vous pouvez désactiver cette fonction en suivant les instructions de <http://support.microsoft.com/kb/154501> cet article de Microsoft.

12.3.10 Restauration de d3d8.dll et de d3d9.dll

Les suppléments invité de VirtualBox pour Windows et inférieurs à la 4.1.8 ne sauvegardaient pas les fichiers système d'origine d3d8.dll et d3d9.dll lors de l'installation du support expérimental de Direct3D. Ce processus remplace ces deux fichiers système par des fichiers des suppléments invité de VirtualBox gérables correctement par les appels de Direct3D. Si ce problème a été corrigé avec VirtualBox 4.1.8, il n'y a aucun moyen de faire réparer ces fichiers par l'installateur des suppléments invité.

La corruption de ces fichiers n'a pas d'implications si l'accélération 3D est activée et si le support de base de Direct3D est installé, à savoir sans WDDM (sur Windows Vista ou supérieur) ou sur les anciens systèmes Windows comme Windows XP. Avec le support Direct3D de base, toutes les applications Direct3D 8.0 et Direct3D 9.0 utiliseront directement les fichiers Direct3D de VirtualBox et fonctionneront ainsi comme prévu.

Par contre, pour le support WDDM Direct3D, les fichiers d3d8.dll et d3d9.dll inclus d'origine sont nécessaires pour lancer des applications Direct3D 8.0 et Direct3D 9.0. Il résulte de la corruption des fichiers système ci-dessus que ces applications ne fonctionneront plus. Voir ci-dessous pour une guide pas à pas sur la restauration des fichiers systèmes d'origine d3d8.dll et d3d9.dll si l'installateur des suppléments invité de VirtualBox a averti que ces fichiers étaient incorrects ou en cas de problème en exécutant les applications Direct3D.

Note: À partir de Windows 7 le bureau 3D (aka Aero) utilise DirectX 10 pour être affiché afin que les fichiers d3d8.dll et d3d9.dll corrompus n'aient aucun effet sur la session en cours.

C'est pourquoi la détection d'une telle corruption de fichier n'est pas considérée comme fatale pour l'installation basique de Direct3D sur tous les invités Windows supportés et pour une installation de WDDM Direct3D sur les invités Windows 7 et supérieur.

Extraire d3d8 et d3d9.dll du CD d'installation de Windows XP :

1. Téléchargez et installez la dernière version de 7-Zip File Manager <http://www.7-zip.org>
2. Parcourez le CD d'installation, par exemple E:\i386 (ou AMD64 pour la version 64 bits)
3. Localisez le fichier d3d8.dll et d3d9.dll, cliquez deux fois dessus et extrayez d3d8.dll et d3d9.dll
4. Redémarrez Windows en mode sans échec
5. Copiez d3d8.dll et d3d9.dll extraits dans C:\Windows\system32 and C:\Windows\system32\dllcache
6. Redémarrez

Extraction de d3d8 et de d3d9.dll du pack service de Windows XP

1. 1, 3-6 Identiques au CD d'installation
2. Utilisez 'Ouvrir avec' pour ouvrir WindowsXP-KB936929-SP3-x86.exe en tant qu'archive et parcourez le répertoire i386.

Extraction de d3d8 et de d3d9.dll du CD d'installation de Vista/Windows7 ou des images du pack Service

1. Téléchargez et installez la dernière version de 7-Zip File Manager <http://www.7-zip.org>
2. Parcourez le CD d'installation, par exemple E:\sources

3. Localisez le fichier install.wim et cliquez deux fois dessus. Après l'ouverture du fichier par 7-Zip, vous verrez un certain nombre de dossiers. Chaque sous-dossier numéroté représente une version différente de Windows (Starter, Home Basic, and ainsi de suite)
4. Après être entré dans les dossiers numérotés adéquats, parcourez le répertoire Windows\System32 (ou C:\Windows\SysWOW64 pour la version 64 bits) et localisez d3d8.dll et d3d9.dll puis extrayez
5. Copiez d3d8.dll et d3d9.dll extraits dans C:\Windows\system32 ou C:\Windows\SysWOW64 (les fichiers de system32 devraient aller dans system32, ceux de SysWOW64 dans SysWOW64)
6. Redémarrez

12.4 Invités Linux et X11

12.4.1 Les invités Linux peuvent entraîner une forte charge du processeur

Certains invités Linux peuvent entraîner une forte charge du processeur même si le système invité semble inactif. Cela peut venir d'une fréquence horloge élevée du noyau invité. Certaines distributions Linux, par exemple Fedora, incluent un noyau Linux configuré pour une fréquence d'horloge de **1000Hz**. Nous vous recommandons de recompiler le noyau invité et de sélectionner une fréquence d'horloge de 100Hz.

Les noyaux Linux inclus avec Linux Red Hat Enterprise (RHEL) entre la version 4.7 et 5.1 ainsi que les noyaux des distributions Linux associées (par exemple, CentOS et Oracle Enterprise Linux) supportent un paramètre *divider=N* du noyau. D'où le fait que de tels noyaux supportent une fréquence d'horloge plus faible sans recompilation. Nous vous suggérons d'ajouter le paramètre *divider=10* du noyau pour sélectionner une fréquence de l'horloge du noyau invité de 100Hz.

12.4.2 Processeurs AMD Barcelona

La plupart des invités basés sur Linux échoueront avec l'AMD Phenoms ou Barcelona-level Opterons du fait d'un bogue dans le noyau Linux. Activez l'APIC E/S pour contourner le problème (voir chapitre [3.4, Paramètres système](#), page 52).

12.4.3 Versions bugguées du noyau Linux Linux 2.6

Les bogues suivants des noyaux Linux les empêchent de les exécuter correctement dans VirtualBox, ce qui fait planter la VM au démarrage :

- La version du noyau Linux 2.6.18 (et certaines versions 2.6.17) ont introduit un conflit de condition (race condition) qui peut provoquer un plantage au démarrage dans VirtualBox. Merci d'utiliser une version du noyau 2.6.19 ou supérieur.
- Avec la virtualisation matérielle et l'APIC E/S activé, les noyaux inférieurs au 2.6.24-rc6 peuvent planter au démarrage avec le message suivant :

```
Kernel panic - not syncing: IO-APIC + timer doesn't work! Boot with
apic=debug and send a report. Then try booting with the 'noapic' option
```

Si vous voyez ce message, soit désactivez la virtualisation matérielle, soit l'APIC E/S (voir chapitre [3.4, Paramètres système](#), page 52), ou mettez à jour l'invité vers un noyau plus récent.⁴

⁴Voir <http://www.mail-archive.com/git-commits-head@vger.kernel.org/msg30813.html> pour des détails sur le correctif du noyau.

12.4.4 Presse-papier partagé, redimensionnement automatique et bureau transparent dans les invités X11

Les services du bureau invité dans les invités exécutant le système X11 window (Solaris, Linux et autres) sont fournis par un service invité qui s'appelle `VBoxClient`, qui fonctionne sous l'ID de l'utilisateur qui démarre la session du bureau et qui est démarré automatiquement en utilisant les lignes de commande suivantes

```
VBoxClient --clipboard
VBoxClient --display
VBoxClient --seamless
```

quand votre session utilisateur X11 est lancée si vous utilisez un environnement de bureau courant (Gnome, KDE et autres). Si un service du bureau particulier ne fonctionne pas bien, il vaut la peine de vérifier si le processus qui devrait le fournir est en fonction.

Les processus `VBoxClient` créent des fichiers dans le dossier personnel de l'utilisateur avec des noms sous la forme `.vboxclient-*.pid` quand ils fonctionnent, pour empêcher un service donné de se démarrer deux fois. Il peut arriver, à cause d'une mauvaise configuration, que ces fichiers se créent sous la propriété de l'administrateur et ne sont pas effacés quand les services s'arrêtent, ce qui les empêchera de démarrer à l'avenir. Si vous ne pouvez pas démarrer les services, vous pourriez vérifier si ces fichiers existent.

12.5 Invités Solaris

12.5.1 Les versions inférieures à Solaris 10 plantent en mode 64 bits

Les versions de Solaris 10 inférieures ou égales à Solaris 10 8/07 ("S10U4") détectent mal les processeurs Intel récents fabriqués depuis 2007. Ce problème fait planter ou stopper le noyau Solaris 64 bits presqu'immédiatement lors du démarrage, tant dans un environnement virtualisé que physique.

La solution recommandée est de mettre à jour vers Solaris au moins 10 5/08 ("S10U5"). D'autres solutions consistent à obliger Solaris à toujours démarrer le noyau 32 bits ou à appliquer un correctif au bogue 6574102 (tant que Solaris utilise le noyau 32 bits).

12.6 Hôte Windows

12.6.1 Problème du serveur `VBoxSVC` out-of-process COM

VirtualBox utilise le Component Object Model (COM) de Microsoft pour la communication inter et intra-processus. Cela permet à VirtualBox de partager une configuration commune entre les processus de différentes machines virtuelles et de fournir plusieurs versions de l'interface utilisateur basées sur une architecture commune. Toutes les informations d'état et la configuration globales sont maintenues par le processus `VBoxSVC.exe`, qui est un service COM hors des processus. À chaque fois que le processus de VirtualBox est démarré, il demande un accès au serveur COM et Windows démarre automatiquement le processus. Remarquez que l'utilisateur final ne devriez jamais le démarrer.

Quand le dernier processus se déconnecte du serveur COM, il se terminera lui-même après quelques secondes. La configuration de VirtualBox (fichiers XML) est maintenue et appartient au serveur COM et les fichiers sont verrouillés à chaque fois que le serveur s'exécute.

Dans certains cas - comme quand une machine virtuelle se termine de manière imprévue -, le serveur COM ne remarquera pas que le client est déconnecté et il restera actif longtemps (10 minutes voire plus), gardant verrouillés les fichiers de configuration. Dans de rares cas, le serveur COM pourrait connaître une erreur interne et, en conséquence, 15s autres processus pourraient

ne pas pouvoir l'initialiser. Dans ces situations, il est recommandé d'utiliser le gestionnaire des tâches de Windows pour tuer le processus `VBoxSVC.exe`.

12.6.2 Changements de CD/DVD non reconnus

Si vous avez affecté un lecteur CD/DVD physique à un invité et si l'invité ne remarque pas les changements de médias, assurez-vous que la fonction de notification de changement de média (MCN) de Windows n'est pas désactivée. Elle est représentée par la clé suivante dans le registre Windows ::

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Cdrom\Autorun
```

Il se peut que certaines applications désactivent cette clé contre l'avis de Microsoft. Si elle est définie à 0, passez-la à 1 et redémarrez votre système. VirtualBox s'appuie sur la notification de Windows de changement de média.

12.6.3 Réponse lente en utilisant le client RDP de Microsoft

Si vous vous connectez à une machine virtuelle avec le client RDP de Microsoft (appelé Remote Desktop Connection), il peut y avoir d'importants délais entre l'entrée (le déplacement de la souris dans un menu est la situation la plus évidente) et la sortie. Ceci car le client RDP récupère l'entrée pendant un certain temps avant de l'envoyer au serveur RDP.

Vous pouvez diminuer l'intervalle en désenchantant une clé du registre Windows sur des valeurs plus petites que celles par défaut, 100. La clé n'existe pas au départ, elle doit être de type DWORD. Son unité de valeur est en millisecondes. Les valeurs autour de 20 conviennent aux connexions avec faible bande passante entre le client et le serveur RDP. Des valeurs autour de 4 peuvent être utilisées pour une connexion Internet à 4 gigaoctets. En général, les valeurs inférieures à 10 donnent une performance très réduite par rapport aux périphériques d'entrée locaux et à l'écran de l'hôte sur lequel fonctionne la machine virtuelle.

Selon que le paramètre à modifier est pour un utilisateur individuel ou pour le système, vous pouvez définir soit

```
HKEY_CURRENT_USER\Software\Microsoft\Terminal Server Client\Min Send Interval
```

soit

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Terminal Server Client\Min Send Interval
```

correctement.

12.6.4 Lancer un initiateur et une cible iSCSI sur un seul système

Des Deadlocks peuvent se produire sur un hôte Windows quand on essaie d'accéder à une cible iSCSI en fonction dans une machine virtuelle invitée avec un initiateur iSCSI (comme Microsoft iSCSI Initiator) en fonction sur l'hôte. Cela vient d'un flaw dans le composant du gestionnaire de cache de Windows et cela donne une réponse lente du système hôte, de plusieurs minutes, suivies d'un message d'erreur "Delayed Write Failed" (délai d'écriture différé) dans la barre système ou dans une fenêtre de message distincte. L'invité est bloqué pendant ce temps et il peut afficher des messages d'erreur ou devenir instable.

La définition d'une variable d'environnement `VBOX_DISABLE_HOST_DISK_CACHE` à 1 activera un contournement de ce problème jusqu'à ce que Microsoft le traite. Par exemple, ouvrez une fenêtre d'invite de commande et démarrez VirtualBox comme ceci :

```
set VBOX_DISABLE_HOST_DISK_CACHE=1
VirtualBox
```

Si cela réduira les performances du disque invité (surtout en écriture), cela ne concerne pas les performances d'autres applications en fonction sur l'hôte.

12.6.5 Adaptateurs réseaux bridgés absents

Si aucun adaptateur bridgé n'apparaît dans la section “Réseau” des paramètres de la VM, cela signifie généralement que le pilote du réseau bridgé n'a pas été installé correctement sur votre hôte. Cela pourrait venir des raisons suivantes :

- Le nombre maximum de filtres autorisés a été atteint sur l'hôte. Dans ce cas, le journal MSI indiquerait le code d'erreur 0x8004a029 retourné à l'installation du composant réseau NetFlt :

```
VBoxNetCfgWinInstallComponent: Install failed, hr (0x8004a029)
```

Vous pouvez essayer d'augmenter le nombre de filtre maximum dans le registre Windows avec la clé suivante :

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Network\MaxNumFilters
```

Le nombre maximum autorisé est de 14. Après le redémarrage, essayez de réinstaller VirtualBox.

- Le cache INF est corrompu. Dans ce cas, le journal d'installation (%windir%\inf\setupapi.log sur XP ou %windir%\inf\setupapi.dev.log sur Vista ou supérieur) indiquerait normalement un échec pour trouver le paquet du pilote adapté aux composants sun_VBoxNetFlt ou sun_VBoxNetFltmp. La solution est alors de désinstaller VirtualBox, de supprimer le cache INF (%windir%\inf\INFCACHE.1), de redémarrer et d'essayer de réinstaller VirtualBox

12.6.6 L'adaptateur réseau Host-only ne peut pas être créé

Si l'adaptateur host-only ne peut pas être créé (soit avec le gestionnaire soit avec VBox-Manage), le cache INF est probablement corrompu. Dans ce cas, le journal d'installation (%windir%\inf\setupapi.log sur XP ou %windir%\inf\setupapi.dev.log sur Vista ou supérieur) indiquerait généralement un échec pour trouver un paquet de pilote adapté au composant sun_VBoxNetAdp. De nouveau, comme pour le problème du réseau bridgé décrit ci-dessus, la solution consiste à désinstaller VirtualBox, à supprimer le cache JNF (%windir%\inf\INFCACHE.1), à redémarrer et à essayer de réinstaller VirtualBox.

12.7 Hôtes Linux

12.7.1 Le module du noyau Linux refuse de se charger

Si le module du noyau Linux (vboxdrv) refuse de se charger, c'est-à-dire que vous avez un message “Error inserting vboxdrv: Invalid argument”, vérifiez (en tant qu'administrateur) la sortie de la commande dmesg pour trouver la raison de l'échec du chargement. Probablement, le noyau n'est pas d'accord avec la version de gcc utilisée pour compiler le module. Assurez-vous d'utiliser le même compilateur que celui utilisé pour construire le noyau.

12.7.2 Lecteur CD/DVD de l'hôte Linux non trouvé

Si vous avez configuré une machine virtuelle pour utiliser le lecteur CD/DVD de l'hôte, mais s'il semble ne pas fonctionner, assurez-vous que l'utilisateur actuel a le droit d'accéder au fichier de périphérique Linux correspondant (/dev/hdc ou /dev/scd0 ou /dev/cdrom ou identique). Sur la plupart des distributions, l'utilisateur doit être ajouté à un groupe correspondant (qui s'appelle en général cdrom ou cdrw).

12.7.3 Lecteur CD/DVD non trouvé sur l'hôte Linux (distributions anciennes)

Sur les anciennes distributions Linux, si votre lecteur CD/DVD a un autre nom, il se peut que VirtualBox soit incapable de le trouver. Sur les hôtes Linux anciens, VirtualBox suit les étapes suivantes pour trouver vos lecteurs CD/DVD :

1. VirtualBox examine si la variable d'environnement `VBOX_CDROM` est définie (voir ci-dessous). Si tel est le cas, VirtualBox ne fait pas les vérifications suivantes.
2. VirtualBox teste si `/dev/cdrom` fonctionne.
3. En plus, VirtualBox vérifie si des lecteurs CD/DVD sont montés en vérifiant `/etc/mtab`.
4. En outre, VirtualBox vérifie si une des entrées de `/etc/fstab` pointe vers un lecteur CD/DVD.

En d'autres termes, vous pouvez essayer de définir `VBOX_CDROM` pour contenir vos lecteurs CD/DVD, séparés par des deux-points, par exemple comme suit :

```
export VBOX_CDROM='/dev/cdrom0:/dev/cdrom1'
```

Sur les distributions Linux modernes, VirtualBox utilise la couche d'abstraction matérielle (hal) pour localiser le matériel CD et DVD.

12.7.4 Disquette non trouvée sur un hôte Linux

Les instructions précédentes (pour les lecteurs CD et DVD) s'appliquent aussi aux disquettes, sauf que sur les distributions anciennes, VirtualBox teste par défaut les périphériques `/dev/fd*` ce que vous pouvez changer avec la variable d'environnement `VBOX_FLOPPY`.

12.7.5 Messages d'erreur étranges de l'IDE invité lors de l'écriture sur un CD/DVD

Si le support expérimental d'écriture sur un CD/DVD est activé avec une mauvaise configuration de l'hôte et de l'invité VirtualBox, il est possible que vos efforts pour accéder à l'écriture sur CD/DVD échouent et n'aboutissent qu'à des messages d'erreur du noyau invité (pour les invités Linux) ou à des messages d'erreur de l'application (pour les invités Windows). VirtualBox effectue les vérifications de cohérence habituelles quand une VM est allumée (en particulier, il quitte avec un message d'erreur si l'utilisateur qui démarre la VM ne peut pas écrire sur le périphérique du graveur CD/DVD), mais il ne peut pas détecter toutes les mauvaises configurations. La configuration de l'OS hôte et de l'invité requise n'est pas spécifique à VirtualBox, mais quelques problèmes fréquents sont listés ici, ils se sont produits en lien avec VirtualBox.

Vous devez faire très attention à utiliser le bon périphérique. Le nom du fichier du lecteur CD/DVD de l'hôte configuré (dans la plupart des cas, `/dev/cdrom`) doit pointer vers le périphérique qui permet d'écrire sur l'unité CD/DVD. Pour les unités du graveur CD/DVD, connecté à un contrôleur SCSI ou à un contrôleur IDE qui fait interface avec le sous-système SCSI de Linux (ce qui est classique pour certains contrôleurs SATA), il doit renvoyer au nœud de périphérique SCSI (comme `/dev/scd0`). Même pour les unités de graveurs de CD/DVD en IDE, il doit renvoyer au nœud du lecteur CD-ROM adéquat (comme `/dev/scd0`) si le module du noyau `ide-scsi` est chargé. Ce module est requis pour le support du graveur CD/DVD avec tous les noyaux Linux 2.4 et avec certains noyaux 2.6 des débuts. De nombreuses distributions Linux chargent ce module à chaque fois que le graveur CD/DVD est détecté dans le système, même si le noyau supporterait des graveurs CD/DVD sans le module. VirtualBox supporte l'utilisation des fichiers de périphérique IDE (comme `/dev/hdc`), pourvu que le noyau le supporte et que le module `ide-scsi` ne soit pas chargé.

Des règles similaires (sauf que dans l'invité, le graveur CD/DVD est toujours un périphérique IDE) s'appliquent à la configuration de l'invité. Ce paramétrage étant très classique, il est probable que la configuration par défaut de l'invité fonctionne comme prévu.

12.7.6 Problème de l'IPC VBoxSVC

Sur Linux, VirtualBox utilise une version personnalisée de XPCOM de Mozilla (modèle d'objet du composant multi-plateformes) pour la communication inter et intra processus (IPC). Le processus VBoxSVC sert de hub de communication entre plusieurs processus de VirtualBox et il maintient la configuration globale, c'est-à-dire la base de données XML. Au démarrage d'un composant de VirtualBox, les processus VBoxSVC et VirtualBoxXPCOMIPCD sont lancés automatiquement. Ils ne sont accessibles qu'à partir du compte utilisateur qui l'a lancé. VBoxSVC possède la base de données de la configuration de VirtualBox qui se trouve normalement dans `~/.config/VirtualBox`, ou dans le répertoire de configuration adéquat de votre système d'exploitation. Tant qu'il est en fonction, les fichiers de configuration sont verrouillés. La communication entre les composants de VirtualBox et VBoxSVC est faite via une socket de domaine local qui se trouve dans `/tmp/.vbox-<username>-ipc`. En cas de problèmes de communication (par exemple si une application VirtualBox ne peut pas communiquer avec VBoxSVC), clôturez les démons et supprimez le répertoire de la socket du domaine local.

12.7.7 L'USB ne fonctionne pas

Si l'USB ne fonctionne pas sur votre hôte Linux, assurez-vous que l'utilisateur actuel fait partie du groupe `vboxusers`. Sur les hôtes anciens, vous devez vous assurer que l'utilisateur a le droit d'accéder au système de fichiers USB (`usbfs`), sur lequel s'appuie VirtualBox pour récupérer des informations valides sur les périphériques USB de votre hôte. Le reste de cette section ne s'applique qu'à ces anciens systèmes.

Comme `usbfs` est un système de fichiers virtuel, un `chmod` sur `/proc/bus/usb` n'a aucun effet. Les droits sur `usbfs` ne peuvent donc être changés que si vous éditez le fichier `/etc/fstab`.

Par exemple, la plupart des distributions Linux comportent un groupe utilisateur qui s'appelle `usb` ou similaire, dont l'utilisateur actuel doit faire partie. Pour donner à tous les utilisateurs de ce groupe un accès à `usbfs`, assurez-vous que la ligne suivante est présente :

```
# 85 is the USB group
none      /proc/bus/usb      usbfs      devgid=85,devmode=664      0      0
```

Remplacez 85 par l'ID du groupe correspondant à votre système (cherchez dans `/etc/group` "usb" ou proche). Sinon, si vous vous moquez des considérations de sécurité, donnez à tous les utilisateurs l'accès à l'USB en changeant "664" en "666".

Les distributions sont très créatives sur le script qui monte le système de fichiers `usbfs`. Parfois, la commande est cachée à des endroits improbables. Pour SuSE 10.0, la commande de montage fait partie du fichier de configuration `udev /etc/udev/rules.d/50-udev.rules`. Comme cette distribution n'a aucun groupe d'utilisateurs appelé `usb`, vous pouvez utiliser par exemple le groupe `vboxusers` qui a été créé par l'installateur de VirtualBox. Les numéros des groupes étant affectés de manière dynamique, l'exemple suivant utilise 85 comme modèle. Modifiez la ligne contenant (on a inséré un retour à la ligne pour améliorer la lisibilité)

```
DEVPATH="/module/usbcore", ACTION=="add",
        RUN+="/bin/mount -t usbfs usbfs /proc/bus/usb"
```

et ajoutez les options nécessaires (assurez-vous que tout est sur une seule ligne) :

```
DEVPATH="/module/usbcore", ACTION=="add",
        RUN+="/bin/mount -t usbfs usbfs /proc/bus/usb -o devgid=85,devmode=664"
```

Debian Etch a sa commande de montage dans `/etc/init.d/mountkernfs.sh`. Cette distribution n'ayant pas de groupe `usb`, la solution la plus simple est d'autoriser tous les membres du groupe `vboxusers` à accéder au sous-système USB. Modifiez la ligne

```
domount usbfs usbdevfs /proc/bus/usb -onoexec,nosuid,nodev
```

pour qu'elle contienne

```
domount usbfs usbdevfs /proc/bus/usb -onoexec,nosuid,nodev,devgid=85,devmode=664
```

Comme d'habitude, remplacez 85 par le vrai numéro du groupe qui devrait avoir accès aux périphériques USB.

D'autres distributions font des opérations identiques dans des scripts stockés dans le répertoire `/etc/init.d`.

12.7.8 Noyaux PAX/grsec

Les noyaux Linux incluant le correctif grsec (voir <http://www.grsecurity.net/>) et ses dérivés doivent désactiver `PAX_MPROTECT` pour que les binaires VBox puissent démarrer une VM. Ceci car VBox doit créer un code exécutable en mémoire anonyme.

12.7.9 pool vmalloc du noyau Linux dépassé

Quand on exécute un grand nombre de VMs avec un beaucoup de RAM sur un système Linux (disons 20 VMs de 1Go de RAM chacune), les VMs supplémentaires pourraient ne pas réussir à démarrer avec une erreur du noyau disant que le pool `vmalloc` est dépassé et que vous devriez l'agrandir. Le message d'erreur vous dit aussi de spécifier `vmalloc=256MB` dans votre liste des paramètres du noyau. Si l'ajout de ce paramètre à votre configuration de GRUB ou de LILO empêche le noyau de démarrer (avec un message d'erreur bizarre tel que "failed to mount the root partition"), vous avez probablement un conflit de mémoire entre votre noyau et la RAM disque initiale. Vous pouvez résoudre cela en ajoutant le paramètre suivant à votre configuration de GRUB :

```
uppermem 524288
```

12.8 Hôtes Solaris

12.8.1 Ne peut pas démarrer de VM, pas assez de mémoire contiguë

Le système de fichiers ZFS est connu pour utiliser presque toute la RAM disponible comme du cache si les paramètres système par défaut ne sont pas modifiés. Cela peut conduire à une énorme fragmentation de la mémoire de l'hôte, empêchant les VMS de VirtualBox de démarrer. Nous vous recommandons de limiter la limite du cache ZFS en ajoutant une ligne

```
set zfs:zfs_arc_max = xxxx
```

à `/etc/system` où `xxxx` octets est la quantité de mémoire utilisable pour le cache ZFS.

12.8.2 La VM s'arrête avec des erreurs de dépassement de mémoire sur les hôtes Solaris 10

Les hôtes Solaris 10 32 bits (bogue 1225025) exigent un espace d'échange supérieur ou égal à la taille de la mémoire physique de l'hôte. Par exemple, 8 Go de mémoire physique exigerait au moins 8 Go d'échange. Vous pouvez configurer cela pendant l'installation de Solaris 10 en choisissant une 'installation personnalisée' et en modifiant les partitions par défaut.

Note: Cette restriction ne s'applique qu'aux hôtes Solaris 32 bits, les hôtes 64 bits ne sont pas concernés !

Pour les installations Solaris 10 existantes, il faut monter une image d'échange supplémentaire et l'utiliser comme échange. D'où le fait que si vous avez un échange de 1 Go et 8 Go de mémoire physique, vous devez ajouter un échange supplémentaire de 7 Go. Vous pouvez faire cela comme suit :

Pour ZFS (en tant qu'administrateur) :

```
zfs create -V 8gb /<ZFS volume>/swap
swap -a /dev/zvol/dsk/<ZFS volume>/swap
```

Pour monter le système de fichiers au démarrage, ajoutez la ligne suivante à /etc/vfstab :

```
/dev/zvol/dsk/<ZFS volume>/swap - - swap - no -
```

Sinon, vous pouvez agrandir l'espace existant en utilisant :

```
zfs set volsize=8G rpool/swap
```

Et redémarrer le système pour que les changements prennent effet.

Pour UFS (en tant qu'administrateur) :

```
mkfile 7g /path/to/swapfile.img
swap -a /path/to/swapfile.img
```

Pour le monter au redémarrage, ajoutez la ligne suivante à /etc/vfstab :

```
/path/to/swap.img - - swap - no -
```

13 Guide de sécurité

13.1 Aperçu

13.1.1 Principes généraux de sécurité

Les principes suivants sont fondamentaux pour utiliser une application en toute sécurité.

Maintenir à jour le logiciel Une des bonnes pratiques de sécurité est d'avoir des versions du logiciel et des correctifs à jour. Activez la notification de mise à jour de VirtualBox pour être averti quand une nouvelle version de VirtualBox est disponible. Quand vous mettez à jour VirtualBox, n'oubliez pas de mettre à jour aussi les suppléments invité. Maintenez à jour le système d'exploitation hôte ainsi que l'invité.

Restriindre l'accès réseau des services critiques Utilisez les moyens, tels qu'un pare-feu, pour protéger votre ordinateur et vos invités de l'extérieur. Choisir le bon mode de réseau pour des VMs permet de distinguer le réseau de l'hôte de l'invité et vice versa.

Suivre le principe du privilège le plus restreint Le principe du privilège le plus restreint dispose qu'il faut donner aux utilisateurs les priviléges strictement nécessaires pour accomplir leurs tâches. Exécutez toujours VirtualBox en tant qu'utilisateur ordinaire. Nous déconseillons fortement d'exécuter VirtualBox avec des priviléges système.

Choisissez des droits restrictifs quand vous créez des fichiers de configuration, par exemple quand vous créez /etc/default/virtualbox, voir chapitre 2.3.3.7, *Options d'installation automatique*, page 44. Le mode 0600 serait idéal.

Surveiller l'activité du système La sécurité du système repose sur trois piliers : de bons protocoles de sécurité, une bonne configuration du système et la surveillance du système. L'évaluation et la lecture des fruits de l'évaluation constituent la troisième exigence. Chaque composant d'un système a un certain degré de capacité à être surveillé. Suivez les conseils d'évaluation de ce document et surveillez régulièrement les fruits de l'évaluation.

Garder à jour les informations de sécurité Oracle améliore en permanence ses logiciels et sa documentation. Vérifiez ce passage chaque année pour truquer les révisions.

13.2 Installation et configuration sécurisées

13.2.1 Aperçu de l'installation

Vous ne devriez télécharger le paquet de base de VirtualBox qu'à partir d'une source de confiance, telle que le site Internet officiel <http://www.virtualbox.org>. Vous devriez vérifier l'intégrité du paquet avec la somme de contrôle SHA256 fournie sur le site officiel.

Vous pouvez trouver des instructions générales d'installation de VirtualBox pour les hôtes supportés au chapitre 2, *Détails sur l'installation*, page 37.

Sur les hôtes Windows, l'installateur permet de désactiver le support USB, le support du réseau bridgé, le support du réseau host-only et les bindings du langage Python, voir chapitre 2.1, *Installation sur des hôtes Windows*, page 37. Toutes ces fonctionnalités sont activées par défaut mais la désactivation de certaines d'entre elles pourrait être adaptée si vous n'avez pas besoin des fonctionnalités correspondantes sur une machine virtuelle. Les bindings du langage Python ne sont

nécessaires que si vous devez utiliser l'API de VirtualBox avec des applications Python externes. En particulier, le support USB et le support des deux modes réseaux nécessitent d'installer des pilotes du noyau Windows sur l'hôte. Donc, la désactivation de ces fonctions peut non seulement permettre d'encadrer l'utilisateur dans certaines fonctionnalités, mais aussi de minimiser le champ d'attaque d'un pirate potentiel.

En général, on installe tout le paquet VirtualBox. Il faut faire l'installation avec les priviléges système. Tous les binaires de VirtualBox devraient être lancés en tant qu'utilisateur ordinaire et jamais en tant qu'utilisateur privilégié.

Le pack d'extension d'Oracle VM VirtualBox apporte des fonctionnalités supplémentaires et il doit être téléchargé et installé à part, voir chapitre [1.5, Installer VirtualBox et les packs d'extension](#), page [17](#). Comme pour le paquet de base, vous devriez vérifier la somme de contrôle SHA256 du pack d'extension. Comme le système d'installation exige des priviléges système, VirtualBox vous demandera le mot de passe système pendant l'installation du pack d'extension.

13.2.2 Configuration post installation

Normalement, aucune configuration post installation d'un composant de VirtualBox n'est nécessaire. Cependant, sur les hôtes Solaris et Linux, il faut configurer les droits adaptés pour que les utilisateurs exécutent des VMs et puissent accéder à certaines ressources de l'hôte. Par exemple, les utilisateurs Linux doivent faire partie du groupe `vboxusers` pour pouvoir donner des périphériques USB à un invité. Si vous devriez accéder à une interface série à partir d'une VM, il faut donner les bons droits à l'utilisateur pour qu'il puisse accéder à ce périphérique. La même chose s'applique à d'autres ressources comme la partition brute, les lecteurs DVD/CD et les périphériques de son.

13.3 Fonctions de sécurité

Cette section évoque les mécanismes de sécurité spécifiques à VirtualBox.

13.3.1 Le modèle de sécurité

Une des propriétés des gestionnaires de machines virtuelles (VMMs) comme VirtualBox est d'enfermer un invité en l'exécutant dans un environnement protégé, une machine virtuelle laquelle fonctionne en tant que processus d'un utilisateur du système d'exploitation hôte. L'invité ne peut pas communiquer directement avec le matériel hôte ou avec d'autres ordinateurs, mais uniquement via le VMM. Le VMM offre des ressources physiques et des périphériques émulés à l'invité, auxquelles on accède par le système d'exploitation hôte pour effectuer les tâches nécessaires. Les paramètres de la VM contrôlent les ressources fournies à l'invité, par exemple la quantité de mémoire de l'invité ou le nombre de processeurs invités (voir chapitre [3.3, Paramètres généraux](#), page [51](#)) et les fonctionnalités activées pour cet invité (par exemple le contrôle à distance, certains paramètres d'affichage et autres).

13.3.2 Configuration sécurisée des machines virtuelles

Plusieurs aspects de la configuration d'une machine virtuelle sont sujets à des considérations de sécurité.

13.3.2.1 Le réseau

Le mode réseau par défaut des VMs est NAT, ce qui signifie que la VM se comporte comme un ordinateur derrière un routeur, voir chapitre [6.3, Network Address Translation \(NAT\)](#), page [100](#). L'invité fait partie d'un sous-réseau privé appartenant à cette VM et l'adresse IP de l'invité n'est

pas visible de l'extérieur. Ce mode réseau fonctionne sans paramétrage supplémentaire et il suffit pour la plupart des besoins.

Si vous utilisez le réseau bridgé, la VM se comporte comme un ordinateur dans le même réseau que l'hôte, voir chapitre [6.4, Réseau Bridged](#), page [102](#). Dans ce cas, l'invité a un accès réseau identique à l'hôte et un pare-feu pourrait être nécessaire pour protéger d'autres ordinateurs du sous-réseau contre des invités malveillants potentiels et pour protéger l'invité contre un accès direct par les autres ordinateurs. Dans certains cas, il est intéressant de songer à utiliser une règle de redirection pour un port spécifique en mode NAT, plutôt que d'utiliser le réseau bridgé.

Certaines configurations n'ont pas besoin que la VM soit connectée au réseau public. Le réseau interne (voir chapitre [6.5, Réseau interne](#), page [103](#)) ou le réseau host-only (voir chapitre [6.6, Réseau Host-only](#), page [104](#)) suffisent souvent pour connecter des VMs entre elles ou pour ne connecter des VMs qu'à l'hôte mais pas au réseau public.

13.3.2.2 Authentification sur un bureau distant (VRDP)

Quand on utilise le pack d'extension de VirtualBox fourni par Oracle pour accéder à distance (VRDP), on peut éventuellement utiliser plusieurs méthodes pour configurer l'authentification RDP. La méthode "null" est très peu sûre, vous devriez l'éviter sur un réseau public. Voir chapitre [7.1.5, Authentification RDP](#), page [113](#) pour les détails.

13.3.2.3 Presse-papier

Le presse-papier partagé permet aux utilisateurs de partager des données entre l'hôte et l'invité. L'activation du presse-papier en "mode bidirectionnel" permet à l'invité de lire et d'écrire dans le presse-papier de l'hôte. Le mode "hôte vers invité" et "Invité vers hôte" limitent l'accès à un seul sens. Si l'invité peut accéder au presse-papier de l'hôte, il peut aussi accéder potentiellement à des données sensibles de l'hôte partagées dans le presse-papier.

Si l'invité peut lire et/ou écrire dans le presse-papier de l'hôte, un utilisateur distant qui se connecte à l'invité par le réseau aura également cette possibilité, ce qui peut ne pas être souhaitable. Par conséquent, le presse-papier partagé est désactivé pour les nouvelles machines.

13.3.2.4 Dossiers partagés

Si un dossier de l'hôte est partagé avec l'invité, n'importe quel utilisateur connecté à distance à l'invité par le réseau peut accéder aussi à ces fichiers car le mécanisme de partage des dossiers ne peut pas être désactivé de manière sélective pour des utilisateurs distants.

13.3.2.5 Accélération graphique 3D

L'activation de la 3D avec les suppléments invité expose l'hôte à des risques supplémentaires de sécurité ; voir chapitre [4.4.1, Accélération 3D matérielle \(OpenGL and Direct3D 8/9\)](#), page [77](#).

13.3.2.6 CD/DVD passthrough

L'activation du CD/DVD passthrough permet à l'invité d'effectuer des opérations avancées sur le lecteur CD/DVD, voir chapitre [5.9, Support des CD/DVD](#), page [96](#). Cela peut poser un problème de sécurité car un invité pourrait écraser des données sur un DVD.

13.3.2.7 USB passthrough

La présentation de périphériques USB à l'invité offre à l'invité un accès complet à ces périphériques, voir chapitre [3.10.1, Paramètres USB](#), page [60](#). Par exemple, outre la lecture et l'écriture du contenu des partitions d'un disque USB externe, l'invité pourra également lire et écrire la table de partitions et des données matérielles sur ce disque.

13.3.3 Configurer et utiliser l'authentification

Les composants suivants de VirtualBox peuvent utiliser des mots de passe pour l'authentification :

- Lors de l'utilisation du stockage iSCSI à distance et si le serveur de stockage exige une authentification, vous pouvez fournir un mot de passe d'initiateur avec la commande `VBoxManage storageattach`. Tant que vous ne fournissez pas de réglage de mots de passe (l'option

`--settingspwmfile`

en ligne de commande), ce mot de passe secret est stocké **sans chiffrement** dans la configuration de la machine et il est donc potentiellement lisible sur l'hôte. Voir chapitre 5.10, *Serveurs iSCSI*, page 97 et chapitre 8.18, *VBoxManage storageattach*, page 142.

- Quand vous utilisez le service Web de VirtualBox pour contrôler un hôte VirtualBox à distance, les connexions au service sont authentifiées de plusieurs façons. Ceci est décrit en détails dans manuel de référence du kit de développement logiciel de VirtualBox (SDK) ; merci de voir chapitre 11, *Interfaces de programmation de VirtualBox*, page 207.

13.3.4 Opérations potentiellement non sécurisées

Les fonctions suivantes de VirtualBox peuvent présenter des problèmes de sécurité :

- L'activation de la 3D par les suppléments invité expose l'hôte à des risques de sécurité supplémentaires ; voir chapitre 4.4.1, *Accélération 3D matérielle (OpenGL and Direct3D 8/9)*, page 77.
- En téléportant une machine, le flux de données par lequel passe le contenu de la mémoire de la machine est transféré d'un hôte à l'autre sans chiffrement. Un tiers ayant un accès au réseau par lequel les données sont transférées pourrait donc intercepter ces données. On pourrait utiliser un tunnel SSH pour sécuriser la connexion entre les deux hôtes. Mais au moment de téléporter une VM par un réseau non fiable, la première question à vous poser est celle de savoir comment les VMs peuvent accéder de manière sécurisée à la/aux même(s) image(s) de disque virtuel avec une performance raisonnable.
- Quand vous utilisez le service Web de VirtualBox pour contrôler un hôte VirtualBox à distance, les connexions au service (par lesquelles les appels de l'API sont transférées en SOAP XML) ne sont pas chiffrées, elles utilisent par défaut le HTTP en clair. C'est un risque potentiel de sécurité ! Pour des détails sur le service Web, merci de voir chapitre 11, *Interfaces de programmation de VirtualBox*, page 207.

Les services web ne sont pas lancés par défaut. Merci de vous reporter au chapitre 9.21, *Démarrer le service Web de VirtualBox automatiquement*, page 188 pour voir comment démarrer ce service et activer le support SSL/TLS. Il faut le démarrer en tant qu'utilisateur ordinaire et seules les VMs de cet utilisateur sont contrôlables. Par défaut, le service sonde localhost, empêchant toute connexion distante.

- Le trafic envoyé par une connexion réseau en tunnel UDP n'est pas chiffré. Vous pouvez soit le chiffrer au niveau du réseau hôte (avec IPsec), soit utiliser des protocoles chiffrés dans le réseau invité (tel que SSH). The security properties are similar to bridged Ethernet.

13.3.5 Chiffrement

Les composants suivants de VirtualBox utilisent le chiffrement pour protéger ces données sensibles :

- Quand on utilise le pack d'extension de VirtualBox fourni par Oracle pour le support du bureau distant (VRDP), les données peuvent être éventuellement chiffrées. Voir chapitre 7.1.6, *Chiffrement RDP*, page 115 pour des détails. Seule la méthode Enhanced RDP Security (RDP5.2) avec le protocole TLS offre une connexion sécurisée. La Standard RDP Security (RDP4 et RDP5.1) est vulnérable à une attaque man-in-the-middle.

14 Limites connues

14.1 Fonctions expérimentales

Certaines fonctions de VirtualBox sont étiquetées comme étant expérimentales. De telles fonctions sont offertes sur une base “as-is” et elles ne sont pas formellement supportées. Cependant, des retours et les suggestions sur ces fonctions sont bienvenus. Voici une liste complète des fonctions expérimentales :

- Pilote graphique WDDM Direct3D pour les invités Windows
- Support de l'accélération matérielle 3D pour les invités Windows, Linux, et Solaris
- Support de l'accélération 2D pour les invités Windows
- PCI pass-through (hôtes Linux uniquement)
- Invités Mac OS X (hôtes hosts uniquement)
- Émulation du chipset ICH9
- firmware EFI
- Pass-through lecteur CD/DVD de l'hôte
- Support d'iSCSI par le réseau interne
- Signalement synthétique du processeur

14.2 Problèmes connus

La section suivante décrit les problèmes connus avec VirtualBox 4.2.51. Sauf autrement spécifié, il est prévu de corriger ces problèmes dans les prochaines versions.

- Les **limites SMP (multiprocesseur) de l'invité** existent :
 - **Faibles performances** avec des invités 32 bits sur des processeurs AMD. Cela concerne surtout les invités Windows et Solaris, mais aussi probablement certaines révisions du noyau Linux. Corrigé partiellement dans la 3.0.6 pour les invités Windows NT, 2000, XP et 2003 32 bits. Exige l'installation des suppléments invité 3.0.6 ou supérieur.
 - **Faibles performances** avec les invités 32 bits sur certains modèles de processeurs Intel qui n'incluent pas le support de l'optimisation matérielle APIC virtuel. Cela concerne surtout les invités Windows et Solaris, mais probablement également certaines révisions du noyau Linux. Corrigé partiellement dans la 3.0.12 pour les invités Windows NT, 2000, XP et 2003 32 bits. Exige l'installation des suppléments invité 3.0.12 ou supérieur.
- **Invités 64 bits sur des systèmes hôtes 32 bits avec VT-x** peut provoquer des instabilités sur votre système. Si vous vivez cela, n'essayez pas d'exécuter des invités 64 bits. Reportez-vous au forum des utilisateurs de VirtualBox pour des informations supplémentaires.

- **NX (empêche l'exécution et l'exécution de données)** ne fonctionne que sur les invités en fonction sur des hôtes 64 bits ou sur des hôtes 32 bits ayant activé PAE et la virtualisation matérielle doit être activée.
- Pour que le **support Direct3D de base dans les invités Windows** fonctionne, il faut installer les suppléments invité en mode “sans échec” dans Windows. Appuyez sur F8 quand l'invité Windows démarre et sélectionnez “Mode sans échec”, puis installez les suppléments invité. Sans cela, le mécanisme de protection des fichiers de Windows interfèrera dans le remplacement des DLLs installés par VirtualBox et il restaurera les DLLs du système Windows d'origine.

Note: Cela ne s'applique pas au pilote graphique expérimental Direct3D WDDM disponible pour les invités Vista et Windows 7 inclus avec VirtualBox 4.1.

- **contrôle de l'invité.** Sur les invités Windows, un processus lancé via le support d'exécution du contrôle de l'invité ne pourra pas afficher d'interface graphique, *sauf* si le compte utilisateur sous lequel il est lancé est connecté et s'il a une session de bureau.

En outre, pour utiliser des comptes avec ou sans mot de passe vide, vous devez modifier la politique de groupe de l'invité. Pour ce faire, ouvrez un éditeur des règles de groupes en ligne de commande en tapant `gpedit.msc`, ouvrez la clé `Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options` et passez la valeur de `Accounts: Limit local account use of blank passwords to console logon only` en `Disabled`.

- **La compression d'images de disques virtuels est limitée aux fichiers VDI.** La commande `VBoxManage modifyhd --compact` n'est actuellement implémentée que les fichiers VDI. Pour l'instant, la seule façon d'optimiser la taille des images de disques virtuels dans d'autres formats (VMDK, VHD) est de cloner l'image puis d'utiliser l'image clonée dans la configuration de la VM.

- **import/export d'OVF :**

- La localisation des OVF (plusieurs langues dans un fichier OVF) n'est pas encore supportée.
- Certaines sections OVF comme `StartupSection`, `DeploymentOptionSection` et `InstallSection` sont ignorées.
- Les documents de vnement OVF, y compris leurs sections de propriétés et la configuration d'applicatifs avec des images ISO, ne sont pas encore supportés.
- Les fichiers distants avec HTTP ou d'autres mécanismes ne sont pas encore supportés.

- Ni le **mode échelonné** ni le **mode transparent** ne fonctionnent bien avec les invités qui utilisent les fonctions 3D d'OpenGL (telles que celles avec les `get:nnn` de fenêtres ayant activé compiz).

- Le serveur RDP du pack d'extension de VirtualBox ne supporte que les flux audio au format 22.05kHz stereo 16 bit. Si le client RDP demande d'autres formats de son, ils ne seront pas sonores.

- La préservation de l'apparence de l'affichage en mode échelonné ne fonctionne que sur les hôtes Windows et sur les hôtes Mac OS X.

- Sur les **hôtes Mac OS X**, les fonctions suivantes ne sont pas encore implémentées :

- Émulation du verrouillage numérique
- Mesure de la fréquence du processeur

- Jeu de ballon avec la mémoire

- **Invités Mac OS X :**

- Les invités Mac OS X ne peuvent fonctionner que sur un certain matériel hôte. Pour des détails sur les limites de la licence et du matériel hôte, merci de voir chapitre [3.1.1, Invités Mac OS X](#), page [49](#) et vérifiez les conditions de la licence logicielle d'Apple.
- VirtualBox n'offre pas de suppléments invité pour Mac OS X pour l'instant.
- La résolution graphique est par défaut de 1024x768 car Mac OS X se rabat sur le support d'affichage EFI intégré. Voir chapitre [3.12.1, Modes graphiques dans EFI](#), page [63](#) pour plus d'informations sur la façon de modifier les modes graphiques EFI.
- Même quand il est inactif, un invité Mac OS X consomme 100% du processeur. C'est un problème de gestion d'énergie qui sera corrigé dans une prochaine version.
- Les invités Mac OS X ne fonctionnent qu'avec un processeur sur une VM. Le support du SMP sera fourni dans une version future.
- Selon votre système et votre version de Mac OS X, vous pourriez connaître des plantages de l'invité après quelque temps. Vous pouvez corriger cela en désactivant l'économie d'énergie (passez le timeout à "Never" (jamais)) dans les préférences du système.
- Par défaut, l'EFI de VirtualBox active la sortie de débogage du noyau Mac OS X pour vous aider à diagnostiquer les problèmes de démarrage. Remarquez qu'il y a de nombreux messages et toutes les erreurs ne sont pas fatales (elles s'afficheraient aussi sur votre Mac physique). Vous pouvez désactiver ces messages en lançant cette commande :

```
VBoxManage setextradata "nom VM" "VBoxInternal2/EfiBootArgs" " "
```

To revert to the previous behavior, use:

```
VBoxManage setextradata "nom VM" "VBoxInternal2/EfiBootArgs" ""
```

- **Hôtes Solaris :**

- Il n'y a pas de support des périphériques USB connectés aux hôtes Solaris 10.
- Le support USB sur les hôtes Solaris exige Solaris version 11 snv_124 ou supérieur. Les Webcams et les autres périphériques de temps (isochronous) sont connus pour donner de faibles performances.
- Aucune informations d'ACPI (état de la batterie, source d'énergie) n'est signalée à l'invité.
- Pas de support des adaptateurs wifi avec le réseau bridé.
- Le réseau bridé basé sur Crossbow sur les hôtes Solaris ne fonctionne pas directement avec les liens agrégés. Cependant, vous pouvez créer à la main un VNIC (en utilisant `dladm`) via le lien agrégé et utilisez-le avec une VM. Cette limite technique sera corrigée dans une future version de Solaris 11.

- **Suppléments invité de la version 4.1, 4.1.2 et 4.1.4 pour Windows** Il se peut que le pilote graphique WDDM de VirtualBox peut être installé et gardé dans le système invité après la désinstallation des suppléments invité. Cela vient d'un bogue du désinstalleur des suppléments invité.

Note: Cela **ne s'applique pas** aux mises à jour des suppléments invité, c'est-à-dire que l'installation d'une version des suppléments invité par-dessus une autre fonctionne correctement.

Pour résoudre ce problème, vous devriez désinstaller le pilote graphique WDDM de VirtualBox à la main. Pour cela, ouvrez le gestionnaire de périphériques et vérifiez si l'adaptateur d'affichage s'appelle "VirtualBox Graphics Adapter ..". Sinon - il n'y a rien à faire. Si oui, - faites un clic droit sur VirtualBox Graphics Adapter dans le gestionnaire de périphériques, sélectionner "Désinstaller", cochez "Effacer le pilote de ce périphérique" et cliquez sur "OK". Une fois que la désinstallation est finie - dans le gestionnaire de périphériques, allez dans le menu "Action" et sélectionnez "Analyser les changements matériels" pour utiliser le bon pilote (celui par défaut de Windows) pour l'adaptateur graphique.

- Ni le pilote *virtio* ni *Intel PRO/1000* des **invités Windows XP** ne supportent la segmentation hors charge (offloading). Donc, les invités Windows XP ont des vitesses de transmission plus lentes que d'autres types d'invités. Reportez-vous à l'article 842264 du MS Knowledge base pour des informations supplémentaires.
- **Suppléments invité pour OS/2.** Les dossiers partagés ne sont pas encore supportés avec les invités OS/2. De plus, les fenêtres transparentes et le redimensionnement de l'invité ne seront probablement jamais implémentés à cause de limites intrinsèques du système graphique d'OS/2.

15 Historique des changements

Cette section résume les changements entre les versions de VirtualBox. Remarquez que cet historique n'est pas exhaustif ; tous les changements ne sont pas listés.

Les numéros de version de VirtualBox consistent en trois nombres séparés par des points où le premier et le second numéro représentent la version majeure et le 3ème nombre la version mineure. Les numéros des versions mineures des versions officielles sont toujours là. Un numéro de version mineure ad hoc représente une construction de développement ou de test. En outre, chaque construction contient un numéro de révision.

15.1 Version xx.xx.0 (xxxx-xx-xx)

This is a major update. The following major new features were added:

- dummy

In addition, the following items were fixed and/or added:

- dummy
- Main: use the XDG standard configuration folder instead of .VirtualBox on systems where it is appropriate (bug #5099)
- VBoxManage: list more information about hard disk/DVD/floppy media, and support the --long option to show really all available details.
- Shared Clipboard/X11: support for BMP-format images, contributed by François Revol.

15.2 Version 4.2.0 (2012-09-13)

This is a major update. The following major new features were added:

- Improved Windows 8 support, in particular many 3D-related fixes
- GUI: VM groups (bug #288)
- GUI: expert mode for wizards
- GUI: allow to alter some settings during runtime
- Support for up to 36 network cards, in combination with an ICH9 chipset configuration (bug #8805)
- Resource control: added support for limiting network IO bandwidth; see chapitre 6.9, *Limiter la bande passante des E/S réseaux*, page 107 (bug #3653)
- Added possibility to start VMs during system boot on Linux, OS X and Solaris; see chapitre 9.24, *Démarrer des machines virtuelles lors de l'amorçage du système*, page 194 (bug #950)
- Added experimental support for Drag'n'drop from the host to Linux guests. Support for more guests and for guest-to-host is planned. (bug #81)

- Added support for parallel port passthrough on Windows hosts
- Enhanced API for controlling the guest; please see the SDK reference and API documentation for more information

In addition, the following items were fixed and/or added:

- Mac OS X hosts: sign application and installer to avoid warnings on Mountain Lion
- VMM: fixed a potential host crash triggered by shutting down a VM when another VM was running (only affected 32-bit hosts and 64-bit OS X hosts, 4.1 regression, bug #9897)
- VMM: fixed a potential host crash under a high guest memory pressure (seen with Windows 8 guests)
- VMM: improved VM context switch performance for Intel CPUs using nested paging
- VMM: added support for *FlushByASID* features of AMD CPUs (Bulldozer and newer)
- VMM: fixed unreal mode handling on older CPUs with VT-x (gPXE, Solaris 7/8/9; bug #9941)
- VMM: fixed MP tables fixes for I/O APIC interrupt routing relevant for ancient SMP guests (e.g. old OS/2 releases)
- VMM: support recent VIA CPUs (bug #10005)
- VMM: fixed handling of task gates if VT-x/AMD-V is disabled
- VMM: page fusion fixes
- GUI: network operations manager
- GUI: allow taking screenshots of the current VM window content (bug #5561)
- GUI: allow automatically sorting of the VM list
- GUI: allow starting of headless VMs from the GUI
- GUI: allow reset, shutdown and poweroff from the Manager window
- GUI: allow to globally limit the maximum screen resolution for guests
- GUI: show the full medium part on hovering the list of recently used ISO images
- GUI: do not create additional folders when a new machine has a separator character in its name (bug #6541)
- GUI: don't crash on terminate if the settings dialog is still open (bug #9973)
- GUI: consider scaled DPI values when display fonts on Windows hosts (bug #9864)
- GUI: if a bridged network interface cannot be found, don't refuse to start the VM but allow the user to change the setting immediately
- Snapshots: fixed a crash when restoring an old snapshot when powering off a VM (bugs #9364, #9604, #10491)
- Clipboard: disable the clipboard by default for new VMs (see chapitre 13.3.2.3, *Presse-papier*, page 228). It can be enabled at any time using the VM menu.
- Settings: sanitise the name of VM folders and settings file (bug #10549)

- Settings: allow to store the iSCSI initiator secret encrypted
- NAT: improvements for the built-in TFTP server (bugs #7385, #10286)
- NAT: fixed memory leak when disabling the NAT engine (bug #10801)
- E1000: 802.1q VLAN support (bug #10429)
- Storage: implemented burning of audio CDs in passthrough mode
- Storage: fixed audio CD passthrough for certain media players
- Storage: implemented support for discarding unused image blocks through TRIM for SATA and IDE and UNMAP for SCSI when using VDI images
- Storage: added support for QED images
- Storage: added support for QCOW (full support for v1 and readonly support for v2 images)
- Storage: added readonly support for VHDX images
- USB: don't crash if a USB device is plugged or unplugged when saving or loading the VM state (SMP guests only)
- Solaris additions: added support for X.org Server 1.11 and 1.12
- Solaris additions: switched to using an in-kernel mouse driver
- Windows hosts: no need to recreate host-only adapters after a VirtualBox update
- Windows hosts: updated toolchain; make the source code compatible to VC 2010 and enable some security-related compiler options
- Windows Additions: fixed memory leak in VBoxTray (bug #10808)

15.3 Version 4.1.18 (2012-06-06)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed *VERR_REM_VIRTUAL_CPU_ERROR* under rare conditions after the guest has been reset (bug #5164 and others)
- VMM: fixed host freezes with 64-bit guests on 32-bit Linux hosts (bug #10528)
- VRDP: added a workaround for rdesktop clients not properly updating the screen size when minimized
- AHCI: fixed a rare bug which can cause a guest memory corruption after the guest storage controller has been reset
- NAT: another attempt to fix crashes under rare conditions (Windows hosts only; bug #10513)
- Mac OS X hosts: addressed issues running Leopard / Snow Leopard (bug #10631)
- Linux hosts / Bridged Networking: fixed the problem with device driver unloading on kernels 3.2.18 and newer due to an invalid reference counter (bug #10624)
- Linux hosts / guests: Linux 3.5-rc1 fixes
- Linux Additions: the guest content was sometimes not properly updated (bug #9887)
- Solaris Additions: installer fix for X.org Server 1.11 and 1.12

15.4 Version 4.1.16 (2012-05-22)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed a Guru Meditation *VERR_ACCESS_DENIED* with certain guests (bugs #7589, #8247)
- VMM: fixed a Guru Meditation *VERR_PAGE_TABLE_NOT_PRESENT* with Ubuntu 32-bit guests with nested paging enabled on AMD CPUs (bug #10183)
- VMM: preserve segment limits and attributes when switching to unreal mode required for some legacy guests to work properly (VT-x without unrestricted guest execution only; bug #9941)
- VMM: fixed a VM hang after a resume from pause / savestate with SMP guests in rare cases
- 3D Support: several fixes for the Windows WDDM video driver crash
- NAT: fixed a crash on attempt to process ICMP datagram under some circumstances (bug #10527)
- Host-only Networking: lifted the maximal number of interfaces to 128 on Linux and Mac OS X hosts (there is no such limitation for Solaris and Windows hosts)
- EFI: fixed wrong SEC/PEI Core entry point calculation (bug #10531)
- VRDP: fixed a display update problem (bug #10503)
- Main: set the proper VM state if savestate failed for some reason (bug #6189)
- Main: more useful error message if a medium is inaccessible
- VBoxManage: fixed *controlvm savestate* if the VM is already paused
- Mac OS X hosts: addressed issues running on Mountain Lion Preview 3 (bug #10267)
- Linux hosts: Linux 3.4 compile fixes
- Linux hosts: fixed wrong help path in some rpm-based packages (bug #10418)
- Guest Additions: fixed handling of custom environment variables during *VBoxManage guestcontrol execute* (bug #10581)
- Windows Additions: fixed guest driver crash of VBoxSF in certain cases (4.1.10 regression, bug #10408)
- Windows Additions: don't load the WDDM driver if 3D support is not available for Windows 8 guests to keep the guest maintainable in that case (still better to miss some features than providing a blank screen)
- Solaris Additions: added support for X.org Server 1.11 and 1.12

15.5 Version 4.1.14 (2012-04-13)

This is a maintenance release. The following items were fixed and/or added:

- Network: fixed the problem with packets larger than MTU-4 when PCnet or PRO/1000 was bridged to certain types of adapters on OS X hosts (bug #3783)
- NAT: fixed a segfault under rare circumstances

- 3D Support: fixed Windows WDDM video driver crash for SMP guests (bugs #10200, #10331)
- Windows Guest Additions, VRDP: fixed occasional corruption of vertical text

15.6 Version 4.1.12 (2012-04-03)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed *VERR_NOT_SUPPORTED* and *VERR_RAW_MODE_INVALID_SMP* guru meditation due to an invalid reschedule to raw mode (bug #10370)
- VMM: fixed *PDMCritSectLeave* guru meditation under rare circumstances with SMP guests
- VMM: proper *Math Fault* handling with certain legacy guests (bug #9042)
- NAT: fixed a socket leak under certain conditions
- Storage: better sanity check against reading beyond end-of-file
- Audio: fixed a crash in the NUL audio backend (bug #10374; 4.1.10 regression)
- HGCM: fixed a crash during savestate under rare circumstances
- Metrics: fixed an occasional crash during VM shutdown if host RAM/VMM metrics are enabled
- VBoxSVC: several locking fixes
- VBoxManage: return the correct error code if *controlvm savestate* failed (bug #10134)
- Guest Additions: VBoxService should not crash on exit (bug #10326; 4.1.10 regression)
- Windows Additions: set the correct time stamp when a file is closed (bug #6473)
- Windows Additions: better help if the DirectX backups are not properly installed
- Linux Additions: Linux 3.4-rc1 compile fixes

15.7 Version 4.1.10 (2012-03-13)

This is a maintenance release. The following items were fixed and/or added:

- GUI: if 3D support on the host is not available for some reason, do not only disable this VM setting but also uncheck the checkbox
- VMM: fixed a potential problem causing to schedule interrupts during *SYSEXIT* right after *STI*
- VMM: fixed a potential guest memory corruption issue with page fusion
- VMM: adjusted the module matching algorithm for page fusion a little, generally resulting in slightly more shared pages
- Main: host interfaces no longer have “defaults” for IP address and network mask attributes
- Main: don’t depend on a password for certain guest control operations (bug #10030)
- 3D Support: fixed Windows XP hosts support (4.1.8 regression; bugs #10071 and #10088)

- 3D Support: rendering fixes for Linux hosts with NVIDIA graphics
- 3D Support: fixed saved state issues (4.1.8 regression; bug #10126)
- 3D Support: WDDM driver: fixed powershell_ise crashes (bug #10167), make WPF-based apps work with Aero enabled, fixed additional possible WinSAT crashes
- VRDP: fixed remote clipboard compatibility issues with some clients
- Storage: fixed a possible data corruption when compacting VDI or VHD images with snapshots (32-bit hosts only)
- iSCSI: fixed crash when using incorrect credentials when authenticating with a LIO target (bug #10173)
- Serial: don't abort in host mode under rare error conditions (non-Windows hosts only)
- SDK: actually ship current C bindings
- SDK: fixed the Java glue code for JDK 1.7.0 (bug #9848)
- SDK: added Python example
- Metrics: make metrics collection more robust regarding blocked VMs
- Web service: added SSL/TLS support
- VBoxShell: fixed Guest Additions information lookup
- Solaris installer: fixed dependency checking while installing VirtualBox in a zone
- Linux hosts/guests: Linux 3.3-rc1 compile fixes
- Solaris hosts: fixed debug kernel panics while opening module CTF data (bug #9651)
- Mac OS X hosts: fixed Python support on Lion
- Linux Additions: make 3D passthrough work on OL/RHEL 6.2 (bug #10010)
- Linux Additions: fixed missing implementation when copying shared folder data through kernel high memory (bug #9878)
- Linux Additions: make sure all data is written when closing a memory mapped file on a shared folder
- Linux Additions: added support for X.Org Server 1.12
- Solaris Additions: fixed guest kernel driver to load properly on guest reboot (4.1.8 regression; bug #10113)
- Solaris Additions: fixed missing 64-bit OpenGL library VBoxOGL.so (bug #10151)
- Solaris Additions: fixed VBoxService import and start for Solaris 11 guests.
- Windows Additions: some Windows 8 adaptions
- Windows Additions: several fixes for shared folders (bug #9753)
- Guest control: miscellaneous bugfixes

15.8 Version 4.1.8 (2011-12-19)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed *VERR_MAP_FAILED* during savestate under certain circumstances (bug #7929)
- GUI: stop updating the VM status icons when the VM is paused (bug #8368)
- VBoxManage: fixed wrong return code after *startvm* (bug #9642)
- BIOS: fixed hang at launch of DOS applications generated by Clipper 5.3 (note that hardware virtualization may be required)
- USB: fixed OS/2 boot hang when using recent USB drivers
- NAT: increase maximum number of parallel connections making connections with port forwarding more robust (#8471)
- Metrics: fixed potential problem with invalid access in guest metrics collection upon VM destruction
- Main: don't crash if a medium is ejected twice (bug #9922)
- VBoxSVC: fixed crash under rare circumstances (e.g. client crash)
- VRDP: fixed screen freeze (bug #9620)
- OVF/OVA: fixed broken disk images on import under rare circumstances
- OVF/OVA: better error message when importing corrupted appliances
- VMDK/VHD: fixed a possible corruption with host cache disabled when using snapshots under rare circumstances (bug #9749)
- 3D Support: fixed full screen mode issues for ATI graphics (bug #9775), Windows Media Player rendering for XPDM-based Direct3D support (bug# 8341). Multiple fixes to XPDM and WDDM - based 3D support for Windows Guests and for 3D support in general
- Linux hosts: fixes for Fedoras Linux 2.6.41 (bug #9948)
- Linux hosts/guests: fixes for Linux 3.2 (bug #9743)
- Solaris Additions: various shared folder fixes (bugs #9856, #9862, #9917)
- Windows Additions: various fixes for Direct3D support (un)installation, added detection of missing or wrong Direct3D system files

15.9 Version 4.1.6 (2011-11-04)

This is a maintenance release. The following items were fixed and/or added:

- VRDP: fixed screen corruption
- NAT: the interface stopped working after a lot of failed ICMP requests (bug #9371)
- E1000: fixed rare Windows 7 guest hangs, either at boot time or when reconfiguring the network card in unusual setups (bug #6223)
- ATA: fixed a possible crash during ATAPI passthrough with certain guests
- ATA: improved compatibility with ancient Linux kernels

- Main: fixed incorrect framebuffer information after leaving the full screen mode with X11 guests, this lead to a scrambled preview window in the GUI for example
- Mac OS X hosts: fixed the problem with duplicate packets when bridged to a wireless interface (bug #9648)
- Linux hosts: fix for 3D support on Fedora 15 (bug #9799)
- Linux hosts: don't call *del_timer_sync* from an interrupt context (Fedora bug report 746331)
- Windows Vista and 7 guests: more WDDM fixes
- Linux Additions: fixed kernel module compilation failure on Redhat Enterprise Linux 4 (bug #9709)
- Linux Additions: install the DRI driver correctly on Ubuntu 11.10 guests
- Solaris Additions: added read-only mmap support for shared folders
- Solaris Additions: added directory, file mode and mask options for shared folders
- Windows Additions: implemented faster detection of logged-in guest users and stale sessions
- X.Org Additions: fixed graphical corruption when switching to a virtual terminal (bug #9490)

15.10 Version 4.1.4 (2011-10-03)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed PAE guests running on 32-bit hosts (4.0 regression; bug #9458)
- VMM: fixed *INVALID_PARAMETER* guru meditation caused by insufficient memory conditions (4.1.0 regression; bug #9240)
- VMM: fixed clobbered CPU registers during *stos/lods/ins/outs* emulation under rare circumstances
- VMM: another fix for 64-bit guests on recent AMD CPUs
- GUI: warn the user if he uses an outdated extension pack and allow to download and upgrade in that case (bug #8025)
- GUI: fixed spurious LCtrl key events on some Windows hosts (4.0.6 regression; bug #373)
- VBoxManage: another fix for *extpack install -replace* (bug #9308)
- VBoxManage: allow to specify the UUID of the target image in *convertfromraw*
- VRDP: fixed a rare crash when two or more clients connect to the server (bug #9503)
- VRDP: fixed a case when a client was not resized correctly (bug #9242)
- USB: make device capturing work on Windows hosts with usbhub class filter drivers installed (bug #9298); bugfixes for Windows host USB support
- VHD: fixed reads crossing block boundaries (bug #8532)
- VMDK: fixed progress indicator when creating split VMDK images

- Floppy: fixed medium size detection for physical mediums (Windows hosts only)
- Main: fixed VM initialization if a shared folder does not exist (bug #7941)
- Main: fixed possible deadlock between hard disk enumeration and resetting of immutable disks (bug #9549)
- Main: fixed not detaching of virtual storage after a snapshot restore operation (bug #8296)
- Main: always reset modified state after restoring a snapshot (bug #9387)
- Main: fixed writing of changed VM settings to disk when using snapshots (bug #8932)
- Main: fixed taking screen shots (e.g. the preview window) for headless VMs if the Guest Additions are active
- Virtio-net: fixed the problem with UDP packets exceeding MTU (bug #9370)
- 3D Support: fixed incorrect rendering and possible crashes when switching to/from full screen with enabled 2D acceleration
- 3D Support: fixed compiz under ubuntu 9.10
- Guest control: miscellaneous small bugfixes
- Linux / Solaris hosts: don't use hard links in packages (4.1.2 regression; bug #9441)
- Mac OS X hosts: fix installation issues when installed as root (bug #1578)
- Mac OS X hosts: fixed packet capture issues with vboxnetX host-only interfaces (bug #8076)
- Solaris hosts: fixed incoming multicast packets for Crossbow based bridged networking (bug #9532)
- Solaris hosts: fixed starting EFI guests due to missing EFI ROM files (bug #9535)
- Windows hosts installer: fixed *ADDLOCAL* usage on command line (4.1.0 regression; bug #9488)
- Windows hosts installer: fixed dangling shortcut to the .chm help file (bug #9646)
- Windows hosts installer: try to fix installation error 2869 which appeared under certain circumstances (bug #9664)
- Windows hosts: VBoxHeadless should never open a separate console window (bug #3549)
- Guest Additions: fixed hang while waiting for guest execution output (4.1.2 regression; bug #9446)
- Linux Additions: add support for X.Org Server 1.11 (bug #9519)
- Linux Additions: suppress an incorrect error message and allow a shared folder to be mounted in multiple places (bug #9627)
- Linux Additions: start VBoxService correctly on openSUSE 12.1 (bug #6229)
- Linux Additions: properly finish the installation even if the compilation of the DRM module fails (which is not fatal)
- Solaris Additions: fixed memory leaks and panics while mounting (from console) and unmounting shared folders

- Windows Additions: fixed swapped mouse cursor hotspot coordinates (4.1.0 regression; bug #9447)
- Windows Additions: fixed *PAGE_FAULT_IN_NONPAGED_AREA* BSOD in VBoxDisp.dll when running SnippingTool (bug #9508)
- Windows Additions: make image viewer work correctly with Aero enabled (Vista and Windows 7)
- Windows Additions: fixed incorrect window border blurs under Aero with ATI cards (Vista and Windows 7)
- Windows Additions: fixed incorrect rendering when moving 3D applications outside guest desktop and back under Aero (Vista and Windows 7)
- Windows Additions: fixed guest hangs when switching back from full screen text mode applications (bug #9636)

15.11 Version 4.1.2 (2011-08-15)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed 64-bit guests on AMD Fusion CPUs (bug #8824)
- VMM: fixed handling of the *sysenter/sysexit* machine instructions in 64-bit guests on Intel hosts
- GUI: added linked clone support
- GUI: fixed error check when changing the VM controller settings (4.1.0 regression; bug #9246)
- GUI: fixed the inclusion of additional information (license, vendor, ...) on OVF export
- GUI: when taking a snapshot from the VM selector, don't do a live snapshot
- GUI: fixed immediate crash during start on certain environments due to a library version conflict
- Guest control execution: implemented copying single files from guest to the host (directories and filters not supported yet)
- Guest control execution: limited to serve up to 5 guest processes at a time by default
- Main: return the correct RDP default port fixing settings warnings in the GUI if the remote display was enabled at runtime (bug #9368)
- VBoxSVC: made the path comparison OS independent when images are registered (bug #9303)
- VBoxSVC: several fixes/performance improvements for cloning of VMs
- Snapshots: fixed "Solid-State drive" flag survival on snapshot creation (bug #9379)
- VBoxManage: implemented *copyfrom* and *stat* guest control tools to copy files from guest to host and to check for existence of files on the guest
- VBoxManage: fixed *extpack install -replace* (bug #9308)
- VBoxManage: allow to start multiple VMs with *startvm*

- VBoxManage: experimental support for SATA CD/DVD hotplugging
- Mouse support: fixed an off-by-one pointer position error affecting Windows guests (bugs #7566, #9321)
- VRDP: fixed a case when the screen was not redrawn correctly (bug #9242)
- NAT: changed TCP connection establishment according to Winsock API requirements (Windows hosts only; bug #8919)
- USB: fixed device recognition issues (bugs #9299) and fixed several device capturing issues (Windows hosts; bug #9299)
- USB: fixed issues with remote filters being applied to local USB devices
- Guest Additions: keep logged in user count in out-of-memory situations and warn about it
- OVA: fixed import of renamed OVA files (bug #9033)
- Windows host: fixed icon for .VDI files (bug #9393)
- Solaris hosts: fixed a bug preventing parallel usage of multiple host NICs with Crossbow based bridged networking
- Linux hosts: fixed random kernel panics on host suspend / shutdown (4.1.0 regression; bug #9305)
- Solaris Additions: fixed shared folders due to missing symbols in the vboxfs module (4.0.12 / 4.1.0 regression; bug #9264)
- Windows Additions: fixed file truncation on a shared folder with some applications (bugs #9276, #9315)
- Windows Additions: fixed shared folder issues with antivirus software and 64 bit Windows guests (bug #9318)
- Windows Vista and 7 guests: WDDM driver fixes and performance enhancements, fixed WinSAT crashes (#9267)
- Linux Additions: fixed *llseek* for Linux kernels 2.6.37 and above

15.12 Version 4.1.0 (2011-07-19)

This version is a major update. The following major new features were added:

- Support for cloning of VMs (bug #5853, see chapitre 1.13, *Cloner des machines virtuelles*, page 32): full clones can be created through the GUI and VBoxManage, linked clones only through VBoxManage
- GUI: enhanced wizard for creating new virtual disks
- GUI: new wizard for copying virtual disks
- GUI: keep the aspect ratio in scale mode (Windows and OSX hosts only; bug #7822)
- VMM: raised the memory limit for 64-bit hosts to 1TB
- Experimental support for PCI passthrough for Linux hosts, see chapitre 9.6, *PCI passthrough*, page 171

- Windows guests: Experimental WDDM graphics driver, supporting Windows Aero (bug #4607) and providing Direct3D support using a cleaner approach (no need to install the guest drivers in Safe Mode anymore)
- Guest Additions: status of modules and features can now be queried separately by the frontends
- Networking: new network attachment mode “Generic Driver”, which offers an open plugin architecture for arbitrary and separately distributable virtual network implementations
- Host-only Networking: fixed host crash in kernels prior to 2.6.29
- New Networking Mode *UDP Tunnel*: allows to interconnect VMs running on different hosts easily and transparently, see chapitre [6.2, Introduction aux modes réseaux](#), page [99](#)
- Experimental support for SATA hard disk hotplugging available with VBoxManage
- Solaris hosts: New Crossbow based bridged networking driver for Solaris 11 build 159 and above

In addition, the following items were fixed and/or added:

- VMM: more SMP timer fixes
- VMM: fixed sporadic recompiler crashes with SMP guests
- VMM: many small fixes
- GUI: when restoring a snapshot, ask for taking a snapshot of the current state
- GUI: added a *View* menu
- GUI: added a setting for the promiscuous mode policy for internal networks, bridged networks and host-only networks
- GUI: added slider for setting the CPU execution cap allowing to limit the amount of CPU time spent for the execution of the guest, see chapitre [3.4.2, Onglet “Processeur”](#), page [54](#)
- GUI: the VM description is editable during the runtime of a VM (bug #1551)
- GUI: added proxy settings (bug #2870)
- GUI: made the number of SATA ports configurable
- GUI: decrease time before showing the VM configuration dialog
- VBoxManage: more convenient configuration of storage controller attachments by automatically determining the port or device parameter when a storage controller has only one port or device per port
- VBoxManage: changed syntax of the *guestcontrol* command group, fixed various bugs, removed obsolete options
- VBoxBalloonCtrl: new service for automatic dynamic adjustment of the balloon size for running VMs
- Settings: machine names and snapshot names are not allowed to be a valid UUID
- Settings: provide better diagnostics if a single medium is used twice in a VM configuration
- Settings: provide better diagnostics for errors in medium create/merge/clone operations, and fix memory leaks in error cases

- Storage: ATA/SATA drives can be marked as non-rotational, i.e. the guest OS will detect them as a SSD if supported, which can improve performance
- Storage: virtual CD/DVD images will be detached if the guest ejects the medium, unless the drive is marked to handle ejects only on a temporary basis
- Storage: the medium UUID can be changed again when attaching a medium for the first time, which allows using images which are exact duplicates including the UUID
- Storage: fixed possible data corruption under certain circumstances with VHD and Parallels images (bug #9150)
- Storage: fixed unnecessary expansion when cloning differential images in VDI format
- Storage: fixed detection code to handle empty files for VDI and VMDK format
- Storage: fixed access to CD/DVD images beyond 4GB when using the SATA controller (bug #8592)
- Floppy: several bugs have been fixed
- Floppy: make it possible to unmount a host floppy disk (bug #6651)
- BIOS: disk-related structures are now checksummed correctly (bug #8739)
- USB: many fixes for the Windows USB host driver
- NAT: reduced memory footprint
- Networking: fixed the problem with segmentation offloading when several VMs are transmitting in parallel
- Networking: workaround for a bug in wireshark when operating directly on a capture file created by VirtualBox
- Serial: announce the serial devices in the ACPI tables to make Windows guests find the virtual hardware (bug #7411)
- VRDP: support for TLS connections (see chapitre 7.1.6, *Chiffrement RDP*, page 115)
- VRDP: support for multimonitor client configurations with MS RDP clients
- VRDP: fixed a rare screen corruption
- 3D support: fixed *GL_VERSION* string for different locales (bug #8916)
- Web service: fixed timeout handling with HTTP 1.1 keepalive, and be more robust when connections fail
- VBoxSVC: fixed regression when several clients trigger autostart simultaneously
- Main: fixed incorrect handling of the medium location for media which are not file based (e.g. iSCSI), which resulted in confusing location values in many places
- JAX-WS client bindings: fixed resource leak
- Sources: fixed USB 2.0 support using extension packs for non-official builds
- Mac OS X hosts: fixed non-VT-x mode on Lion hosts
- Windows hosts: fixed copy'n'paste in the GUI and for the VM window (bug #4491)
- Windows hosts (64-bit only): enabled removing of all mediums when removing a VM

- Windows hosts (64-bit only): enabled live snapshot deletion
- Windows hosts: use native controls in the installer (bug #5520)
- Solaris hosts: fixed preemption issue with Solaris 11 hosts (builds 166 and above)
- Solaris hosts: better control of USB device access on Solaris 11 hosts
- Guest Additions: improved driver installation on Windows guests
- Guest Additions: fixed high CPU usage while executing guest programs from the host
- Solaris Additions: fixed automounting of shared folders (bug #8014)

15.13 Version 4.0.14 (2011-10-13)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed 64bit guests on AMD Fusion CPUs (bug #8824)
- VMM: fixed handling of the *sysenter/sysexit* machine instructions in 64-bit guests on Intel hosts
- GUI: fixed the inclusion of additional information (license, vendor, ...) on OVF export
- GUI: when taking a snapshot from the VM selector, don't do a live snapshot
- GUI: fixed spurious LCtrl key events on some Windows hosts (4.0.6 regression; bug #5908)
- VBoxManage: fixed *extpack install –replace*
- VRDP: fixed a rare screen corruption
- E1000: fixed rare Windows 7 guest hangs, either at boot time or when reconfiguring the network card in unusual setups (bug #6223)
- Mouse support: fixed an off-by-one pointer position error affecting Windows guests (bugs #7566, #9321)
- NAT: changed TCP connection establishment according to Winsock API requirements (Windows hosts only; bug #8919)
- VHD: fixed reads crossing block boundaries (bug #8532)
- VMDK: fixed progress indicator when creating split VMDK images
- Floppy: fixed medium size detection for physical mediums (Windows hosts only)
- VBoxSVC: made the path comparison OS independent when images are registered (bug #9303)
- Main: return the correct RDP default port fixing settings warnings in the GUI if the remote display was enabled at runtime (bug #9368)
- Main: fix VM initialization if a shared folder does not exist (bug #7941)
- Main: fixed taking screen shots (e.g. the preview window) for headless VMs if the Guest Additions are active
- Main: fixed possible deadlock between hard disk enumeration and the resetting of immutable disks (bug #9549)

- OVA: fixed import of renamed OVA files (bug #9033)
- Virtio-net: fixed the problem with UDP packets exceeding MTU (bug #9370)
- Solaris hosts: fixed starting EFI guests due to missing EFI ROM files (bug #9535)
- Mac OS X hosts: fix installation issues when installed as root (bug #1578)
- Windows Additions: fixed file truncation on a shared folder with some applications (bugs #9276, #9315)
- Linux Additions: suppress an incorrect error message and allow a shared folder to be mounted in multiple places (bug #9627)
- Linux Additions: add support for X.Org Server 1.11 (bug #9519)
- Linux Additions: fixed *lseek* for Linux kernels 2.6.37 and above
- Linux Additions: start VBoxService correctly on openSUSE 12.1 (bug #6229)
- Linux Additions: properly finish the installation even if the compilation of the DRM module fails (which is not fatal)

15.14 Version 4.0.12 (2011-07-15)

This is a maintenance release. The following items were fixed and/or added:

- Mac OS X hosts: Lion fixes
- Solaris hosts: fixed preemption issue with Solaris 11 hosts (builds 166 and above)
- VBoxManage: more convenient configuration of storage controller attachments by automatically determining the port or device parameter when a storage controller has only one port or device per port (bug #9188)
- Storage: fixed possible data corruption under certain circumstances with VHD and Parallels images (bug #9150)
- Storage: fixed access to CD/DVD images beyond 4GB when using the SATA controller (bug #8592)
- Floppy: make it possible to unmount a host floppy disk (bug #6651)
- Networking: fixed the problem with segmentation offloading when several VMs were transmitting at once
- 3D support: fixed *GL_VERSION* string for different locales (bug #8916)
- Sources: fixed USB 2.0 support using extension packs for non-official builds
- Solaris Additions: fixed automounting of shared folders (bug #8014)

15.15 Version 4.0.10 (2011-06-22)

This is a maintenance release. The following items were fixed and/or added:

- GUI: fixed disappearing settings widgets on KDE hosts (bug #6809)
- Storage: fixed hang under rare circumstances with flat VMDK images
- Storage: a saved VM could not be restored under certain circumstances after the host kernel was updated (bug #8983)
- Storage: refuse to create a medium with an invalid variant (for example Split2G with VDI; bug #7227)
- iSCSI: pause the VM if a request times out
- Snapshots: none of the hard disk attachments must be attached to another VM in normal mode when creating a snapshot
- USB: fixed occasional VM hangs with SMP guests (bug #4580)
- USB: proper device detection on RHEL/OEL/CentOS 5 guests (partial fix for bug #8978)
- ACPI: force the ACPI timer to return monotonic values for improve behavior with SMP Linux guests (bug #8511 and others)
- VRDP: fixed screen corruption under rare circumstances (bug #8977)
- rdesktop-vrdp: updated to version 1.7.0
- OVF: under rare circumstances some data at the end of a VMDK file was not written during export
- Mac OS X hosts: Lion fixes
- Mac OS X hosts: GNOME 3 fix
- Linux hosts: fixed VT-x detection on Linux 3.0 hosts (bug #9071)
- Linux hosts: fixed Python 2.7 bindings in the universal Linux binaries
- Windows hosts: fixed leak of thread and process handles
- Windows Additions: fixed bug when determining the extended version of the Guest Additions (4.0.8 regression; bug #8948)
- Solaris Additions: fixed installation to 64-bit Solaris 10u9 guests (4.0.8 regression)
- Linux Additions: RHEL6.1/OL6.1 compile fix
- Linux Additions: fixed a memory leak during `VBoxManage guestcontrol execute` (bug #9068)

15.16 Version 4.0.8 (2011-05-16)

This is a maintenance release. The following items were fixed and/or added:

- Mac OS X hosts: fixed incompatibility with recent Mac OS X versions in 64-bit mode (bug #8474)
- Mac OS X hosts: fixed incompatibility with hosts with more than 16 cores (bug #8389)
- Mac OS X hosts: fixed painting corruptions on a second monitor in 64-bit mode (bug #7606)
- GUI: restored functionality to set an empty host key to disallow any host key combination (4.0.6 regression; bug #8793)
- GUI: more expressive error messages for USB proxy permission problems (mainly Linux hosts; bug #8823)
- VBoxManage: added `controlvm screenshotpng` subcommand for saving the screenshot of a running VM in PNG format
- VBoxHeadless: fixed potential crash during shutdown (Windows hosts only)
- NAT: built-in services use the correct Ethernet addresses in Ethernet header and in ARP requests
- Host-only networking: fixed adapter reference counting
- E1000: fixed rare guest crashes with Linux SMP guests (bug #8755)
- SATA: fixed guest disk corruption under rare circumstances (only relevant for guests with more than 2GB RAM; bug #8826)
- Storage: fixed data corruption after a snapshot was taken with asynchronous I/O enabled (bug #8498)
- Floppy: several improvement
- HPET: another fix for time jumps (bug #8707)
- USB: use correct permissions when creating `/dev/vboxusb` (Linux hosts only)
- USB: removed assumption that string descriptors are null-terminated (Windows hosts only)
- 3D support: fixed a potential crash when resizing the guest window
- 3D support: fixed GNOME 3 rendering under Ubuntu 11.04 and Fedora 15
- Snapshots: fixed another bug which could lose entries in the media registry when restoring a snapshot (bug #8363)
- Shared Folders: don't stop mounting the other valid folders if one host folder is inaccessible (4.0.6 regression)
- Linux Additions: check whether `gcc` and `make` are installed before building kernel modules (bug #8795)
- Solaris Additions: added support for X.Org Server 1.10
- Guest Additions: fixed inappropriate Guest Additions update notification when using vendor-specific version suffixes (bug #8844)

15.17 Version 4.0.6 (2011-04-21)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed incorrect handling of ballooned pages when restoring a VMM from a saved state
- VMM: don't crash on hosts with more than 64 cores / hyperthreads; implemented support for up to 256 host cores (except Windows hosts; bug #8489)
- VMM: fixed guru meditation for PAE guests running on hosts without PAE (bug #8006)
- VMM: fixed slow Linux guests with raw mode and recent guest kernels (bug #8726)
- GUI: support host key combinations (bug #979)
- GUI: fixed progress indicator (bug #7814)
- GUI: show the mouse pointer while the VM is paused if the USB tablet mouse emulation is used (bug #6799)
- GUI: adapt the snapshot folder as well when renaming a VM (bug #8469)
- GUI: persistently remember the last folders of the disk/DVD/floppy selectors
- GUI: never allow to start a VM with USB-2.0 activated if the proper extension pack is missing (bug #8182)
- GUI: fixed hang/crash when opening a file dialog in an non-existing folder (bug #8673)
- Snapshots: fixed a bug which could lose entries in the media registry when restoring a snapshot (bug #8363)
- Snapshots: allow snapshots to be stored in the VM directory
- 3D support: fixed a crash if a VM was forced to terminate (Windows hosts only; bug #7133)
- Storage: fixed memory leak (4.0 regression; bug #7966)
- Storage: fixed access to iSCSI targets over internal network
- Storage: fixed reading from disks with more than one snapshot for VHD and VMDK images with disabled host cache (bug #8408)
- Storage: fixed a possible hang during VM suspend after an I/O error occurred
- Storage: fixed a possible hang during VM suspend / reset (bug #8276, #8294)
- Storage: automatically create a diff image when attaching a streamOptimized VMDK image to a VM
- ATA/SATA: fixed automounting of virtual CD/DVD mediums with recent Linux distributions by correctly reporting the current profile as 'none' if no medium is present
- Buslogic: fixed emulation for certain guests (e.g. jRockit VE)
- Host-Only Networking: fixed interface creation failure on Windows hosts (4.0.4 regression; bug #8362)
- Host-Only & Bridged & Internal Networking: fix for processing promiscuous mode requests by VMs, defaulting to switch behaviour

- Host-Only Networking: fixed connectivity issue after resuming the host from sleep (bug #3625)
- Bridged Networking: support for interface bonding on Mac OS X hosts (bug #8731)
- NAT: fixed processing of ARP announcements for guests with static assigned IPs (bug #8609)
- VRDP: backward compatibility with VRDPAuth external authentication library (bug #8063)
- Shared Folders: don't fail to start a VM if a path is not absolute, for example when importing an OVF from a different host (bug #7941)
- Audio: fixed crash under certain conditions (bug #8527)
- USB: fixed a crash when plugging certain USB devices (bug #8699)
- HPET: fixed time jumps when reading the counter (bug #8707)
- OVF/OVA: automatically adjust disk paths if the VM name is changed on import
- OVF/OVA: fix export to slow medias
- OVF/OVA: automatically repair inconsistent appliances with multiple disks (bug #8253)
- rdesktop-vrdp: fixed an assertion triggered under certain conditions (bug #8593)
- Windows hosts: fixed occasional hangs during VM shutdown because sometimes COM was not properly uninitialized
- Mac OS X hosts: prevent the mouse from leaving the VM window while captured
- Mac OS X hosts: keep aspect ratio while resizing in scale mode (shift for old behaviour) (part of bug #7822)
- X11 hosts: fixed Yen key support (bug #8438)
- X11 hosts: fixed a regression which caused Host+F1 to pop up help instead of sending Ctrl+Alt+F1
- Linux hosts / Linux Additions: mangle IPRT symbols to allow installing VirtualBox inside a VM while the Guest Additions are active (bug #5686)
- Linux hosts / Linux guests: workaround for a bug in GLIBC older than version 1.11 leading to crashes under certain conditions (signed/unsigned problem with memchr on 64-bit machines)
- Solaris hosts: fixed a deadlock in event semaphores that could lead to unkillable VM processes
- Windows Additions: fixed Sysprep parameter handling
- Windows Additions: fixed spontaneous guest reboots under certain circumstances (4.0.2 regression; bugs #8406, #8429)
- Windows Additions: added auto logon support for locked workstations on legacy Windows versions
- Windows Additions: fixed driver bugcheck error when handling PnP messages (4.0 regression; bug #8367)
- Windows Additions: fixed memory leak in VBoxVideo

- X11 Additions: added support for X.Org Server 1.10 final
- Linux Additions: Linux kernel 2.6.39-rc1 fixes
- Linux Additions: improved auto-run support (bug #5509)
- Linux Additions: fix mouse support on SUSE 11 SP 1 guests (bug #7946)
- Solaris Additions: added support for X.Org Server 1.9
- Guest Additions: various bugfixes for guest control execution
- Web service: use own log file, with log rotation to limit size

15.18 Version 4.0.4 (2011-02-17)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed recompiler crashes under certain conditions (bugs #8255, #8319 and further)
- VMM: fixed running 64-bit guests on 32-bit host with nested paging enabled on AMD CPUs (4.0 regression; bug #7938)
- VMM: fixed timing issues / hangs for certain guests using the programmable interval timer (bugs #8033 and #8062)
- VMM: large page and monitoring fixes for live snapshots (bugs #7910, #8059, #8125)
- GUI: fixed error message when trying to exceed the maximum number of host network interfaces
- GUI: fixed saving of changes to the metadata of an existing snapshot (bug #8145)
- GUI: fixed rare crash on X11 hosts (bug #8131)
- GUI: when selecting a shared folder, start the file dialog in the users home directory (bug #8017)
- ExtPack: enforce the correct permissions which might be restricted by umask when creating directories (non-Windows hosts only; bug #7878)
- VBoxSDL: fixed crash when starting by specifying the VM UUID (4.0 regression; bug #8342)
- VBoxManage: allow savestate even if the VM is already paused
- VBoxManage: fixed *modifyvm --synthcpu* (bug #6577)
- VBoxManage: fixed hang when doing *guestcontrol execute --wait-for exit* and displaying process status on exit (bug #8235)
- VBoxManage: decreased CPU load during *guestcontrol execute --wait-for exit/stdout* while waiting for the guest process to terminate (bug #7872)
- VBoxManage: fixed *list hostdvds/hostfloppies*
- VBoxManage: fixed *storageattach* for host DVD drives and host floppy drives
- Metrics: introduced RAM/VMM base metric
- Main: improved sanity check when taking a VM screen shot (bug #7966)

- Main: fixed a crash under rare circumstances if a VM failed to start
- Main: fixed attaching of immutable disk images (bug #8105)
- Main: fixed a crash at VM shutdown (bug #6443)
- Main: fixed incorrect handling of cross-referenced medium attachments (bug #8129)
- Settings: fixed truncating of big integer values (4.0 regression)
- Settings: properly store the ICH9 chipset type (bug #8123)
- Host-Only & Bridged Networking: fixed VBox DHCP server startup issue for Windows hosts (4.0 regression; bug #7905)
- Host-Only Networking: re-create vboxnetX interfaces after vboxnetadp.ko module reload on Linux and Darwin (bugs #5934, #6341)
- NAT: fixed an mbuf leak under rare circumstances (bug #7459)
- ACPI: don't allow the guest to enter S4 by default and don't announce S1 and S4 in the ACPI tables if disabled (bug #8008)
- Graphics card: made re-enabling disabled screens work correctly to prevent problems when X11 guests enter screen saving mode (bug #8122)
- Storage: fixed write errors with snapshots if the host cache is disabled (4.0 regression; bug #8221)
- ATA/SATA: fixed reset handling after ACPI suspend/resume
- BusLogic: fixed hang with SMP VMs
- Serial: another attempt to prevent lost characters during transmission (bug #1548)
- Linux hosts/guests: Linux 2.6.38-rc1 compile fixes
- Mac OS X hosts: fixed VBoxSVC crash when listing host interfaces without default gateway (64-bit hosts only; bug #7955)
- Solaris/Darwin hosts: fixed VM CPU execution cap
- X.Org guests: fixed a crash on X server restart (bug #8231)
- X.Org guests: support X.Org Server 1.10 pre-release and Ubuntu 11.04 Alpha
- X.Org guests: Add EDID emulation in the graphics driver to prevent GNOME settings daemon changing the mode on login
- X.Org guests: never send graphics modes to the host that older VirtualBox versions can't handle
- Linux Additions: fixed a memory leak in the shared folders code if a host link is not readable (bug #8185)
- Windows Additions: fixed handling of Security Attention Sequence (SAS) with VBoxGINA

15.19 Version 4.0.2 (2011-01-18)

This is a maintenance release. The following items were fixed and/or added:

- GUI: don't crash if a removable host drive referenced from the VM settings vanished
- GUI: fixed a crash when using the KDE4 Oxygen theme and clicked on the settings button (4.0 regression; bug #7875)
- GUI: properly warn if the machine folder cannot be created (bug #8031)
- GUI: several fixes for multimonitor X11 guests
- ExtPack: don't make the installer helper application suid root (Linux .deb/.rpm packages only)
- ExtPack: improved user experience on Vista / Windows 7 when installing an extension pack
- ExtPack: fixed issue with non-ascii characters in the path name during installing an extension pack (bug #9717)
- ExtPack: fixed SELinux issues on 32-bit Linux hosts
- VBoxManage: Host-only interface creation and removal is now supported for all platforms except Solaris (bug #7741)
- VBoxManage: fixed segmentation fault when removing non-existent host-only interface
- Storage: fixed possible crashes with VMDK/VHD images with snapshots and asynchronous I/O (4.0 regression)
- Storage: don't eject the physical medium if a DVD/CDROM/floppy drive is detached from a VM (bug #5825)
- Storage: be more robust when a faulty guest sends ATA commands to an ATAPI device (bug #6597)
- Parallels: fixed deletion of the image during suspend, pause or power off (4.0 regression)
- Bridged networking: fixed host kernel panic when bridging to devices with no TX queue (4.0 regression; Linux hosts only; bug #7908)
- NAT: port-forwarding rule registration respects protocol parameter (bug #8094)
- E1000: fixed PXE boot issues with WDS (bug #6330)
- Virtio-net: fixed the issue with TX performance in some Linux guests
- ICH9: fixed VM crash (software virtualization only; bug #7885)
- VGA: fixed VESA screen issue (4.0 regression; bug #7986)
- Shared Folders: fixed parameter parsing when creating symbolic links, fixes 32-bit/64-bit bitness issue (bug #818)
- Main: fixed crash under rare circumstances due to an invalid logging string (4.0 regression)
- Main: improve error information propagation for errors preventing a VM start
- Main: fixed problems with snapshots and non-ASCII characters in machine paths (bug #8024)

- Web service: now listens to localhost by default as documented (bug #6067)
- Settings: do not fail loading machine settings if removable drive attachment (host drive or image) cannot be found; with 4.0 this is much more likely when machines are moved from one host to another
- Settings: fixed issue that changing a snapshot name or description was not saved to machine XML
- OVF/OVA: fixed import of files created by other OVF tools (bug #7983)
- rdesktop-vrdp: fix a crash during USB device enumeration (bug #7981)
- Linux hosts: fixed a crash during USB device enumeration
- Linux hosts: try a bit harder to allocate memory (bug #8035; 4.0 regression)
- Guest Additions: fixed parsing of parameters for guest control in VBoxService (4.0 regression; bug #8010)
- Windows Guest Additions: automatic logon on Windows Vista/Windows 7 now supports unlocking previously locked workstations

15.20 Version 4.0.0 (2010-12-22)

This version is a major update. The following major new features were added:

- Reorganization of VirtualBox into a base package and Extension Packs; see chapitre 1.5, [Installer VirtualBox et les packs d'extension](#), page 17
- New settings/disk file layout for VM portability; see chapitre 10.1, [Où VirtualBox stocke ses fichiers](#), page 196
- Major rework of the GUI (now called “VirtualBox Manager”):
 - Redesigned user interface with guest window preview (also for screenshots)
 - New “scale” display mode with scaled guest display; see chapitre 1.8.5, [Redimensionner la fenêtre de la machine](#), page 25
 - Support for creating and starting .vbox desktop shortcuts (bug #1889)
 - The VM list is now sortable
 - Machines can now be deleted easily without a trace including snapshots and saved states, and optionally including attached disk images (bug #5511; also, VBoxManage unregistervm --delete can do the same now)
 - Built-in creation of desktop file shortcuts to start VMs on double click (bug #2322)
- VMM: support more than 1.5/2 GB guest RAM on 32-bit hosts
- New virtual hardware:
 - Intel ICH9 chipset with three PCI buses, PCI Express and Message Signaled Interrupts (MSI); see chapitre 3.4.1, [Onglet “Carte mère”](#), page 52
 - Intel HD Audio, for better support of modern guest operating systems (e.g. 64-bit Windows; bug #2785)
- Improvements to OVF support (see chapitre 1.14, [Importer et exporter des machines virtuelles](#), page 33):

- Open Virtualization Format Archive (OVA) support
- Significant performance improvements during export and import
- Creation of the manifest file on export is optional now
- Imported disks can have formats other than VMDK
- Resource control: added support for limiting a VM's CPU time and IO bandwidth; see chapitre 5.8, *Limiter la bande passante des images de disque*, page 96
- Storage: support asynchronous I/O for iSCSI, VMDK, VHD and Parallels images
- Storage: support for resizing VDI and VHD images; see chapitre 8.23, *VBoxManage modifyhd*, page 147
- Guest Additions: support for multiple virtual screens in Linux and Solaris guests using X.Org server 1.3 and later
- Language bindings: uniform Java bindings for both local (COM/XPCOM) and remote (SOAP) invocation APIs

In addition, the following items were fixed and/or added:

- VMM: Enable large page support by default on 64-bit hosts (applies to nested paging only)
- VMM: fixed guru meditation when running Minix (VT-x only; bug #6557)
- VMM: fixed crash under certain circumstances (Linux hosts only, non VT-x/AMD-V mode only; bugs #4529 and #7819)
- GUI: add configuration dialog for port forwarding in NAT mode (bug #1657)
- GUI: show the guest window content on save and restore
- GUI: certain GUI warnings don't stop the VM output anymore
- GUI: fixed black full screen minitoolbar on KDE4 hosts (Linux hosts only; bug #5449)
- BIOS: implemented multi-sector reading to speed up booting of certain guests (e.g. Solaris)
- Bridged networking: improved throughput by filtering out outgoing packets intended for the host before they reach the physical network (Linux hosts only; bug #7792)
- 3D support: allow use of *CR_SYSTEM_GL_PATH* again (bug #6864)
- 3D support: fixed various clipping/visibility issues (bugs #5659, #5794, #5848, #6018, #6187, #6570)
- 3D support: guest application stack corruption when using *glGetVertexAttrib[ifd]v* (bug #7395)
- 3D support: fixed OpenGL support for libMesa 7.9
- 3D support: fixed Unity/Compiz crashes on natty
- 2D Video acceleration: multimonitor support
- VRDP: fixed rare crash in multimonitor configuration
- VRDP: support for upstream audio
- Display: fixed occasional guest resize crash

- NAT: port forwarding rules can be applied at runtime
- SATA: allow to attach CD/DVD-ROM drives including passthrough (bug #7058)
- Floppy: support readonly image files, taking this as the criteria for making the medium readonly (bug #5651)
- Audio: fixed memory corruption during playback under rare circumstances
- Audio: the DirectSound backend now allows VMs to be audible when another DirectSound application is active, including another VM (bug #5578)
- EFI: support for SATA disks and CDROMs
- BIOS: reduce the stack usage of the VESA BIOS function #4F01 (Quake fix)
- OVF/OVA: fixed export of VMs with iSCSI disks
- Storage: Apple DMG image support for the virtual CD/DVD (bug #6760)
- Linux host USB support: introduced a less invasive way of accessing raw USB devices (bugs #1093, #5345, #7759)
- Linux hosts: support recent Linux kernels with *CONFIG_DEBUG_SET_MODULE_RONX* set
- Guest Additions: Shared Folders now can be marked as being auto-mounted on Windows, Linux and Solaris guests
- Linux Additions: Shared Folders now support symbolic links (bug #818)
- Linux Additions: combined 32-bit and 64-bit additions into one file
- Windows Additions: automatic logon on Windows Vista/Windows 7 is now able to handle renamed user accounts; added various bugfixes

15.21 Version 3.2.12 (2010-11-30)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed rare host crash when running 64-bit guests on 32-bit hosts (bug #7577)
- VMM: fixed host reboots under rare circumstances due to NMIs triggered by active performance counters (Linux hosts in non-VT-x/AMD-V mode only; bug #4529)
- VMM: fixed out of memory guru meditation for large memory guests (bug #7586)
- VMM: fixed a guru meditation related to large pages
- VMM: use new VT-x feature to keep the guest from hogging the CPU
- Snapshots: implemented deleting the last remaining snapshot while the VM is running
- GUI: perform the checks for exceeding the size limit of the host file system and for broken asynchronous I/O on older Linux kernels with ext4 / xfs file systems not only when starting the VM from scratch but also when starting from a saved state
- NAT: fixed memory leak (3.2.0 regression; bugs #6918, #7353)
- NAT: fixed Linux NFS root issue (bugs #7299)
- Networking: fixed VM reset handling in e1000

- VRDP: fixed rare crash in multimonitor configuration
- Display: fixed occasional guest resize crash
- Mouse: don't send relative mouse events together with absolute mouse events (3.2.10 regression; bug #7571)
- Keyboard: fixes for the USB keyboard emulation; fixes for Korean keyboards
- Serial: don't hang if the host device would block during open (bugs #5756, #5380)
- Serial: fixed modem status lines (Linux hosts only; bug #812)
- Graphics: Horizontal resolutions are no longer restricted to a multiple of 8 pixels (bug #2047; requires Guest Additions update).
- USB: fixed a crash with older Linux kernels and non-ASCII characters in device strings (Linux hosts only; bug #6983, #7158, #7733; version 3.2.8 contained an incomplete fix)
- USB: fixed a crash under rare circumstances (bug #7409; Windows hosts only)
- “iSCSI: respond to NOP-In requests from the target immediately to avoid being disconnected if the guest is idle
- 3D support: fixed a crash under certain circumstances (bug #7659)
- 3D support: fixed crashes for GLUT based apps (bug #6848)
- 3D support: added missing GLX 1.3 functionality (bugs #7652, #7195)
- 2D Video acceleration: fixed potential deadlock when saving the VM state (bug #4124)
- Windows hosts: another fix for BSODs under certain circumstances in VBoxNetFlt.sys (bug #7601)
- Solaris hosts: fixed host USB DVD drive detection
- Mac OS X hosts: fixed swapped keys for certain ISO keyboard types (bug #2996)
- Linux hosts: added link state handling for TAP devices needed for proper operation with bridged networking on kernels 2.6.36 and above (bug #7649)
- Linux hosts/guests: Linux 2.6.37 fixes
- Linux Additions: properly compile the vboxvideo module if DKMS is not installed (bug #7572)
- Linux Additions: fixed a memory leak when accessing non-existing files on a Shared Folders (bug #7705)
- Windows Additions: skip none-mapped user accounts when enumerating user accounts for VM information

15.22 Version 3.2.10 (2010-10-08)

This is a maintenance release. The following items were fixed and/or added:

- VMM: V8086 mode fix for legacy DOS/Windows guests with EMM386 (3.2.8 regression)
- VMM: stability fix (bug #7342)
- VMM: fixed a Guru meditation related to large pages (bug #7300)
- VMM: fixed support for large pages on Linux hosts
- VMM: fixed a Guru meditation for large memory 64-bit guests on 32-bit hosts with nested paging (bug #7544)
- VMM: performance improvements for VMs with more than 2GB RAM (bug #6928)
- GUI: fixed host key handling if the host key is set to Left Alt (Linux/Solaris hosts only; 3.2.0 regression; bug #6758)
- GUI: the VM can be minimized from the mini toolbar (bug #4952)
- GUI: handle Ctrl+Break properly on X11 hosts (3.2.0 regression; bug #6122)
- GUI: fixed the case where the user aborted the media selector for selecting the boot hard disk from the VM wizard
- GUI: added a check for Linux kernels 2.6.36 or later which are known to have the asynchronous I/O bug on ext4 / xfs file systems fixed (Linux hosts only)
- OpenSolaris guests: use SATA controller by default
- Storage: fixed I/O errors in the guest after compacting VDI images (3.2.6 regression; bug #7294)
- Storage: automatically repair base disk images with non-zero parent UUID which made them inaccessible (bug #7289)
- Storage: fixed corrupted images if a merge operation was canceled
- IDE: added ATAPI passthrough support for audio CDs
- SATA: fixed a potential hang during boot of recent Solaris guests
- SATA: handle out of disk space and similar conditions better
- iSCSI: fixed sporadic hangs when closing the connection
- VGA: fixed missing redraw with multiple screens under certain circumstances (bug #7291)
- VGA: several small fixes for legacy VGA graphics modes
- Bridged networking: fixed occasional host freeze during VM shutdown (Linux hosts only)
- NAT: don't check for the existence of the TFTP prefix when delivering a file via bootp (bug #7384)
- NAT: fixed resolving of names at the host resolver (bug #7138)
- NAT: under rare conditions the NAT engine consumed 100% CPU load (non-Windows hosts only)
- VRDP: fixed memory leak under certain circumstances (bug #5966)

- VRDP: fixed missing redraws with Windows guests under certain circumstances
- USB: properly discard blocking outstanding bulk URBs, fixes some printers
- USB: Blackberry fix (bug #6465)
- VBoxHeadless: fixed event queue processing problems which led to hangs if the VM could not be started successfully
- VBoxManage: don't crash if parameters with invalid characters are passed (bug #7388)
- VBoxManage: clonehd: fixed a bug where the command aborted with an error message under rare circumstances
- VBoxManage metrics: made it work for directly started VMs again (3.2.8 regression; bug #7482)
- 3D support: report *GLX_ARB_get_proc_address* as supported extension
- 3D support: guest application stack corruption when using *glGetVertexAttrib[ifd]v* (bug #7395)
- 3D support: fixed broken 3D support when switching to full screen / seamless modes (bug #7314)
- 3D support: fixed 32bit OpenGL apps under 64bit Windows XP/Vista (bug #7066)
- OVF: fixed bug when exporting a VM with multiple attached disks (bug #7366)
- OVF: fixed slow export for certain filesystems (bug #3719)
- OVF: disabled manifest (.mf file) support; manifests are no longer verified on import nor written on export
- Shared clipboard/Windows: improved the reliability of the shared clipboard on Windows hosts and guest (partial fix to bug #5266)
- Shared Folders: don't show an empty directory if filenames with an invalid encoding exist on the host (bug #7349)
- Shared Folders: return the proper error code when trying to list files for a non-existing wildcard (bug #7004)
- Audio: fixed guest memory corruption when capturing from the NULL audio backend (bug #6911)
- Audio: improved playback quality (less choppy)
- Web service: avoid unnecessary creation of idle threads
- Additions: fixed bug in the guest execution feature when passing more than one environment variable
- Additions: refresh all guest properties written by VBoxService after the VM was restored from a saved state
- Additions: fixed a *division by zero* crash of VBoxService under certain circumstances
- Additions: immediately resynchronize the guest time with the host time after the VM was restored from a saved state (bug #4018)
- Additions/Windows: fixed *LsaEnumerate* error when enumerating logged in users

- Additions/X.Org: support X.Org Server 1.9 (bug #7306)
- Additions/X.Org: don't crash VBoxClient during reboot
- Solaris hosts: fixed host DVD drive enumeration on Solaris 10
- Solaris hosts: added a custom core dumper to procure more data in the event of a VM crash
- Solaris guests: fixed user idle detection
- Solaris guests: fixed a possible panic in Shared Folders when using the wrong user or group IDs (bug #7295)
- Solaris guests: fixed Shared Folders from truncating files to 2GB on 32-bit guests (bug #7324)
- Windows hosts: fixed a BSOD under certain circumstances in VBoxNetFlt.sys (bug #7448)
- Linux hosts/guests: Linux 2.6.36 fixes
- Linux hosts/guests: DKMS fixes (bug #5817)
- Mac OS X hosts: fixed missing dock menu entries (bug #7392)

15.23 Version 3.2.8 (2010-08-05)

This is a maintenance release. The following items were fixed and/or added:

- VMM: properly terminate the VM with an error if the guest is trying to switch to the PAE mode but PAE is disabled in the VM settings
- GUI: switch to native file dialogs (Windows hosts only; bug #5459)
- GUI: don't use native file dialogs on KDE hosts (Linux hosts only; bug #6809)
- 3D support: fixed *GL_EXT_texture_sRGB* support
- PXE: fixed ZENworks PXE boot regression
- OVF: fixed slower export and larger images under certain circumstances (3.2.6 regression; bug #7073)
- USB: properly signal an interrupt if the port suspend status changes
- USB: respect the remote-only filter
- USB: avoid VM hang when changing the configuration of certain devices (Windows hosts only)
- USB: fix a crash with older Linux kernels and non-ASCII characters in device strings (Linux hosts only; bug #6983)
- PageFusion: fixed conflict with the guest execution feature
- PageFusion: fixed stability issues with a large number of VMs
- PageFusion: fixed host crashes with guest SMP and Win64 guests
- Memory ballooning: fixed problems restoring VMs with pre-allocation enabled
- Bridged networking: fixed performance issue with GRO enabled on bridged device (bug #7059)

- Hostonly networking: fixed performance issue (3.2.6 regression; bug #7081)
- Hard disks: fix auto-reset of immutable disk at VM startup (bug #6832)
- BusLogic: several fixes for Windows NT/2000 and SCO OpenServer guests
- LsiLogic: fixed I/O errors under rare circumstances
- Sharing disks: support for attaching one disk to several VMs without external tools and tricks
- Shared Folders: several fixes and performance enhancements for Solaris guests (bugs #4154 and #6512)
- Solaris Installer: added support for remote installations
- Guest Properties API: correctly support enumerating the properties of a running VM with an empty “patterns” field (bug #7171)
- Guest properties: properly delete transient properties on shutdown
- VRDP video redirection performance improvements and stability fixes
- Settings: silently fix host audio driver when reading machine XML settings files or OVF written by VirtualBox on a different host OS, for example convert DirectSound to PulseAudio (bug #7209)
- Settings: properly store the NAT network setting in XML settings file version 1.10 and later (bug #6176)
- VBoxManage: handle differencing images with parent UUID correctly in subcommand *openmedium disk* (bug #6751)
- Web service: enabled HTTP keepalive for much better performance
- Web service: added timestamps to logging output
- Web service: treat 8-bit strings as UTF-8 not ASCII
- X11 Additions: fix for Xorg 6.8 guests (e.g. RHEL4)

15.24 Version 3.2.6 (2010-06-25)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed host crash when running 64-bit guests on 32-bit hosts with certain Intel CPUs (VT-x only; bug #6166)
- VMM: allow 64-bit SMP guests on 32-bit hosts (VT-x and AMD-V only; does not apply to Mac OS X, which already supports it)
- VMM: fixed Guru mediation if guests with more than 2GB are booted with VT-x/AMD-V disabled (bug #5740)
- VMM: fixed TR limit trashing (VT-x and 64-bit host only; bug #7052)
- Page Fusion: several bug fixes for SMP guests (including bug #6964)
- Teleportation: several fixes and improvements
- Mac OS X server guests: compatibility fix

- EFI: fixed memory detection for guests with 2GB or more RAM assigned
- GUI: added a workaround for a Linux kernel bug which affecting asynchronous I/O on ext4 / xfs file systems (Linux hosts only)
- GUI: added setting for multiple VRDP connections; useful if multiple screens are enabled
- GUI: another fix for the keyboard capturing bug under metacity (bug #6727)
- GUI: fixed quit dialog when used in seamless or full screen mode (Mac OS X hosts only; bug #6938)
- GUI: handle the extra key on the Brazilian keyboard on X11 hosts again (bug #7022).
- 2D Video acceleration: fixed crashes when leaving the full screen mode (bug #6768)
- VBoxManage: fixed *storageattach* error handling (bug #6927)
- VBoxManage: fixed *dhcpserver add* (3.2.0 regression; bug #7031)
- Storage: fixed hang with images located on filesystems which don't support asynchronous I/O (bug #6905)
- Storage: fixed raw disks on Windows hosts (3.2.0 regression; bug #6987)
- LsiLogic: fixed hang with older Linux guests
- BusLogic: fixed hang during I/O
- SATA: set initial number of ports to 1 as some guests can't handle 30 ports (e.g. CentOS 4 and FreeBSD; bug #6984)
- SATA: performance improvement
- SCSI: fixed error when using the full format option during Windows installation (bug #5101)
- iSCSI: fixed authentication (bug #4031)
- Host-only/bridged networking: fixed excessive host kernel warnings under certain circumstances (Linux hosts only; 3.2.0 regression; bug #6872)
- NAT: fixed potential memory leaks
- NAT: increased the size of the memory pool for 16K Jumbo frames (performance tweak)
- NAT: allow to link/unlink the network cable even if the VM is currently paused
- E1000: disconnect cable was not properly handled if the NIC was not yet initialized by the guest
- OVF: export performance optimization
- OVF: upgraded OS type definitions to CIM 2.25.0 so that Windows 7 and other OSes are now tagged correctly on export
- Settings: the setting for disabling the host I/O cache was sometimes not properly saved
- Settings: save machine state into XML correctly even when snapshot folder has been changed to a non-default location (bug #5656)
- USB: allow the guest to disable an EHCI port

- USB: find a valid language ID before querying strings (bug #7034)
- POSIX hosts: fixed several memory leaks (3.2.0 regression)
- Solaris hosts: fixed VDI access problem under certain circumstances (IDE/SATA; 3.2.0 regression)
- Solaris hosts: fixed VM fails to start on 32-bit hosts (3.2.0 regression; bug #6899)
- Windows hosts (32-bit): increase guest RAM limit if the host kernel allows for more virtual address space
- Linux Additions: re-read a directory after a file was removed (bug #5251)
- Linux Additions: install the DRI driver in the right location on ArchLinux guests (bug #6937)
- X11 Additions: fixed spurious mouse movement events (bug #4260)
- Solaris Additions: fixed guest execution feature
- Windows Additions: automatic logon on Windows Vista/Windows 7 is now able to handle renamed and principal user accounts; added various bugfixes
- Windows Additions: improved command line parsing of the installer
- Windows Additions: fixed driver verifier bugcheck in VBoxMouse (bug #6453)
- 3D support: fixed OpenGL support for 32bit applications under 64bit Windows guests

15.25 Version 3.2.4 (2010-06-07)

This is a maintenance release. The following items were fixed and/or added:

- GUI: fixed superfluous resize-event on powering-on VM for X11 (improvement for the 3.2.2 fix)
- GUI: fixed keyboard capturing bug under metacity (bug #6727)
- Host-only/bridged networking: fixed guest-to-guest communication over wireless (3.2.0 regression; bug #6855)
- Storage: fixed a potential guest disk corruption with growing images (3.2.0 regression)
- Page Fusion: fixed shared module detection for Win64 guests
- 3D support: allow use of *CR_SYSTEM_GL_PATH* again (bug #6864)
- 3D support: fixed a host assertion for some multi-threaded guest applications (bug #5236)
- 3D support: fixed host crashes with nVIDIA drivers on WDDM startup
- OVF: fixed import of OVF's with a VM description (annotation) (3.2.2 regression; bug #6914)
- VRDP: fixed issues with secondary monitors (bug #6759)

15.26 Version 3.2.2 (2010-06-02)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed rare invalid guest state guru meditation (VT-x only)
- VMM: fixed poor performance with nested paging and unrestricted guest execution (VT-x only; bug #6716)
- VMM: fixed occasional guru meditation during Windows 7 bootup (bug #6728)
- GUI: keep the status for remote control in sync with the actual state
- GUI: don't exit after a successful refresh of an invalid VM configuration
- GUI: fixed keyboard capturing bug under metacity (bug #6727)
- GUI: fixed crash during VM termination if a modal dialog is open
- GUI: default controllers names of New VM Wizard are synchronized with VM settings
- GUI: fixed superfluous resize-event on powering-on VM for X11
- GUI: fixed regression - missed USB item's tool-tip of USB devices menu
- GUI: Activate VM window on mouse-hovering for multi-monitor VMs
- VBoxSDL/Linux hosts: automated keyboard type detection (bug #5764)
- SATA: fixed crash during VM suspend under rare circumstances
- SATA: fixed crash during VM reset after a snapshot was taken
- Storage: fixed sporadic hang of SMP guests using SATA or LSI Logic SCSI and asynchronous I/O
- Virtio-net: fix for guests with more than about 4GB RAM (bug #6784)
- Page Fusion: fixed VBoxService crash with enabled Page Fusion on Win64 guests
- Page Fusion: added kernel module sharing
- HGCM: fixed memory leak which showed up if the Guest Additions were accessing a non-existing HGCM service
- Teleportation: several fixes
- Floppy: don't disable the host I/O cache by default
- USB: fixed 3.1 regression with certain devices (e.g. iPhone); Windows host only
- Serial: updated the guest device emulation to 16550A and reduced the probability for losing bytes during transmission (bug #1548)
- NAT: re-fetch the name server parameters from the host on guest DHCP requests to handle host network switches more gracefully (bug #3847)
- NAT: fixed parsing of IPv4 addresses in CIDR notation (bug #6797)
- NAT: limit the number of name servers passed to the guest to four (non-Windows hosts only; bug #4098)
- NAT: fixed DNS transaction ID mismatch (bug #6833)

- VDE: fixed changing the attachment during runtime
- Bridged networking: fixed memory leak in the Bridged Networking driver for Windows hosts (bug #6824)
- Windows Additions: fix for NT4 guests (bug #6748)
- Windows Additions: re-introduced system preparation feature
- Linux guests: enable PAE for RedHat guests by default
- Linux guests: fix support for disabling mouse integration (bug #6714)
- Web service: fixed a rare crash when calling IGuest methods from the web service
- OVF: fixed wrong hard disk UUIDs on export (bug #6802)
- OVF: fixed 3.2.0 regression importing legacy OVF 0.9 files
- 3D support: fixed OpenGL support for 64bit applications on windows guests
- 3D support: fixed various host crashes (bugs #2954, #5713, #6443)

15.27 Version 3.2.0 (2010-05-18)

This version is a major update. The following major new features were added:

- Following the acquisition of Sun Microsystems by Oracle Corporation, the product is now called “Oracle VM VirtualBox” and all references were changed without impacting compatibility
- Experimental support for Mac OS X Server guests (see chapitre 3.1.1, *Inviqés Mac OS X*, page 49)
- Memory ballooning to dynamically in- or decrease the amount of RAM used by a VM (64-bit hosts only) (see chapitre 4.8.1, *Faire du ballon avec la mémoire*, page 83)
- Page Fusion automatically de-duplicates RAM when running similar VMs thereby increasing capacity. Currently supported for Windows guests on 64-bit hosts (see chapitre 4.8.2, *Fusion de page*, page 83)
- CPU hot-plugging for Linux (hot-add and hot-remove) and certain Windows guests (hot-add only) (see chapitre 9.5, *Montage de processeur à chaud*, page 170)
- New Hypervisor features: with both VT-x/AMD-V on 64-bit hosts, using large pages can improve performance (see chapitre 10.6, *Vagation nestée et VPIDs*, page 205); also, on VT-x, unrestricted guest execution is now supported (if nested paging is enabled with VT-x, real mode and protected mode without paging code runs faster, which mainly speeds up guest OS booting)
- Support for deleting snapshots while the VM is running
- Support for multi-monitor guest setups in the GUI for Windows guests (see chapitre 3.5, *Paramètres d'affichage*, page 55)
- USB tablet/keyboard emulation for improved user experience if no Guest Additions are available (see chapitre 3.4.1, *Onglet “Carte mère”*, page 52)
- LsiLogic SAS controller emulation (see chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85)

- VRDP video acceleration (see chapitre 7.1.9, *Redirection graphique VRDP*, page 116)
- NAT engine configuration via API and VBoxManage
- Use of host I/O cache is now configurable (see chapitre 5.7, *Mise en cache des E/S dans l'hôte*, page 95)
- Guest Additions: added support for executing guest applications from the host system (replaces the automatic system preparation feature; see chapitre 4.7, *Contrôle de l'invité*, page 82)
- OVF: enhanced OVF support with custom namespace to preserve settings that are not part of the base OVF standard

In addition, the following items were fixed and/or added:

- VMM: fixed Windows 2000 guest crash when configured with a large amount of RAM (bug #5800)
- Linux/Solaris guests: PAM module for automatic logons added
- GUI: guess the OS type from the OS name when creating a new VM
- GUI: added VM setting for passing the time in UTC instead of passing the local host time to the guest (bug #1310)
- GUI: fixed seamless mode on secondary monitors (bugs #1322 and #1669)
- GUI: offer to download the user manual in the OSE version (bug #6442)
- GUI: allow to set an empty host key to disallow any host key combination (bug #684)
- GUI: allow to restrict the possible actions when shutting down the VM from the GUI
- Main: allow to start a VM even if a virtual DVD or floppy medium is not accessible
- Settings: be more robust when saving the XML settings files
- Mac OS X: rewrite of the CoreAudio driver and added support for audio input (bug #5869)
- Mac OS X: external VRDP authentication module support (bug #3106)
- Mac OS X: moved the realtime dock preview settings to the VM settings (no global option anymore). Use the dock menu to configure it
- Mac OS X: added the VM menu to the dock menu
- 3D support: fixed corrupted surface rendering (bug #5695)
- 3D support: fixed VM crashes when using *ARB_IMAGING* (bug #6014)
- 3D support: fixed assertion when guest applications uses several windows with single OpenGL context (bug #4598)
- 3D support: added *GL_ARB_pixel_buffer_object* support
- 3D support: added OpenGL 2.1 support
- 3D support: fixed Final frame of Compiz animation not updated to the screen (Mac OS X only) (bug #4653)
- 3D support: fixed blank screen after loading snapshot of VM with enabled Compiz

- Added support for *Virtual Distributed Ethernet* (VDE) (Linux hosts only; see chapitre [6.2, Introduction aux modes réseaux](#), page [99](#))
- Added support for virtual high precision event timer (HPET)
- OVF: fixed mapping between two IDE channels in OVF and the one IDE controller in VirtualBox
- OVF: fix VMDK format string identifiers and sort XML elements from rasd: namespace alphabetically as prescribed by standard
- VBoxShell: interactive Python shell extended to be fully functional TUI for VirtualBox
- Linux Additions: support Fedora 13 (bug #6370)
- VBoxManage: fixed overly strict checks when creating a raw partition VMDK (bugs #688, #4438)

15.28 Version 3.1.8 (2010-05-10)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed crash with the OpenSUSE 11.3 milestone kernel during early boot (software virtualization only)
- VMM: fixed invalid state during teleportation
- VMM: fixed OS/2 guest crash with nested paging enabled
- VMM: fixed massive display performance loss (AMD-V with nested paging only)
- GUI: fixed off-by-one bug when passing absolute mouse coordinates to the guest (3.1.6 regression)
- GUI: show the real version of the Guest Additions, not the interface version
- GUI: when adding a DVD or floppy slot in the VM mass storage settings dialog, don't attach a random medium but just leave the slot empty
- GUI: added `--seamless` and `--fullscreen` command line switches (bug #4220)
- GUI: fixed a `SEGVFAULT` under rare circumstances
- 2D Video acceleration: fixed display issues when working with non 32-bit modes (bugs #6094 & #6208)
- LsiLogic: fixed detection of hard disks attached to port 0 when using the drivers from LSI
- ATA: fixed sporadic crash with Linux guests when having a hard disk and DVD drive on the same channel (bug #6079)
- Network: allow to start a VM even if not all network adapters are attached
- Network: promiscuous mode support for e1000 and paravirtualized adapters (bug #6519)
- NAT: fixed ICMP latency (non-Windows hosts only; bug #6427)
- SCSI: fixed guest crashes under certain circumstances when booting from SCSI devices
- VBoxManage: fixed several bugs in cloning of images (one of them is bug #6408)

- VBoxManage: fixed `modifyvm --natnet default`
- Solaris hosts: fixed a kernel panic when bridged networking might fail to initialize
- Solaris hosts: fixed priority tagged VLAN packets in bridged networking
- Shared Folders: fixed issue with copying read-only files (Linux guests only; bug #4890)
- Shared Folders: renamed the guest kernel module from `vboxvfs` to `vboxsf` to make it load on demand by the Linux kernel. Fixes mounting from `/etc/fstab` in Ubuntu 10.04
- Shared Folders: fixed setuid file permissions (Solaris guests only)
- Shared Folders: fixed deleting directories recursively (Solaris guests only; bug #6513)
- Guest Additions: support seamless and dynamic resizing on certain older X11 guests (bug #5840)
- Solaris Additions: fixed OpenGL library dependencies (bug #6435)
- Keyboard/Mouse emulation: fixed handling of simultaneous mouse/keyboard events under certain circumstances (bug #5375)
- Mouse emulation: never switch straight back from Explorer to IntelliMouse mode as it confuses the FreeBSD mouse driver (bug #6488)
- SDK: fixed memory leak in `IDisplay::takeScreenShotSlow()` (bug #6549)
- 3D support: fixed Final frame of Compiz animation not updated to the screen (Mac OS X only) (bug #4653)
- VRDP: allow to bind to localhost only on Mac OS X (bug #5227)
- Linux hosts: add host USB support for Ubuntu 10.04 and other hosts without the hal daemon or usbfs (bug #6343)
- Web service: more structs and array fixes in PHP bindings
- Windows hosts: make the bridged networking driver notify dll be correctly unregistered on uninstall (bug #5780)

15.29 Version 3.1.6 (2010-03-25)

This is a maintenance release. The following items were fixed and/or added:

- Linux hosts: fixed timing issue on hosts with Linux kernels 2.6.31 or later with certain CPUs (asynchronous timer mode; bug #6250)
- Linux hosts: properly handle host suspend/resume events on Linux kernels 2.6.30 or later (bug #5562)
- Mac OS X hosts: fixed VBoxSVC crash while enumerating the host network interfaces under certain circumstances
- Snapshots: fixed image corruption after snapshot merge under certain circumstances (bug #6023)
- Snapshots: fixed crash with VBoxHeadless / OSE
- VMM: fixed reference counting guru meditation (bug #4940)

- VMM: improved guest SMP stability
- VMM: fixed VT-x hardware debug issues (bugs #477 & #5792)
- VMM: fixed *PGMDynMapHCPPage* guru meditation (Mac OS X; VT-x only; bug #6095)
- VMM: fixed *pgmPoolTrackFlushGCPphysPTInt* guru meditations (Mac OS X; VT-x only; bugs #6095 & #6125)
- VMM: fixed host crash when running PAE guests in VT-X mode (Mac OS X only; bug #5771)
- GUI: fix displaying of error message (bug #4345)
- GUI: fix inability to enter seamless mode (bugs #6185, #6188)
- 3D support: fixed assertion and flickering when guest application uses several windows with a single OpenGL context (bug #4598)
- 3D support: fixed host crashes when using *GL_EXT_compiled_vertex_array* and array element calls (bug #6165)
- 3D support: fixed runtime linker errors with OpenGL guest libs (bug #5297)
- 3D support: fixed OpenGL extension viewer crash on startup (bug #4962)
- NAT: fixed a 3.1.4 regression on Windows hosts where graceful connection termination was broken (bug #6237)
- NAT: alternative network setting was not stored persistent (bug #6176)
- NAT: fixed memory corruption during ICMP traffic under certain circumstances
- Network: allow to switch the host interface or the internal network while a VM is running (bug #5781)
- VHD: fix for images with a block size different than 2MB
- USB: fixed filtered device attach regression (bug #6251)
- USB: fixed crash in OHCI under rare circumstances (bug #3571)
- VRDP: fixed hang under rare circumstances when attaching USB devices
- ACPI: prevent guest freezes when accessing /proc/acpi for determining the state of the host battery and the AC adapter (Linux hosts only; bug #2836)
- PulseAudio: fixed guest freezes under certain conditions (3.1.4 regression; bug #6224)
- BIOS: increased space for DMI strings
- BIOS: fixed interrupt routing problem for certain configurations (I/O-APIC enabled, ACPI not used; bug #6098)
- iSCSI: be more robust when handling the *INQUIRY* response
- iSCSI: be more robust when handling sense data
- BusLogic: fixed FreeBSD guests
- Web service: vboxwebsrv is now multithreaded
- Web service: fixed handling of structs and arrays in PHP bindings

- Solaris Installer: fixed netmask to stay persistent across reboots for Host-only interface (bug #4590)
- Linux installer: removed external dependency to libpng12.so (bug #6243)
- Solaris Additions: fixed superfluous kernel logging (bug #6181)
- Linux Additions: fixed hang when starting the X server in Fedora12 guests and in guests with Linux 2.6.33 or later (bug #6198)
- Linux Additions: support Mandriva speedboot runlevel (bug #5484)
- Linux Additions: fixed SELinux security context of mount.vboxsf (bug #6362)
- Linux Additions: support Ubuntu 10.04 (bug #5737)
- Web service: update PHP bindings to fix problems with enums and collections

15.30 Version 3.1.4 (2010-02-12)

This is a maintenance release. The following items were fixed and/or added:

- VMM: SMP stability fixes
- VMM: fixed guru meditation in certain rare cases (bug #5968)
- VMM: activate NXE for PAE enabled guests (VT-x and AMD-V on 32 bits hosts only; bug #3578)
- VMM: added workaround for broken BIOSes that make VirtualBox think AMD-V is in use (for details see bug #5639)
- VMM: fixed rare host reboot when restoring a saved state (bug #3945)
- VMM: fixed incompatibility with 2.6.32 Linux kernels (software virtualization only; bug #6100)
- VMM: turn on nested paging by default for new VMs (if available; VT-x and AMD-V only)
- VMM: turn on VPID by default for new VMs (if available; VT-x only)
- VMM: perform strict CPUID compatibility checks when teleporting; to get the old behavior set “VBoxInternal/CPUM/StrictCpuIdChecks” to 0
- VMM: fixed VM crash with certain 16 bits Windows applications (software virtualization only; bug #5399)
- Snapshots: fixed a 3.1 regression that broke deletion of snapshots when a machine had immutable or writethrough storage attached (bug #5727)
- Saved state: fixed *VERR_SSM_LOADED_TOO MUCH* error when loading *DisplayScreenshot*(bug #6162)
- VBoxManage: add *restorecurrent* operation to snapshots command
- VBoxManage: fixed broken snapshot lookup by name (bug #6070)
- GUI: fixed the broken “Reload” button that reloads the machine XML when a machine is inaccessible
- GUI: fixed guest full screen mode after reboot (bug #5372)

- GUI: handle Ctrl+Break properly on X11 hosts (bug #6122)
- GUI: fixed status LEDs for storage devices
- GUI: workaround for disabling the seamless mode on KDE hosts (KWin bug)
- 3D support: fixed SELinux warning saying VBoxOGL.so requires text relocation (bug #5690)
- 3D support: fixed Corrupted surface rendering (bug #5695)
- 3D support: free textures on guest application termination (bug #5206)
- 3D support: fixed *ubigraph_server* crashes (bug #4674)
- 3D support: fixes for 64-bit Solaris guests
- Seamless: disable seamless mode when guest changes screen resolution (bug #5655)
- NAT: fixed high CPU load under certain circumstances (Windows hosts only; bug #5787)
- NAT: fixed handling of the *broadcast* flag in DHCP requests
- NAT: fixed rare crash due to an assertion in the ICMP code (bug #3217)
- Virtio-net: don't crash when ports accessed beyond the valid range (bug #5923)
- LsiLogic: fix for Windows 7 guests
- ATA: fix for guru meditation when installing Solaris 8 guests (bug #5972)
- VHD: fixed an incompatibility with Virtual PC (bug #5990)
- VHD: update the footer backup after setting a new UUID (bug #5004)
- Host DVD: really fixed loading “passthrough” setting from configuration file (bug #5681)
- Shared Folders: fixed resolving of symlink target on Linux (3.1.2 regression)
- VRDP: fixed *VERR_NET_ADDRESS_IN_USE* error when restarting a VM (3.1 regression; bug #5902)
- VRDP: fixed crash on Mac OS X when 3D is enabled (3.1 regression)
- PulseAudio: fixed recording (bug #4302)
- USB: fixed a shutdown blue screen (Windows hosts only; bug #5885)
- BIOS: fixed attribute during text scroll (bug #3407)
- OVF: fix strange error messages on disk import errors
- OVF: do not require write access to the .ovf file during import (3.1 regression; bug #5762)
- iSCSI: fix taking snapshots of a running VM (bug #5849)
- Solaris hosts: several USB fixes (including support for Apple iPod; bug #5873)
- Solaris installer: fixed USB module removal and Solaris 10 “id” binary incompatibility
- Guest Additions: fixed wrong guest time adjustment if the guest clock is ahead (3.1 regression; non-Windows guests only)
- Linux Additions: fixed shared folders for Linux 2.6.32 guests (bug #5891)

- Linux Additions: make the mouse driver work on Debian 5.0.3 guests again (3.1.2 regression, bug #5832)
- Windows Additions: fixed malfunctioning VBoxService that broke time-sync (bug #5872)
- Windows Additions: fixed uninstallation issues on 64-bit guests
- Windows Additions: fixed some sysprep execution issues
- X.Org Additions: never reject the saved video mode as invalid (bug #5731)
- XFree86 Additions: accept video mode hints for the initial mode again

15.31 Version 3.1.2 (2009-12-17)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed SMP stability regression
- USB: fixed USB related host crashes on 64 bits Windows hosts (bug #5237)
- Main: wrong default HWVirtExExclusive value for new VMs (bug #5664)
- Main: DVD passthrough setting was lost (bug #5681)
- VBoxManage: iSCSI disks do not support adding a comment (bug #4460)
- VBoxManage: added missing `-cpus` and `-memory` options to OVF `-import`
- GUI: fixed VBox URL in update dialog for German and Dutch languages
- GUI: NLS updates
- OVF: fixed export of non standard storage controller names (bug #5643)
- Solaris hosts: several USB fixes (including support for Apple iPhone)
- Mac OS X hosts: several fixes for the 3D support
- Mac OS X hosts: re-enabled CMD+Key combinations, even if the Host-Key isn't CMD (bug #5684)
- Mac OS X hosts: fixed to fast scrolling if the mouse wheel is used inside the guest (bug #5672)
- Mac OS X hosts: dock & menubar don't disappear in full screen when the VM is not running on the primary display (bug #1762)
- Mac OS X hosts: added an option for enabling "Auto show Dock & Menubar in full screen" (bug #5636)
- Windows host installer: fixed starting VBox with wrong privileges right after installation (bug #4162)
- Host interface and host-only networking: prevent driver from unloading while a VM is still active (Windows host only)
- Host-only networking: fixed host-only interface creation (Windows host only) (bug #5708)
- Virtio-net: don't crash without an attached network
- Virtio-net: fixed the issue with intermittent network in VM with several virtual CPU cores

- NAT: fixed port-forwarding regressions (bug #5666)
- NAT: fixed crash under certain conditions (bug #5427)
- NAT: fixed resolving of names containing a slash or underscore when using the host resolver DNS proxy (bug #5698)
- ATA: fixed sporadic crash when resuming after a VM was forcefully paused (e.g. due to iSCSI target being unavailable)
- SATA: fixed raw vmdk disks (bug #5724)
- Linux guests: increased the default memory for Redhat and Fedora guests
- Linux Guest Additions: fixed installation on RHEL 3.9 guests and on some 64bit guests
- Linux Guest Additions: prevent SELinux warnings concerning text relocations in VBox-OGL.so (bug #5690)
- X11 guests: fixed mouse support for some Xorg 1.4 guests (openSUSE 11.0)
- X11 guests: fixed xorg.conf modification for some older Xorg releases (openSUSE 11.1)
- Windows guests: fixed some VBoxService shutdown issues
- Windows guests: fixed VBoxVideo spinlock issues on NT4
- Windows Guest Additions: fixed uninstallation issues of NT4
- Shared Folders: fixed resolving of symlink target (bug #5631)
- 2D Video acceleration: delay loading of OpenGL dlls for Windows hosts to avoid GUI crashes on misconfigured systems
- 2D Video acceleration: fixed issues with video picture not displayed on playback

15.32 Version 3.1.0 (2009-11-30)

This version is a major update. The following major new features were added:

- Teleportation (aka live migration); migrate a live VM session from one host to another (see chapitre 7.2, *Téléportation*, page 117)
- VM states can now be restored from arbitrary snapshots instead of only the last one, and new snapshots can be taken from other snapshots as well (“branched snapshots”; see chapitre 1.10, *Instantanés*, page 28)
- 2D video acceleration for Windows guests; use the host video hardware for overlay stretching and color conversion (see chapitre 4.4.2, *L'accélération matérielle 2D pour les invités Windows*, page 79)
- More flexible storage attachments: CD/DVD drives can be attached to arbitrary storage controllers, and there can be more than one such drive (chapitre 5, *Le stockage virtuel*, page 85)
- The network attachment type can be changed while a VM is running
- Complete rewrite of experimental USB support for OpenSolaris hosts making use of the latest USB enhancements in Solaris Nevada 124 and higher

- Significant performance improvements for PAE and AMD64 guests (VT-x and AMD-V only; normal (non-nested) paging)
- Experimental support for EFI (Extensible Firmware Interface; see chapitre 3.12, *Autre firmware (EFI)*, page 62)
- Support for paravirtualized network adapters (virtio-net; see chapitre 6.1, *Matériel réseau virtuel*, page 98)

In addition, the following items were fixed and/or added:

- VMM: guest SMP fixes for certain rare cases
- GUI: snapshots include a screenshot
- GUI: locked storage media can be unmounted by force
- GUI: the log window grabbed all key events from other GUI windows (bug #5291)
- GUI: allow to disable USB filters (bug #5426)
- GUI: improved memory slider in the VM settings
- 3D support: major performance improvement in VBO processing
- 3D support: added *GL_EXT_framebuffer_object*, *GL_EXT_compiled_vertex_array* support
- 3D support: fixed crashes in FarCry, SecondLife, Call of Duty, Unreal Tournament, Eve Online (bugs #2801, #2791)
- 3D support: fixed graphics corruption in World of Warcraft (bug #2816)
- 3D support: fixed Final frame of Compiz animation not updated to the screen (bug #4653)
- 3D support: fixed incorrect rendering of non ARGB textures under compiz
- iSCSI: support iSCSI targets with more than 2TiB capacity
- VRDP: fixed occasional VRDP server crash (bug #5424)
- Network: fixed the E1000 emulation for QNX (and probably other) guests (bug #3206)
- NAT: added host resolver DNS proxy (see chapitre 9.12.6, *Utiliser le résolveur de l'hôte comme proxy DNS en mode NAT*, page 180)
- VMDK: fixed incorrectly rejected big images split into 2G pieces (bug #5523, #2787)
- VMDK: fixed compatibility issue with fixed or raw disk VMDK files (bug #2723)
- VHD: fixed incompatibility with Hyper-V
- Support for Parallels version 2 disk image (HDD) files; see chapitre 5.2, *Fichiers images de disque (VDI, VMDK, VHD, HDD)*, page 88
- OVF: create manifest files on export and verify the content of an optional manifest file on import
- OVF: fixed memory setting during import (bug #4188)
- Mouse device: now five buttons are passed to the guest (bug #3773)
- VBoxHeadless: fixed loss of saved state when VM fails to start

- VBoxSDL: fixed crash during shutdown (Windows hosts only)
- X11 based hosts: allow the user to specify their own scan code layout (bug #2302)
- Mac OS X hosts: don't auto show the menu and dock in full screen (bug #4866)
- Mac OS X hosts (64 bit): don't interpret mouse wheel events as left click (bug #5049)
- Mac OS X hosts: fixed a VM abort during shutdown under certain conditions
- Solaris hosts: combined the kernel interface package into the VirtualBox main package
- Solaris hosts: support for OpenSolaris Boomer architecture (with OSS audio backend)
- Shared Folders: VBOXSVR is visible in Network folder (Windows guests, bug #4842)
- Shared Folders: performance improvements (Windows guests, bug #1728)
- Windows, Linux and Solaris Additions: added balloon tip notifier if VirtualBox host version was updated and Additions are out of date
- Solaris guests: fixed keyboard emulation (bug #1589)
- Solaris Additions: fixed *as_pagelock()* failed errors affecting guest properties (bug #5337)
- Windows Additions: added automatic logon support for Windows Vista and Windows 7
- Windows Additions: improved file version lookup for guest OS information
- Windows Additions: fixed runtime OS detection on Windows 7 for session information
- Windows Additions: fixed crash in seamless mode (contributed by Huihong Luo)
- Linux Additions: added support for uninstalling the Linux Guest Additions (bug #4039)
- Linux guest shared folders: allow mounting a shared folder if a file of the same name as the folder exists in the current directory (bug #928)
- SDK: added object-oriented web service bindings for PHP5

15.33 Version 3.0.12 (2009-11-10)

This is a maintenance release. The following items were fixed and/or added:

- VMM: reduced IO-APIC overhead for 32 bits Windows NT/2000/XP/2003 guests; requires 64 bits support (VT-x only; bug #4392)
- VMM: fixed double timer interrupt delivery on old Linux kernels using IO-APIC (caused guest time to run at double speed; bug #3135)
- VMM: re-initialize VT-x and AMD-V after host suspend or hibernate; some BIOSes forgot this (Windows hosts only; bug #5421)
- VMM: fixed loading of saved state when RAM preallocation is enabled
- BIOS: ignore unknown shutdown codes instead of causing a guru meditation (bug #5389)
- GUI: never start a VM on a single click into the selector window (bug #2676)
- Serial: reduce the probability of lost bytes if the host end is connected to a raw file
- VMDK: fixed handling of split image variants and fix a 3.0.10 regression (bug #5355)

- VRDP: fixed occasional VRDP server crash
- Network: even if the virtual network cable was disconnected, some guests were able to send / receive packets (E1000; bug #5366)
- Network: even if the virtual network cable was disconnected, the PCNet card received some spurious packets which might confuse the guest (bug #4496)
- Shared Folders: fixed changing case of file names (bug #2520)
- Windows Additions: fixed crash in seamless mode (contributed by Huihong Luo)
- Linux Additions: fixed writing to files opened in *O_APPEND* mode (bug #3805)
- Solaris Additions: fixed regression in Guest Additions driver which among other things caused lost guest property updates and periodic error messages being written to the system log

15.34 Version 3.0.10 (2009-10-29)

This is a maintenance release. The following items were fixed and/or added:

- VMM: guest SMP stability fixes
- VMM: fixed guru meditation with nested paging and SMP guests (bug #5222)
- VMM: changed VT-x/AMD-V usage to detect other active hypervisors; necessary for e.g. Windows 7 XP compatibility mode (Windows & Mac OS X hosts only; bug #4239)
- VMM: guru meditation during SCO OpenServer installation and reboot (VT-x only; bug #5164)
- VMM: fixed accessed bit handling in certain cases (bug #5248)
- VMM: fixed VPID flushing (VT-x only)
- VMM: fixed broken nested paging for 64 bits guests on 32 bits hosts (AMD-V only; bug #5285)
- VMM: fixed loading of old saved states/snapshots (bug #3984)
- Mac OS X hosts: fixed memory leaks (bug #5084)
- Mac OS X hosts (Snow Leopard): fixed redraw problem in a dual screen setup (bug #4942)
- Windows hosts: installer updates for Windows 7
- Solaris hosts: out of memory handled incorrectly (bug #5241)
- Solaris hosts: the previous fix for #5077 broke the DVD host support on Solaris 10 (VBox 3.0.8 regression)
- Linux hosts: fixed module compilation against Linux 2.6.32rc4 and later
- Guest Additions: fixed possible guest OS kernel memory exhaustion
- Guest Additions: fixed stability issues with SMP guests
- Windows Additions: fixed color depth issue with low resolution hosts, netbooks, etc. (bug #4935)

- Windows Additions: fixed NO_MORE_FILES error when saving to shared folders (bug #4106)
- Windows Additions: fixed subdirectory creation on shared folders (bug #4299)
- Linux Additions: *sendfile()* returned -EOVERFLOW when executed on a shared folder (bug #2921)
- Linux Additions: fixed incorrect disk usage value (non-Windows hosts only)
- Linux installer: register the module sources at DKMS even if the package provides proper modules for the current running kernel
- 3D support: removed invalid OpenGL assertion (bug #5158)
- Network: fixed the Am79C973 PCNet emulation for QNX (and probably other) guests (bug #3206)
- VMDK: fix handling of split image variants
- VHD: do not delay updating the footer when expanding the image to prevent image inconsistency
- USB: stability fix for some USB 2.0 devices
- GUI: added a search index to the .chm help file
- GUI/Windows hosts: fixed CapsLock handling on French keyboards (bug #2025)
- Shared clipboard/X11 hosts: fixed a crash when clipboard initialisation failed (bug #4987)

15.35 Version 3.0.8 (2009-10-02)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed 64 bits guest on 32 bits host regression in 3.0.6 (VT-x only; bug #4947)
- VMM: fixed a recompiler triple fault guru meditation (VT-x & AMD-V only; bug #5058)
- VMM: fixed hang after guest state restore (AMD-V, 32 bits Windows guest and IO-APIC enabled only; bug #5059)
- VMM: fixed paging issue with OS/2 guests
- VMM: fixed guru meditation in rare cases (2.0 regression; software virtualization only)
- VMM: fixed release assertion during state restore when using the Sound Blaster 16 emulation (bug #5042)
- Security: fixed vulnerability that allowed to execute commands with root privileges
- Linux hosts: fixed runtime assertion in semaphore implementation which was triggered under certain conditions (bug #616)
- Linux hosts: change the default USB access mode on certain distributions (bugs #3394 and #4291)
- Linux hosts: on hardened Gentoo, the VBoxSVC daemon crashed by opening the VM network settings (bug #3732)

- Linux hosts, Solaris hosts: pass the XAUTHORITY variable along the DISPLAY variable when starting a VM from VBoxManage or from the VM selector (bug #5063)
- Linux hosts: use sysfs to enumerate host drives if hal is not available
- Solaris hosts: fixed a bug which would hang the host sporadically as interrupts were not re-enabled every time
- Solaris hosts: fixed a kernel panic with bridged and host-only networking (bug #4775)
- Solaris hosts: fixed incorrectly persistent CD/DVD-ROMs when changing them (bug #5077)
- X11-based hosts: support additional function keys on Sun keyboards (bug #4907)
- Mac OS X hosts (Snow Leopard): fixed problem starting headless VMs without a graphical session (bug #5002)
- Mac OS X hosts: fixed problem listing host-only adapter names with trailing garbage (attached VMs won't start)
- Windows Additions: now work with Vista 64-bit Home editions (bug #3865)
- Windows Additions: fixed screen corruption with ZoomText Magnifier
- Windows Additions: fixed NPGetUniversalName failure (bug #4853)
- Windows Additions: fixed Windows NT regression (bug #4946)
- Windows Additions: fixed VBoxService not running if no Shared Folders are installed
- Linux Additions: implemented *ftruncate* (bug #4771)
- VRDP: start VM even if configured VRDP port is in use
- Networking: the PCnet network device stopped receiving under rare conditions (bug #4870)
- VBoxManage: implemented `controlvm vrdpport` command
- iSCSI: fixed issue with NetApp targets (bug #5072)
- SCSI: add support for virtual disks larger than 2TB
- USB: fixed potential crash when unplugging USB2 devices (bug #5089)
- NAT: IPSEC did not properly work with Linux guests (bug #4801)

15.36 Version 3.0.6 (2009-09-09)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed IO-APIC overhead for 32 bits Windows NT, 2000, XP and 2003 guests (AMD-V only; bug #4392)
- VMM: fixed a Guru meditation under certain circumstances when enabling a disabled device (bug #4510)
- VMM: fixed a Guru meditation when booting certain Arch Linux guests (software virtualization only; bug #2149)

- VMM: fixed hangs with 64 bits Solaris & OpenSolaris guests (bug #2258)
- VMM: fixed decreasing *rdtsc* values (AMD-V & VT-x only; bug #2869)
- VMM: small Solaris/OpenSolaris performance improvements (VT-x only)
- VMM: *cpuid* change to correct reported virtual CPU ID in Linux
- VMM: NetBSD 5.0.1 CD hangs during boot (VT-x only; bug #3947)
- Solaris hosts: worked around an issue that caused the host to hang (bug #4486)
- Solaris hosts: fixed a rare host system deadlock when using bridged networking
- Solaris hosts: fixed a potential host system deadlock when CPUs were onlined or offlined
- Solaris hosts installer: added missing dependency for UTF-8 package (bug #4899)
- Linux hosts: don't crash on Linux PAE kernels < 2.6.11 (in particular RHEL/CentOS 4); disable VT-x on Linux kernels < 2.6.13 (bug #1842)
- Linux/Solaris hosts: correctly detect keyboards with fewer keys than usual (bug #4799)
- Mac OS X hosts: prevent password dialogs in 32 bits Snow Leopard
- Python WS: fixed issue with certain enumerations constants having wrong values in Python web services bindings
- Python API: several threading and platform issues fixed
- Python shell: added *exportVM* command
- Python shell: various improvements and bugfixes
- Python shell: corrected detection of home directory in remote case
- OVF: fixed XML comment handling that could lead to parser errors
- Main: fixed a rare parsing problem with port numbers of USB device filters in machine settings XML
- Main: restrict guest RAM size to 1.5 GB (32 bits Windows hosts only)
- Main: fixed possible hang during guest reboot (bug #3792)
- GUI: fixed rare crash when removing the last disk from the media manager (bug #4795)
- VBoxManage: fixed *guestproperty* for Mac OS X hosts (bug #3806)
- VBoxManage: fixed setting guest properties with *-flags* or *-flags*
- Web service: fixed a severe memory leak, at least on platforms using XPCOM
- Serial: fixed host mode (Solaris, Linux and Mac OS X hosts; bug #4672)
- VRDP: Remote USB Protocol version 3
- SATA: fixed hangs and BSODs introduced with 3.0.4 (bugs #4695, #4739, #4710)
- SATA: fixed a bug which prevented Windows 7 from detecting more than one hard disk
- SATA/SCSI: fixed rare random guest crashes and hangs
- SCSI: fixed problem with Fedora 11 refusing to boot after kernel update

- iSCSI: fix logging out when the target has dropped the connection, fix negotiation of parameters, fix command resend when the connection was dropped, fix processing SCSI status for targets which do not use phase collapse
- BIOS: fixed a bug that caused the OS/2 boot manager to fail (2.1.0 regression, bug #3911)
- PulseAudio: don't hang during VM termination if the connection to the server was unexpectedly terminated (bug #3100)
- Mouse: fixed weird mouse behaviour with SMP (Solaris) guests (bug #4538)
- HostOnly Network: fixed failure in *CreateHostOnlyNetworkInterface()* on Linux (no GUID)
- HostOnly Network: fixed wrong DHCP server startup while hostonly interface bringup on Linux
- HostOnly Network: fixed incorrect factory and default MAC address on Solaris
- HostOnly Network: fixed the problem with listing host-only interfaces on Mac OS X when all physical interfaces are down (bugs #4698, #4790)
- DHCP: fixed a bug in the DHCP server where it allocated one IP address less than the configured range
- E1000: fixed receiving of multicast packets
- E1000: fixed up/down link notification after resuming a VM
- NAT: fixed Ethernet address corruptions (bug #4839)
- NAT: fixed hangs, dropped packets and retransmission problems (bug #4343)
- Bridged networking: fixed packet queue issue which might cause DRIVER_POWER_STATE_FAILURE BSOD for Windows hosts (bug #4821)
- Windows Additions: fixed a bug in VBoxGINA which prevented selecting the right domain when logging in the first time
- Windows host installer: should now also work on unicode systems (like Korean, bug #3707)
- Windows host installer: check for sufficient disk space
- Shared clipboard: do not send zero-terminated text to X11 guests and hosts (bug #4712)
- Shared clipboard: use a less CPU intensive way of checking for new data on X11 guests and hosts (bug #4092)
- Guest Additions: do not hide the host mouse cursor when restoring a saved state (bug #4700)
- Windows guests: fixed issues with the display of the mouse cursor image (bugs #2603, #2660 and #4817)
- SUSE 11 guests: fixed Guest Additions installation (bug #4506)
- Guest Additions: support Fedora 12 Alpha guests (bugs #4731, #4733 and #4734)

15.37 Version 3.0.4 (2009-08-04)

This is a maintenance release. The following items were fixed and/or added:

- VMM: 64 bits guest stability fixes (AMD-V only; bugs #3923 & #3666)
- VMM: SMP stability fixes (AMD-V only)
- VMM: SMP performance improvement (esp. for Solaris guests)
- VMM: eliminated several bugs which could lead to a host reboot
- VMM: fixed OS/2 ACP2 boot floppy hang (VT-x only)
- VMM: small performance improvement for OpenSolaris guests (AMD-V only)
- VMM: fixed CentOS/Xen reboot (software virtualization only; bug #4509)
- SATA: fixed hangs / BSOD during Windows XP installation (bug #4342)
- SATA: mark the ports as non hotpluggable (bug #3920)
- 3D support: fix deadlocks and context/window tracking for multithreaded applications (bug #3922)
- 3D support: fix memory leaks when terminating OpenGL guest applications
- 3D support: fix crash in Call of Duty
- NAT: using two or more NAT adapters in one VM was broken (3.0.0 regression)
- NAT: fixed network communication corruptions (bugs #4499, #4540, #4591, #4604)
- NAT: fixed passive ftp access to host server (bug #4427)
- iSCSI: fixed cloning to/from iSCSI disks
- GUI: fixed path separator handling for the OVF export on Windows (bug #4354)
- GUI: the mini toolbar was only shown on the first host display (bug #4654)
- GUI: added a VM option to display the mini toolbar on top
- GUI: don't crash when adding plus configuring host-only network interfaces
- Shared Folders: fixed selection of a drive root directory as a shared folder host path in VirtualBox (Windows host only)
- USB: fixed a bug that may have rendered USB device filter settings inactive (3.0.2 regression, bug #4668)
- Guest Additions: report the Guest Additions version to the guest properties (bug #3415)
- Mac OS X hosts: fix creation of VMDK files giving raw partition access (bug #1461)
- Mac OS X hosts: improved support for Snow Leopard
- Linux hosts: fixed problems leading to wrong colors or transparency in host windows with some graphics drivers (bug #3095)
- Linux hosts: hardware detection fallbacks if the hal service fails to find any DVD drives
- Linux and Solaris hosts: Work around color handling problems in Qt (bug #4353)

- Solaris hosts: fixed memory leaks in host-only networking
- Solaris Installer: fixed incorrect netmask for Host-only interface (bug #4590)
- Solaris Installer: added package dependency for Python and Python-devel (bug #4570)
- X11 guests: prevent windows from being skipped in seamless mode KDE guests (bugs #1681 and #3574)
- X11 guests: fixed screen corruption in X11 guests when large amounts of video RAM were allocated (bug #4430)
- X11 guests: some fixes when switching between host and guest-drawn mouse pointers
- X11 guests: fixed an issue which caused seamless mode to stop working as it should (the main issue listed in bug #2238)

15.38 Version 3.0.2 (2009-07-10)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed network regressions (guest hangs during network IO) (bug #4343)
- VMM: guest SMP performance improvements
- VMM: fixed hangs and poor performance with Kaspersky Internet Security (VT-x/AMD-V only; bug #1778)
- VMM: fixed crashes when executing certain Linux guests (software virtualization only; bugs #2696 & #3868)
- ACPI: fixed Windows 2000 kernel hangs with IO-APIC enabled (bug #4348)
- APIC: fixed high idle load for certain Linux guests (3.0 regression)
- BIOS: properly handle Ctrl-Alt-Del in real mode
- iSCSI: fixed configuration parsing (bug #4236)
- OVF: fix potential confusion when exporting networks
- OVF: compatibility fix (bug #4452)
- OVF: accept ovf:/disk/ specifiers with a single slash in addition to ovf://disk/ (bug #4452)
- NAT: fixed crashes under certain circumstances (bug #4330)
- 3D support: fixed dynamic linking on Solaris/OpenSolaris guests (bug #4399)
- 3D support: fixed incorrect context/window tracking for multithreaded apps
- Shared Folders: fixed loading from saved state (bug #1595)
- Shared Folders: host file permissions set to 0400 with Windows guest (bug #4381)
- X11 host and guest clipboard: fixed a number of issues, including bug #4380 and #4344
- X11 Additions: fixed some issues with seamless windows in X11 guests (bug #3727)
- Windows Additions: added VBoxServiceNT for NT4 guests (for time synchronization and guest properties)

- Windows Additions: fixed version lookup
- Linux Installer: support Pardus Linux
- Linux hosts: workaround for buggy graphics drivers showing a black VM window on recent distributions (bug #4335)
- Linux hosts: fixed typo in kernel module startup script (bug #4388)
- Solaris hosts: several installer fixes
- Solaris hosts: fixed a preemption issue causing VMs to never start on Solaris 10 (bug #4328)
- Solaris guests: fixed mouse integration for OpenSolaris 2009.06 (bug #4365)
- Windows hosts: fixed high CPU usage after resuming the host (bug #2978)
- Fixed a settings file conversion bug which sometimes caused hardware acceleration to be enabled for virtual machines that had no explicit configuration in the XML

15.39 Version 3.0.0 (2009-06-30)

This version is a major update. The following major new features were added:

- Guest SMP with up to 32 virtual CPUs (VT-x and AMD-V only; see chapitre 3.4.2, *Onglet “Processeur”*, page 54)
- Windows guests: ability to use Direct3D 8/9 applications / games (experimental; see chapitre 4.4.1, *Accélération 3D matérielle (OpenGL and Direct3D 8/9)*, page 77)
- Support for OpenGL 2.0 for Windows, Linux and Solaris guests

In addition, the following items were fixed and/or added:

- Solaris hosts: allow suspend/resume on the host when a VM is running (bug #3826)
- Solaris hosts: loosen the restriction for contiguous physical memory under certain conditions
- Mac OS X hosts: fixed guest PAE
- Linux hosts: kernel module compile fixes for 2.6.31 (bug #4264)
- VMM: fixed occasional guru meditation when loading a saved state (VT-x only)
- VMM: eliminated IO-APIC overhead with 32 bits guests (VT-x only, some Intel CPUs don't support this feature (most do); bug #638)
- VMM: fixed 64 bits CentOS guest hangs during early boot (AMD-V only; bug #3927)
- VMM: performance improvements for certain PAE guests (e.g. Linux 2.6.29+ kernels)
- VMM: some Windows guests detected a completely wrong CPU frequency (bug #2227)
- VMM: fixed hanging and unkillable VM processes (bug #4040)
- VMM: fixed random infrequent guest crashes due XMM state corruption (Win64 hosts only)
- VMM: performance improvements for network I/O (VT-x/AMD-V only)

- GUI: added mini toolbar for full screen and seamless mode (Thanks to Huihong Luo)
- GUI: redesigned settings dialogs
- GUI: allow to create/remove more than one host-only network adapters (non Windows hosts)
- GUI: display estimated time for long running operations (e.g. OVF import/export)
- GUI: fixed rare hangs when open the OVF import/export wizards (bug #4157)
- 3D support: fixed VM crashes for client applications using incorrect OpenGL states
- 3D support: fixed memory corruption when querying for supported texture compression formats
- 3D support: fixed incorrect rendering of glDrawRangeElements
- 3D support: fixed memory leak when using VBOs
- 3D support: fixed glew library detection
- 3D support: fixed random textures corruption
- VRDP: support Windows 7 RDP client
- Networking: fixed another problem with TX checksum offloading with Linux kernels up to version 2.6.18
- NAT: fixed “open ports on virtual router 10.0.2.2 - 513, 514” (forum)
- NAT: allow to configure socket and internal parameters
- NAT: allow to bind sockets to specific interface
- PXE boot: significant performance increase (VT-x/AMD-V only)
- VHD: properly write empty sectors when cloning of VHD images (bug #4080)
- VHD: fixed crash when discarding snapshots of a VHD image
- VHD: fixed access beyond the block bitmap which could lead to arbitrary crashes
- VBoxManage: fixed incorrect partition table processing when creating VMDK files giving raw partition access (bug #3510)
- VBoxManage: support cloning to existing image file
- OVF: several OVF 1.0 compatibility fixes
- OVF: fixed exporting of disk images when multiple virtual machines are exported at once
- Virtual mouse device: eliminated micro-movements of the virtual mouse which were confusing some applications (bug #3782)
- Shared Folders: sometimes a file was created using the wrong permissions (2.2.0 regression; bug #3785)
- Shared Folders: allow to change file attributes from Linux guests and use the correct file mode when creating files
- Shared Folders: some content was incorrectly written under certain conditions (bug #1187)

- Shared Folders: fixed incorrect file timestamps, when using Windows guest on a Linux host (bug #3404)
- X11 clipboard: fix duplicate end of lines (bug #4270)
- X11 guests: a number of shared clipboard fixes
- Linux guests: Guest Additions support for SUSE Linux Enterprise Desktop 11
- Linux guests: new daemon vboxadd-service to handle time synchronization and guest property lookup
- Linux guests: implemented guest properties (OS info, logged in users, basic network information)
- Windows host installer: VirtualBox Python API can now be installed automatically (requires Python and Win32 Extensions installed)
- USB: Support for high-speed isochronous endpoints has been added. In addition, read-ahead buffering is performed for input endpoints (currently Linux hosts only). This should allow additional devices to work, notably webcams (bug #242)
- USB: fixed error handling for some USB dongles
- Web service: fixed inability to handle NULL pointers for object arguments, which are valid values for a lot of APIs, in both the raw and the object-oriented web service
- Web service: object-oriented bindings for JAX-WS did not exhibit interface inheritance correctly, fixed
- Web service: added support for IDisplay and IGuest interfaces, which were previously unavailable
- Registration dialog uses Sun Online accounts now

15.40 Version 2.2.4 (2009-05-29)

This is a maintenance release. The following items were fixed and/or added:

- Windows Installer: fixed a potential hang during installation
- Windows Installer: fixed several problems (bug #3892)
- Solaris hosts: make it work with Solaris build 114 or later (bug #3981)
- Solaris hosts: fixed a bug serial port character handling found during loopback (bug #3120)
- Linux hosts: adapted vboxdrv.sh to the latest changes in VBoxManage list runningvms (bug #4034)
- Windows hosts: fixed a crash caused by host-only/bridged networking
- Mac OS X hosts: fixed access to host DVD with passthrough disabled (bug #4077)
- Guest Additions: fixed problems with KDE 4 not recognizing mouse clicks
- Windows Additions: fixed incorrect 8-bit guest color depth in Windows 7 guests
- GUI: warn if VT-x/AMD-V could not be enabled for guests that require this setting (bug #4055)

- VMM: fixed occasional crash due to insufficient memory
- VMM: fixed hanging 64 bits Solaris guests
- VMM: restore from a saved state occasionally failed (bugs #3984 and #2742)
- Clipboard: fixed a deadlock while shutting down the shared clipboard on X11 hosts (bug #4020)
- OVF: fixed potential hang during import
- OVF: fixed potential crashes during import/export on Win64 hosts
- VBoxManage modifyhd --compact: fixed bug which could lead to crashes and image corruption (bug #3864)
- VBoxManage metrics collect: now flushes the output stream
- VHD: made VBoxManage internalcommands sethduuid work for .vhd files (bug #3443)
- VHD: some .vhd files could not be cloned (bug #4080)
- NAT: improvement of TCP connection establishment (bug #2987)
- NAT: fixed order of DNS servers in DHCP lease (bug #4091)
- NAT: fixed DHCP lease for multiple name servers (bug #3692)
- NAT: fixed a potential segfault if the host lost its connectivity (bug #3964)
- Shared Folders: deny access to parent directories on Windows hosts (bug #4090)
- Shared Folders: make rm/rmdir work with Solaris guests on Windows hosts
- Networking: fixed the problem with blocked receiving thread when a broadcast packet arrives too early to be handled by uninitialized e1000 adapter
- Networking: fixed the problem that caused host freezes/crashes when using bridged mode with host's interface having RX checksum offloading on (bug #3926 and related). Fixes problems with TX offloading as well (bug #3870)
- PXE boot: Added support for PRO/1000 MT Server adapter
- Python bindings: fixed keyword conflict
- SCSI: fixed occasional crashes on Win64
- Serial: allow to redirect the serial port to a raw file (bug #1023)
- VRDP: fixed a rare incorrect screen update
- VMDK: fixed creating snapshots

15.41 Version 2.2.2 (2009-04-27)

This is a maintenance release. The following items were fixed and/or added:

- Host and guest clipboard: fixed a number of issues affecting hosts and guests running the X window system
- Guest Additions: make sure the virtual mouse autodetection works on first reboot after installing the Additions on X.Org server 1.5 and later
- Guest Additions: properly report process identity number of running services
- Guest Additions: clean up properly if the X Window server terminates
- Linux Additions: fixed installation path for OpenGL libraries in some 64-bit guests (bug #3693)
- Solaris Additions: fixed installation to work when X.Org is not installed on the guest
- Solaris Additions: fixed a bug that could panic the guest when unmounting a busy shared folder
- Windows Additions: fixed mouse pointer integration of some Windows guests (2.2.0 regression, bug #3734)
- Windows Additions: fixed installation on Windows Server 2008 Core (bug #2628)
- Main: do not try to use older versions of D-Bus (Linux hosts only, bug #3732)
- VMM: fixed out-of-memory conditions on Windows hosts (bug #3657)
- VMM: fixed occasional hangs when attaching USB devices during VM startup (2.2.0 regression; bugs #3787)
- VMM: fixed guru meditation related to memory management (software virtualization only)
- Virtual disks: fix possible data corruption when writing to diff images, incorrect detection of redundant writes
- GUI: reworked network settings dialog
- GUI: properly show the detailed settings dialog of NAT networks (bug #3702)
- GUI: HostKey could not be changed (2.2.0 regression, bug #3689)
- GUI: fixed memory textfield size (Windows hosts only; bug #3679)
- GUI: fixed crash when selecting a shared folder path (Windows hosts only; bugs #3694, #3751, #3756)
- VBoxManage modifyhd --compact: implemented again for VDI files, and now supports relative paths (bug #2180, #2833)
- VBoxManage snapshot discard: made it work again (2.1.0 regression; bug #3714)
- NAT: on some Windows hosts, the guest didn't receive a DHCP lease (bug #3655)
- NAT: fixed release assertion during *poll()* (bug #3667)
- Networking: fixed a deadlock caused by the PCnet network device emulation (2.2.0 regression, bug #3676)

- Clipboard: fixed random crashes (X11 hosts only, bug #3723)
- Shared Folders: fixed incorrect permissions for Solaris guests
- Shared Folders: fixed wrong file sizes with Solaris guests
- CBindings: fixed possible memory leak while releasing the IVirtualBox and ISession Objects
- Solaris hosts: fixed host-only network interface incompatibility with nwam/dhcpagent (bug #3754)
- Windows installer: fixed several install and uninstall issues (bugs #3659, #3686, #1730, #3711, #3373, #3382, #3701, #3685, #3710)
- Mac OS X hosts: preliminary support for Snow Leopard

15.42 Version 2.2.0 (2009-04-08)

This version is a major update. The following major new features were added:

- OVF (Open Virtualization Format) appliance import and export (see chapitre [1.14, Importer et exporter des machines virtuelles](#), page 33)
- Host-only networking mode (see chapitre [6.6, Réseau Host-only](#), page 104)
- Hypervisor optimizations with significant performance gains for high context switching rates
- Raised the memory limit for VMs on 64-bit hosts to 16GB
- VT-x/AMD-V are enabled by default for newly created virtual machines
- USB (OHCI & EHCI) is enabled by default for newly created virtual machines (Qt GUI only)
- Experimental USB support for OpenSolaris hosts
- Shared Folders for Solaris and OpenSolaris guests
- OpenGL 3D acceleration for Linux and Solaris guests (see chapitre [4.4.1, Accélération 3D matérielle \(OpenGL and Direct3D 8/9\)](#), page 77)
- Added C API in addition to C++, Java, Python and Web Services

In addition, the following items were fixed and/or added:

- VMM: FreeBSD guest related fix for V86 flags (bug #2342)
- VMM: fixed guru meditation when booting an AsteriskNow Linux VM (bug #2342)
- VMM: fixed PGMPOOLKIND_FREE guru meditation (bugs #3356, #3431)
- VMM: fixed Windows XP boot hang (guest PAE + nested paging only)
- VMM: allow mixing of VT-x/AMD-V and software virtualization
- VMM: fixed extremely slow safe mode booting in e.g. Windows 2008 (VT-x/AMD-V only)
- VMM: significant speedup of certain GRUB boot loaders (e.g. Solaris) (VT-x/AMD-V only)
- VMM: real-mode IOPL fix for DOS guests (VT-x only)

- VMM: fixed VT-x detection with certain BIOSes that enable VT-x, but don't set the lock bit in MSR_IA32 FEATURE CONTROL
- VMM: fixed hibernation issues on Windows XP hosts (VT-x only; bug #1794)
- VMM: properly emulate RDMSR from the TSC MSR, should fix some NetBSD guests
- VMM: emulate RDPMC; fixes Windows guests crashes when using the Kaspersky virus scanner (bug #1778)
- NAT: fixed truncated downloads (FTP) (bug #3257)
- NAT: blocked UDP packets caused a crash (bug #3426)
- NAT: allow to configure the *next server* and the *boot file* via VBoxManage (bug #2759)
- IDE: fixed hard disk upgrade from XML-1.2 settings (bug #1518)
- Hard disk: support more VMDK file variants (including fixed-size ESX server images)
- Hard disks: refuse to start the VM if a disk image is not writable
- USB: further reduced host CPU utilization for OHCI and EHCI; the "VBoxInternal/Devices/usb-ohci/0/Config/FrameRate" CFG key is no longer necessary and no longer supported
- USB: fixed BSOD on the host with certain USB devices (Windows hosts only; bug #1654)
- E1000: properly handle cable disconnects (bug #3421)
- VRDP: fixed hangs when VRDP server is enabled or disabled in runtime
- Shared Folders: respect umask settings on Linux, OSX and Solaris hosts when creating files
- X11 guests: prevented setting the locale in vboxmouse, as this caused problems with Turkish locales (bug #3563)
- X11 guests: show the guest mouse pointer at the right position if the virtual desktop is larger than the guest resolution (bug #2306)
- Linux Additions: fixed typo when detecting Xorg 1.6 (bug #3555)
- Solaris guests: added xpg4/xcu4 dependency to the Guest Additions installer (bug #3524)
- Windows guests: bind the VBoxMouse.sys filter driver to the correct guest pointing device (bug #1324)
- Windows hosts: fixed BSOD when starting a VM with enabled host interface (bug #3414)
- Linux hosts: do proper reference counting to prevent unloading the vboxnetflt module as long as this code is in use (bug #3104)
- Linux hosts: do not leave zombies of VBoxSysInfo.sh (bug #3586)
- Linux installers: fixes for Slackware, Arch Linux and Linux from Scratch systems
- Windows installers: combined installer executable which contains both (32- and 64-bit) architectures
- VBoxManage: less cryptic command-line error messages
- VBoxManage list vms commands now default to compact format
- VBoxManage controlvm dvdattach did not work if the image was attached before

- VBoxManage: allow creation of all supported disk image variants
- VBoxManage showvminfo: don't spam the release log if the Guest Additions don't support statistics information (bug #3457)
- VBoxManage: big command line processing cleanup, the legacy single-dash options are deprecated and will be removed in the next major release, so switch to the new options now
- Hard disks: improved immutable disk support to auto-reset diff file at VM startup (related to bug #2772)
- GUI: enable the audio adapter by default for new VMs
- GUI: warn if VT-x/AMD-V is not operational when starting a 64-bit guest
- GUI: deactivate 64-bit guest support when the host CPU does not support VT-x/AMD-V
- GUI: removed floppy icon from the status bar
- GUI: show build revision in about dialog
- GUI: fixed sticky status bar text
- GUI: improved error dialogs
- GUI: fail with an appropriate error message when trying to boot a read-only disk image (bug #1745)
- GUI/Mac OS X: fixed disabled close button
- GUI/Windows: re-enabled support for copy and paste (Windows hosts 2.0 regression; bug #2065)
- 3D support: added OpenGL select/feedback support (bug #2920)
- 3D support: close OpenGL subsystem for terminated guest applications (bug #3243)
- 3D support: fixed VM hangs when starting guests with 3D acceleration enabled (bug #3437)
- PXE: fixed boot hangs when hardware virtualization is used (bug #2536)
- LsiLogic: fixed problems with Solaris guests
- Main API: close machine settings XML file when unregistering machine (bug #3548)

15.43 Version 2.1.4 (2009-02-16)

This is a maintenance release. The following items were fixed and/or added:

- Windows hosts: fixed host crashes/hangs on certain 32 bits Windows systems when running Linux guests (bugs #1606, #2269, #2763)
- Windows hosts: fixed network component BSOD issue (bugs #3168, #2916)
- Windows hosts: fixed installation issues (bugs #2517, #1730, #3130)
- Linux hosts: fixed occasional kernel oopses (bug #2556)
- Linux hosts: fixed module dependency for shipped modules (bug #3115)

- Linux hosts: moved the udev rules for USB forward so that they don't override existing system rules (bug #3143)
- Linux hosts: fixed the issue with guest not being able to communicate with each other when attached via TAP interfaces (bug #3215)
- Linux hosts: give up probing for USB gracefully if DBus or hal are not available (bug #3136)
- Linux hosts: fixed warnings in installer when SELinux was disabled (bug #3098)
- Linux hosts: VirtualBox sometimes failed to start if it had been started using sudo previously (bug #3270)
- Solaris hosts: fixed high CPU load while running many guests in parallel
- Solaris hosts: fixed inability to start more than 128 VMs
- VMM: fixed performance regression for Windows guests (bug #3172)
- VMM: ignore CPU stepping when restoring a saved state/snapshot
- REM: fixed inability to use gdb to debug programs in Linux guests with software virtualization (bug #3245)
- GUI: fixed dead key handling on Solaris hosts (bug #3256)
- GUI: in the shutdown dialog, disable the action *send the shutdown signal* if the guest is currently not using ACPI
- GUI: suppress additional key release events sent by X11 hosts when keys are auto-repeated (bug #1296)
- API: restore case insensitive OS type name lookup (bug #3087)
- VBoxHeadless: really don't start X11 services (clipboard service, 3D acceleration; Solaris & Darwin hosts only; bug #3199)
- NAT: fixed occasional crashes when the guest is doing traceroute (non-Windows hosts; bug #3200)
- NAT: fixed crashes under high load (bug #3110)
- NAT: fixed truncated downloads (Windows hosts only, bug #3257)
- NAT: don't intercept TFTP packages with a destination address different from the builtin TFTP server (bug #3112)
- USB: several fixes for USB passthrough on Linux hosts
- USB: reduced host CPU utilization if EHCI is active
- VRDP: fixed VRDP server black screen after a client reconnect (bug #1989)
- VRDP: modified rdesktop client (rdesktop-vrdp) now uses NumLock state synchronization (bug #3253)
- LsiLogic: make FreeBSD guests work (bug #3174)
- ATA: fixed deadlock when pausing VM due to problems with the virtual disk (e.g. disk full, iSCSI target unavailable)
- iSCSI: fixed possible crash when pausing the VM

- 3D support: added missing GL_MAX_TEXTURE_COORDS_ARB (bug #3246)
- Windows Additions: fixed *ERROR (e0000101)* error during installation (bug #1923)
- Windows Additions: fixed Windows Explorer hang when browsing shared folders with 64 bit guests (bug #2225)
- Windows Additions: fixed guest screen distortions during a video mode change
- Windows Additions: fixed the *Network drive not connected* message for mapped shared folders drives after the guest startup (bug #3157)
- Linux Additions: fixed occasional file corruption when writing files in *O_APPEND* mode to a shared folder (bug #2844)
- Linux Additions: the mouse driver was not properly set up on X.Org release candidates (bug #3212)
- Linux Additions: fixed installer to work with openSUSE 11.1 (bug #3213)
- Linux Additions: disable dynamic resizing if the X server is configured for fixed resolutions
- Linux/Solaris Additions: handle virtual resolutions properly which are larger than the actual guest resolution (bug #3096)

15.44 Version 2.1.2 (2009-01-21)

This is a maintenance release. The following items were fixed and/or added:

- USB: Linux host support fixes (bug #3136)
- VMM: fixed guru meditation for PAE guests on non-PAE hosts (AMD-V)
- VMM: fixed guru meditation on Mac OS X hosts when using VT-x
- VMM: allow running up to 1023 VMs on 64-bit hosts (used to be 127)
- VMM: several FreeBSD guest related fixes (bugs #2342, #2341, #2761)
- VMM: fixed guru meditation when installing Suse Enterprise Server 10U2 (VT-x only; bug #3039)
- VMM: fixed guru meditation when booting Novell Netware 4.11 (VT-x only; bug #2898)
- VMM: fixed VERR_ADDRESS_TOO_BIG error on some Mac OS X systems when starting a VM
- VMM: clear MSR_K6_EFER_SVME after probing for AMD-V (bug #3058)
- VMM: fixed guru meditation during Windows 7 boot with more than 2 GB guest RAM (VT-x, nested paging only)
- VMM: fixed hang during OS/2 MCP2 boot (AMD-V and VT-x only)
- VMM: fixed loop during OpenBSD 4.0 boot (VT-x only)
- VMM: fixed random crashes related to FPU/XMM with 64 bits guests on 32 bits hosts
- VMM: fixed occasional XMM state corruption with 64 bits guests
- GUI: raised the RAM limit for new VMs to 75% of the host memory

- GUI: added Windows 7 as operating system type
- VBoxSDL: fixed `-fixed fixedmode` parameter (bug #3067)
- Clipboard: stability fixes (Linux and Solaris hosts only, bug #2675 and #3003)
- 3D support: fixed VM crashes for certain guest applications (bugs #2781, #2797, #2972, #3089)
- LsiLogic: improved support for Windows guests (still experimental)
- VGA: fixed a 2.1.0 regression where guest screen resize events were not properly handled (bug #2783)
- VGA: significant performance improvements when using VT-x/AMD-V on Mac OS X hosts
- VGA: better handling for VRAM offset changes (fixes GRUB2 and Dos DOOM display issues)
- VGA: custom VESA modes with invalid widths are now rounded up to correct ones (bug #2895)
- IDE: fixed ATAPI passthrough support (Linux hosts only; bug #2795)
- Networking: fixed kernel panics due to NULL pointer dereference in Linux kernels < 2.6.20 (Linux hosts only; bug #2827)
- Networking: fixed intermittent BSODs when using the new host interface (Windows hosts only; bugs #2832, #2937, #2929)
- Networking: fixed several issues with displaying hostif NICs in the GUI (Windows hosts only; bugs 2814, #2842)
- Networking: fixed the issue with displaying hostif NICs without assigned IP addresses (Linux hosts only; bug #2780)
- Networking: fixed the issue with sent packets coming back to internal network when using hostif (Linux hosts only; bug #3056).
- NAT: fixed port forwarding (Windows hosts only; bug #2808)
- NAT: fixed booting from the builtin TFTP server (bug #1959)
- NAT: fixed occasional crashes (bug #2709)
- SATA: vendor product data (VPD) is now configurable
- SATA: raw disk partitions were not recognized (2.1.0 regression, Windows host only, bug #2778)
- SATA: fixed timeouts in the guest when using raw VMDK files (Linux host only, bug #2796)
- SATA: huge speed up during certain I/O operations like formatting a drive
- SATA/IDE: fixed possible crash/errors during VM shutdown
- VRDP: fixed loading of `libpam.so.1` from the host (Solaris hosts only)
- VRDP: fixed RDP client disconnects
- VRDP: fixed VRDP server misbehavior after a broken client connection
- VBoxManage showvminfo: fixed assertion for running VMs (bug #2773)

- `VBoxManage convertfromraw`: added parameter checking and made it default to creating VDI files; fixed and documented format parameter (bug #2776)
- `VBoxManage clonehd`: fixed garbled output image when creating VDI files (bug #2813)
- `VBoxManage guestproperty`: fixed property enumeration (incorrect parameters/exception)
- `VHD`: fixed error when attaching certain container files (bug #2768)
- Solaris hosts: added support for serial ports (bug #1849)
- Solaris hosts: fix for Japanese keyboards (bug #2847)
- Solaris hosts: 32-bit and 64-bit versions now available as a single, unified package
- Linux hosts: don't depend on `libcap1` anymore (bug #2859)
- Linux hosts: kernel module compile fixes for 2.6.29-rc1
- Linux hosts: don't drop any capability if the VM was started by root (2.1.0 regression)
- Mac OS X hosts: save the state of running or paused VMs when the host machine's battery reaches critical level
- Mac OS X hosts: improved window resizing of the VM window
- Mac OS X hosts: added GUI option to disable the dock icon realtime preview in the GUI to decrease the host CPU load when the guest is doing 3D
- Mac OS X hosts: polished realtime preview dock icon
- Windows Additions: fixed guest property and logging OS type detection for Windows 2008 and Windows 7 Beta
- Windows Additions: added support for Windows 7 Beta (bugs #2995, #3015)
- Windows Additions: fixed Windows 2000 guest freeze when accessing files on shared folders (bug #2764)
- Windows Additions: fixed CTRL-ALT-DEL handling when using VBoxGINA
- Windows Additions Installer: added `/extract` switch to only extract (not install) the files to a directory (can be specified with `/D=path`)
- Linux installer and Additions: added support for the Linux From Scratch distribution (bug #1587) and recent Gentoo versions (bug #2938)
- Additions: added experimental support for X.Org Server 1.6 RC on Linux guests
- Linux Additions: fixed bug which prevented to properly set `fmode` on mapped shared folders (bug #1776)
- Linux Additions: fixed appending of files on shared folders (bug #1612)
- Linux Additions: ignore `noauto` option when mounting a shared folder (bug #2498)
- Linux Additions: fixed a driver issue preventing X11 from compiling keymaps (bug #2793 and #2905)
- X11 Additions: workaround in the mouse driver for a server crash when the driver is loaded manually (bug #2397)

15.45 Version 2.1.0 (2008-12-17)

This version is a major update. The following major new features were added:

- Support for hardware virtualization (VT-x and AMD-V) on Mac OS X hosts
- Support for 64-bit guests on 32-bit host operating systems (experimental; see chapitre 3.1.2, *Invités 64 bits*, page 49)
- Added support for Intel Nehalem virtualization enhancements (EPT and VPID; see chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201)
- Experimental 3D acceleration via OpenGL (see chapitre 4.4.1, *Accélération 3D matérielle (OpenGL and Direct3D 8/9)*, page 77)
- Experimental LsiLogic and BusLogic SCSI controllers (see chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85)
- Full VMDK/VHD support including snapshots (see chapitre 5.2, *Fichiers images de disque (VDI, VMDK, VHD, HDD)*, page 88)
- New NAT engine with significantly better performance, reliability and ICMP echo (ping) support (bugs #1046, #2438, #2223, #1247)
- New Host Interface Networking implementations for Windows and Linux hosts with easier setup (replaces TUN/TAP on Linux and manual bridging on Windows)

In addition, the following items were fixed and/or added:

- VMM: significant performance improvements for VT-x (real mode execution)
- VMM: support for hardware breakpoints (VT-x and AMD-V only; bug #477)
- VMM: VGA performance improvements for VT-x and AMD-V
- VMM: Solaris and OpenSolaris guest performance improvements for AMD-V (Barcelona family CPUs only)
- VMM: fixed guru meditation while running the Dr. Web virus scanner (software virtualization only; bug #1439)
- VMM: deactivate VT-x and AMD-V when the host machine goes into suspend mode; reactivate when the host machine resumes (Windows, Mac OS X & Linux hosts; bug #1660)
- VMM: fixed guest hangs when restoring VT-x or AMD-V saved states/snapshots
- VMM: fixed guru meditation when executing a one byte debug instruction (VT-x only; bug #2617)
- VMM: fixed guru meditation for PAE guests on non-PAE hosts (VT-x)
- VMM: disallow mixing of software and hardware virtualization execution in general (bug #2404)
- VMM: fixed black screen when booting OS/2 1.x (AMD-V only)
- GUI: pause running VMs when the host machine goes into suspend mode (Windows & Mac OS X hosts)
- GUI: resume previously paused VMs when the host machine resumes after suspend (Windows & Mac OS X hosts)

- GUI: save the state of running or paused VMs when the host machine's battery reaches critical level (Windows hosts)
- GUI: properly restore the position of the selector window when running on the compiz window manager
- GUI: properly restore the VM in seamless mode (2.0 regression)
- GUI: warn user about non optimal memory settings
- GUI: structure operating system list according to family and version for improved usability
- GUI: predefined settings for QNX guests
- IDE: improved ATAPI passthrough support
- Networking: added support for up to 8 Ethernet adapters per VM
- Networking: fixed issue where a VM could lose connectivity after a reboot
- iSCSI: allow snapshot/diff creation using local VDI file
- iSCSI: improved interoperability with iSCSI targets
- Graphics: fixed handling of a guest video memory which is not a power of two (bug #2724)
- VBoxManage: fixed bug which prevented setting up the serial port for direct device access
- VBoxManage: added support for VMDK and VHD image creation
- VBoxManage: added support for image conversion (VDI/VMDK/VHD/RAW)
- Solaris hosts: added IPv6 support between host and guest when using host interface networking
- Mac OS X hosts: added ACPI host power status reporting
- API: redesigned storage model with better generalization
- API: allow attaching a hard disk to more than one VM at a time
- API: added methods to return network configuration information of the host system
- Shared Folders: performance and stability fixes for Windows guests (Microsoft Office Applications)

15.46 Version 2.0.8 (2009-03-10)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed guest hangs when restoring VT-x or AMD-V saved states/snapshots
- VMM: fixed memory allocation issues which can cause VM start failures with VERR_PGM_MAPPING_CONFLICT error
- VMM: fixed host crashes/hangs on certain 32 bits Windows systems when running Linux guests (bugs #1606, #2269, #2763)
- XPCOM/Main: fixed synchronization bug caused by SYSV semaphore key collisions
- ATA: fixed deadlock when pausing VM due to problems with the virtual disk (e.g. disk full, iSCSI target unavailable)

- iSCSI: fixed possible crash when pausing the VM
- iSCSI: fix PDU validity checking and detect final PDU reliably
- VBoxHeadless: really don't start X11 services (clipboard service, 3D acceleration; Solaris & Darwin hosts only; bug #3199)
- Networking: fixed issue where a VM could lose connectivity after a reboot
- Linux hosts: fixed occasional kernel oopses (bug #2556)
- Solaris hosts: fixed high CPU load while running many guests in parallel
- Solaris hosts: fixed inability to start more than 128 VMs
- Solaris/Web services: fixed SMF script to set home directory correctly
- Linux Additions: fixed occasional file corruption when writing files in *O_APPEND* mode to a shared folder (bug #2844)

15.47 Version 2.0.6 (2008-11-21)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed Guru meditation when running 64 bits Windows guests (bug #2220)
- VMM: fixed Solaris 10U6 boot hangs (VT-x and AMD-V) bug #2565)
- VMM: fixed Solaris 10U6 reboot hangs (AMD-V only; bug #2565)
- GUI: the host key was sometimes not properly displayed (Windows hosts only, bug #1996)
- GUI: the keyboard focus was lost after minimizing and restoring the VM window via the Windows taskbar (bugs #784)
- VBoxManage: properly show SATA disks when showing the VM information (bug #2624)
- SATA: fixed access if the buffer size is not sector-aligned (bug #2024)
- SATA: improved performance
- SATA: fixed snapshot function with ports>1 (bug #2510)
- E1000: fixed crash under rare circumstances
- USB: fixed support for iPhone and Nokia devices (Linux host: bugs #470 & #491)
- Windows host installer: added proper handling of open VirtualBox applications when updating the installation
- Windows host installer: fixed default installation directory on 64-bit on new installations (bug #2501)
- Linux/Solaris/Darwin hosts: verify permissions in `/tmp/vbox-$USER-ipc`
- Linux hosts: fixed assertion on high network load (AMD64 hosts, fix for Linux distributions with glibc 2.6 and newer (bug #616)
- Linux hosts: don't crash during shutdown with serial ports connected to a host device
- Solaris hosts: fixed incompatibility between IPSEC and host interface networking

- Solaris hosts: fixed a rare race condition while powering off VMs with host interface networking
- Solaris hosts: fixed VBoxSDL on Solaris 10 by shipping the required SDL library (bug #2475)
- Windows Additions: fixed logged in users reporting via guest properties when using native RDP connections
- Windows Additions: fixed Vista crashes when accessing shared folders under certain circumstances (bug #2461)
- Windows Additions: fixed shared folders access with MS-Office (bug #2591)
- Linux Additions: fixed compilation of vboxvfs.ko for 64-bit guests (bug #2550)
- SDK: added JAX-WS port caching to speedup connections

15.48 Version 2.0.4 (2008-10-24)

This is a maintenance release. The following items were fixed and/or added:

- VMM: better error reporting for VT-x failures
- VMM: don't overflow the release log with PATM messages (bug #1775)
- VMM: fixed save state restore in real mode (software virtualization only)
- GUI: work around a Qt bug on Mac OS X (bug #2321)
- GUI: properly install the Qt4 accessible plugin (bug #629)
- SATA: error message when starting a VM with a VMDK connected to a SATA port (bug #2182)
- SATA: fixed Guru mediation when booting OpenSolaris/64; most likely applies to other guests as well (bug #2292)
- Network: don't crash when changing the adapter link state if no host driver is attached (bug #2333)
- VHD: fixed bug which prevents booting from VHD images bigger than 4GB (bug #2085)
- VRDP: fixed a repaint problem when the guest resolution was not equal to the client resolution
- Clipboard: don't crash when host service initialization takes longer than expected (Linux hosts only; bug #2001)
- Windows hosts: VBoxSVC.exe crash (bug #2212)
- Windows hosts: VBoxSVC.exe memory leak due to a Windows WMI memory leak (Vista only) (bug #2242)
- Windows hosts: VBoxSVC.exe delays GUI startup
- Linux hosts: handle jiffies counter overflow (VM stuck after 300 seconds of host uptime; bug #2247)
- Solaris hosts: fixed host or guest side networking going stale while using host interface networking (bug #2474)

- Solaris hosts: added support for using unplumbed network interfaces and Crossbow Virtual Network Interfaces (VNICs) with host interface networking
- Solaris hosts: reworked threading model improves performance for host interface networking
- Windows Additions: fixed crash when accessing deep directory structures in a shared folder
- Windows Additions: improved shared folder name resolving (bug #1728)
- Windows Additions: fixed Windows 2000 shutdown crash (bug #2254)
- Windows Additions: fixed error code for `MoveFile()` if the target exists (bug #2350)
- Linux Additions: fixed `seek()` for files bigger than 2GB (bug #2379)
- Linux Additions: support Ubuntu 8.10
- Linux Additions: clipboard fixes (bug #2015)
- Web services: improved documentation and fixed example (bug #1642)

15.49 Version 2.0.2 (2008-09-12)

This is a maintenance release. The following items were fixed and/or added:

- VMM: fixed inability to run more than one VM in parallel (AMD-V on CPUs with erratum 170 only; bug #2167)
- VMM: VT-x stability fixes (bug #2179 and others)
- VMM: fixed Linux 2.6.26+ kernel crashes (used by Ubuntu 8.10 Alpha, Fedora 10 Alpha; bug #1875)
- VMM: fixed 64 bits Linux 2.6.26 kernel crashes (Debian)
- VMM: fixed Vista (32 bits) guest crash during boot when PAE and NX are enabled (applied to 64 bits hosts with VT-x enabled only)
- VMM: fixed OS/2 guest crashes during boot (AMD-V; bug #2132)
- GUI: fixed crash when trying to release an inaccessible image in the virtual disk manager
- GUI: fixed invalid error message for a changed snapshot path even if that path wasn't changed (bug #2064)
- GUI: fixed crash when creating a new hard disk image (bug #2060)
- GUI: fixed crash when adding a hard disk in the VM settings (bug #2081)
- GUI: fixed a bug where VirtualBox isn't working with the new QGtkStyle plugin (bug #2066)
- GUI: fixed VM close dialog in seamless mode (Mac OS X hosts only; bug #2067)
- GUI: fixed standard menu entries for NLS versions (Mac OS X hosts only)
- GUI: disable the VT-x/AMD-V setting when it's not supported by the CPU (or on Mac OS X hosts)
- VBoxManage: fixed crash during `internalcommands createrawvmdk` (bug #2184)

- VBoxManage: fixed output of `snapshot showvminfo` (bug #698)
- Guest properties: added information about guest network interfaces (Windows guests only)
- Shared Folders: fixed regression that caused Windows guest crashes
- API: fixed number of installed CPUs (Solaris hosts only)
- VRDP: allow a client to reconnect to an existing session on the VRDP server by dropping the existing connection (configurable and disabled by default; only relevant when multi-connection mode is disabled)
- VRDP: fixed an image repaint problem
- Linux hosts: fixed bug in `vboxdrv.ko` that could corrupt kernel memory and panic the kernel (bug #2078)
- Linux hosts: compile fixes for kernel module on Linux 2.6.27
- Mac OS X hosts: added Python support
- Additions: fixed a possible hang in HGCM communication after a VM reboot
- Windows Additions: added support for Windows XP 64 bits (bug #2117)
- Linux Additions: deactivate dynamic resizing on Linux guests with buggy X servers
- Linux Additions: support Ubuntu 8.10 guests and Fedora 9 guests (dynamic resizing disabled for the latter)
- Linux Additions: added installer check for the system architecture
- Linux Additions: fixed Xorg modules path for some Linux distributions (bug #2128)
- VMDK: be more liberal with ambiguous parts of the format specification and accept more format variants (bug #2062)
- VHD: fixed a bug in the VHD backend which resulted in reading the wrong data (bug #2085)
- Solaris hosts: fixed kernel panic on certain machines when starting VMs with host interface networking (bug #2183)
- Solaris hosts: fixed inability to access NFS shares on the host when host interface networking was enabled
- Solaris hosts: installer now detects and reports when installing under the wrong architecture
- Solaris hosts: fixed security hardening that prevented starting VMs from non-global zones even as root (bug #1948)
- Solaris Additions: combined the 32 bit and 64 bit Additions installer into a single package
- Mac OS X hosts: experimental support for attaching a real serial port to the guest

15.50 Version 2.0.0 (2008-09-04)

This version is a major update. The following major new features were added:

- 64 bits guest support (64 bits host only)
- New native Leopard user interface on Mac OS X hosts
- The GUI was converted from Qt3 to Qt4 with many visual improvements
- New-version notifier
- Guest property information interface
- Host Interface Networking on Mac OS X hosts
- New Host Interface Networking on Solaris hosts
- Support for Nested Paging on modern AMD CPUs (major performance gain)
- Framework for collecting performance and resource usage data (metrics)
- Added SATA asynchronous IO (NCQ: Native Command Queuing) when accessing raw disks/partitions (major performance gain)
- Clipboard integration for OS/2 Guests
- Created separate SDK component featuring a new Python programming interface on Linux and Solaris hosts
- Support for VHD disk images

In addition, the following items were fixed and/or added:

- VMM: VT-x fixes
- AHCI: improved performance
- GUI: keyboard fixes
- Linux installer: properly uninstall the package even if unregistering the DKMS module fails
- Linux Additions: the guest screen resolution is properly restored
- Network: added support for jumbo frames (> 1536 bytes)
- Shared Folders: fixed guest crash with Windows Media Player 11
- Mac OS X: Ctrl+Left mouse click doesn't simulate a right mouse click in the guest anymore. Use Hostkey+Left for a right mouse click emulation. (bug #1766)

Avec VirtualBox 3.2, les informations d'historique des changements pour les versions inférieures à la 2.0 ont été supprimées pour gagner de la place. Pour accéder à ces informations, merci de consulter le manuel de l'utilisateur de VirtualBox version 3.1 ou inférieur.

16 Matériaux tiers et licences

VirtualBox englobe des matériaux de plusieurs logiciels Open Source. Donc, l'utilisation de ces matériaux par VirtualBox est soumise à des licences Open Source. Ce document reproduit ces licences et fournit une liste des matériaux utilisés avec leurs conditions de licences respectives. La section 1 contient une liste des matériaux utilisés. La section 2 reproduit les licences Open Source applicables. Pour chaque matériaux, nous fournissons une référence à sa licence.

Le code source des matériaux listés ci-dessous ainsi que le reste du code de VirtualBox publiés sous licence open source sont disponibles sur <http://www.virtualbox.org>, sous forme d'archives tar pour des versions en particulier et d'un dépôt SVN en direct.

16.1 Matériaux tiers

- VirtualBox contient des morceaux de QEMU soumis aux licences de chapitre 16.2.5, *X Consortium License (X11)*, page 322 et chapitre 16.2.2, *GNU Lesser General Public License (LGPL)*, page 311 et
 - (C) 2003-2005 Fabrice Bellard; Copyright (C) 2004-2005 Vassili Karpov (malc); Copyright (c) 2004 Antony T Curtis; Copyright (C) 2003 Jocelyn Mayer
- VirtualBox contient du code sous licence de chapitre 16.2.5, *X Consortium License (X11)*, page 322 et
 - Copyright 2004 par the Massachusetts Institute of Technology.
- VirtualBox contient du code du BIOS VGA BOCHS soumis [la licence sur chapitre 16.2.2, *GNU Lesser General Public License (LGPL)*, page 311 et
 - Copyright (C) 2001, 2002 L'équipe des développeurs LGPL VGABios.
- VirtualBox contient du code du BIOS BOCHS ROM soumis à la licence de chapitre 16.2.2, *GNU Lesser General Public License (LGPL)*, page 311 et
 - Copyright (C) 2002 MandrakeSoft S.A.; Copyright (C) 2004 Fabrice Bellard; Copyright (C) 2005 Struan Bartlett.
- VirtualBox contient la bibliothèque zlib soumise à la licence de chapitre 16.2.6, *zlib license*, page 322 et
 - Copyright (C) 1995-2003 Jean-loup Gailly et Mark Adler.
- VirtualBox peut contenir OpenSSL soumis à la licence dans chapitre 16.2.7, *OpenSSL license*, page 323 et
 - Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com). Ce produit inclut un logiciel écrit par Tim Hudson (tjh@cryptsoft.com).
- VirtualBox peut contenir NSPR et XPCOM soumis à la licence de chapitre 16.2.3, *Mozilla Public License (MPL)*, page 316 et
 - Copyright (C) Les auteurs.
- VirtualBox contient Slirp soumis à la licence de chapitre 16.2.8, *Slirp license*, page 323 et qui a été écrit par Danny Gasparovski.
 - Copyright (C) 1995, 1996 Tous droits réservés.

- VirtualBox contient liblzf which is governed by the license in chapitre 16.2.9, *liblzf license*, page 324 and
Copyright (C) 2000-2005 Marc Alexander Lehmann <schmorp@schmorp.de>
- VirtualBox may ship with a modified copy of rdesktop which is governed by the license in chapitre 16.2.1, *GNU General Public License (GPL)*, page 307 and
Copyright (C) Matthew Chapman and others.
- VirtualBox may ship with a copy of kchmviewer which is governed by the license in chapitre 16.2.1, *GNU General Public License (GPL)*, page 307 and
Copyright (C) George Yunaev and others.
- VirtualBox may contain Etherboot which is governed by the license in chapitre 16.2.1, *GNU General Public License (GPL)*, page 307 with the exception that aggregating Etherboot with another work does not require the other work to be released under the same license (see <http://etherboot.sourceforge.net/clinks.html>). Etherboot is
Copyright (C) Etherboot team.
- VirtualBox contient code from Wine which is governed by the license in chapitre 16.2.2, *GNU Lesser General Public License (LGPL)*, page 311 and
Copyright 1993 Bob Amstadt, Copyright 1996 Albrecht Kleine, Copyright 1997 David Faure, Copyright 1998 Morten Welinder, Copyright 1998 Ulrich Weigand, Copyright 1999 Ove Koven
- VirtualBox contient code from lwIP which is governed by the license in chapitre 16.2.11, *lwIP license*, page 325 and
Copyright (C) 2001, 2002 Swedish Institute of Computer Science.
- VirtualBox contient libxml which is governed by the license in chapitre 16.2.12, *libxml license*, page 325 and
Copyright (C) 1998-2003 Daniel Veillard.
- VirtualBox contient libxslt which is governed by the license in chapitre 16.2.13, *libxslt licenses*, page 325 and
Copyright (C) 2001-2002 Daniel Veillard and Copyright (C) 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard.
- VirtualBox contient code from the gSOAP XML web services tools, which are licensed under the license in chapitre 16.2.14, *gSOAP Public License Version 1.3a*, page 326 and
Copyright (C) 2000-2007, Robert van Engelen, Genivia Inc., and others.
- VirtualBox ships with the application tunctl (shipped as VBoxTunctl) from the User-mode Linux suite which is governed by the license in chapitre 16.2.1, *GNU General Public License (GPL)*, page 307 and
Copyright (C) 2002 Jeff Dike.
- VirtualBox contient code from Chromium, an OpenGL implementation, which is governed by the licenses in chapitre 16.2.15, *Chromium licenses*, page 331 and
Copyright (C) Stanford University, The Regents of the University of California, Red Hat, and others.
- VirtualBox contient libcurl which is governed by the license in chapitre 16.2.16, *curl license*, page 333 and
Copyright (C) 1996-2009, Daniel Stenberg.

- VirtualBox contient dnsproxy which is governed by the license in chapitre [16.2.4, MIT License](#), page [322](#) and
Copyright (c) 2003, 2004, 2005 Armin Wolfermann.
- VirtualBox may contain iniparser which is governed by the license in chapitre [16.2.4, MIT License](#), page [322](#) and
Copyright (c) 2000-2008 by Nicolas Devillard.
- VirtualBox contient some code from libgd which is governed by the license in chapitre [16.2.17, libgd license](#), page [333](#) and
Copyright 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Pierre-Alain Joye (pierre@libgd.org).
- VirtualBox contient code from the EFI Development Kit II which is governed by the license in chapitre [16.2.18, BSD license from Intel](#), page [334](#) and
Copyright (c) 2004-2008, Intel Corporation.
- VirtualBox contient libjpeg soumis à la licence dans chapitre [16.2.19, libjpeg License](#), page [334](#) et
Copyright (C) 1991-2010, Thomas G. Lane, Guido Vollbeding.
- VirtualBox peut contenir l'extension x86 SIMD de la bibliothèque IJG JPEG, soumis à la licence dans chapitre [16.2.20, x86 SIMD extension for IJG JPEG library license](#), page [335](#) et
Copyright 2009 Pierre Ossman <ossman@cendio.se> pour Cendio AB; Copyright 2010 D. R. Commander; Copyright (C) 1999-2006, MIYASAKA Masaru.
- VirtualBox peut inclure une copie de Qt soumis à la licence de chapitre [16.2.2, GNU Lesser General Public License \(LGPL\)](#), page [311](#) et
Copyright (C) 2010, 2011 Nokia Corporation et/ou sa/ses filiale(s).
- VirtualBox contient ces morceaux du noyau FreeBSD soumis à la licence dans chapitre [16.2.21, FreeBSD license](#), page [336](#).
- VirtualBox contient des morceaux du noyau NetBSD soumis à la licence dans chapitre [16.2.22, NetBSD license](#), page [336](#).

16.2 Licences

16.2.1 GNU General Public License (GPL)

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software

(and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under

these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND

PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

16.2.2 GNU Lesser General Public License (LGPL)

GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the “Lesser” General Public License because it does less to protect the user’s freedom than the ordinary General Public License. It also provides other free software developers less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is less protective of the users’ freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a “work based on the library” and a “work that uses the library”. The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called “this License”). Each licensee is addressed as “you”.

A “library” means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The “Library”, below, refers to any such software library or work which has been distributed under these terms. A “work based on the Library” means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term “modification”.)

“Source code” for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.) Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer’s own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user’s computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients’ exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the

author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

16.2.3 Mozilla Public License (MPL)

MOZILLA PUBLIC LICENSE Version 1.1

1. Definitions.

1.0.1. “Commercial Use” means distribution or otherwise making the Covered Code available to a third party.

1.1. “Contributor” means each entity that creates or contributes to the creation of Modifications.

1.2. “Contributor Version” means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. “Covered Code” means the Original Code or Modifications or the combination of the Original Code and Modifications, in each case including portions thereof.

1.4. “Electronic Distribution Mechanism” means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. “Executable” means Covered Code in any form other than Source Code.

1.6. “Initial Developer” means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. “Larger Work” means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. “License” means this document.

1.8.1. “Licensable” means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. “Modifications” means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. “Original Code” means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. “Patent Claims” means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. “Source Code” means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor’s choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. “You” (or “Your”) means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, “You” includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, “control” means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant. The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are effective on the date Initial Developer first distributes Original Code under the terms of this License.

(d) Notwithstanding Section 2.1(b) above, no patent license is granted: 1) for code that You delete from the Original Code; 2) separate from the Original Code; or 3) for infringements caused by: i) the modification of the Original Code or ii) the combination of the Original Code with other software or devices.

2.2. Contributor Grant. Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

(c) the licenses granted in Sections 2.2(a) and 2.2(b) are effective on the date Contributor first makes Commercial Use of the Covered Code.

(d) Notwithstanding Section 2.2(b) above, no patent license is granted: 1) for any code that Contributor has deleted from the Contributor Version; 2) separate from the Contributor Version; 3) for infringements caused by: i) third party modifications of Contributor Version or ii) the combination of Modifications made by that Contributor with other software (except as part of the Contributor Version) or other devices; or 4) under Patent Claims infringed by Covered Code in the absence of Modifications made by that Contributor.

3. Distribution Obligations.

3.1. Application of License. The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients' rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code. Any Modification which You create or to which You contribute must be made available in Source Code form under the terms of this License either on the same media as an Executable version or via an accepted Electronic Distribution Mechanism to anyone to whom you made an Executable version available; and if made available via Electronic Distribution Mechanism, must remain available for at least twelve (12) months after the date it initially became available, or at least six (6) months after a subsequent version of that particular Modification has been made available to such recipients. You are responsible for ensuring that the Source Code version remains available even if the Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications. You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters

(a) Third Party Claims. If Contributor has knowledge that a license under a third party's intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled "LEGAL" which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying

appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) **Contributor APIs.** If Contributor's Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the *LEGAL* file.

3.5. **Required Notices.** You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients' rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. **Distribution of Executable Versions.** You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients' rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. **Larger Works.** You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

4. **Inability to Comply Due to Statute or Regulation.** If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the *LEGAL* file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5. **Application of this License.** This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6. **Versions of the License.**

6.1. **New Versions.** Netscape Communications Corporation ("Netscape") may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. **Effect of New Versions.** Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may

also choose to use such Covered Code under the terms of any subsequent version of the License published by Netscape. No one other than Netscape has the right to modify the terms applicable to Covered Code created under this License.

6.3. Derivative Works. If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape", "MPL", "NPL" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the Mozilla Public License and Netscape Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7. DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF DEFECTS, MERCHANTABILITY, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declaratory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY. UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS. The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS. This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys' fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS. As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE. Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

"The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.mozilla.org/MPL/>

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is _____.

The Initial Developer of the Original Code is _____. Portions created by _____ are Copyright (C) _____. All Rights Reserved.

Contributor(s): _____.

Alternatively, the contents of this file may be used under the terms of the _____ license (the “[____] License”), in which case the provisions of [____] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [____] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [____] License.“

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

16.2.4 MIT License

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

16.2.5 X Consortium License (X11)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

16.2.6 zlib license

This software is provided ‘as-is’, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly
jloup@gzip.org

Mark Adler
madler@alumni.caltech.edu

16.2.7 OpenSSL license

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape's SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

16.2.8 Slirp license

Copyright (c) 1995,1996 Danny Gasparovski. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by Danny Gasparovski.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DANNY GASPAROVSKI OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

16.2.9 liblzf license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

16.2.10 libpng license

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.

3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

16.2.11 lwIP license

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

16.2.12 libxml license

Except where otherwise noted in the source code (e.g. the files hash.c, list.c and the trio files, which are covered by a similar licence but with different Copyright notices) all the files are:

Copyright (C) 1998-2003 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

16.2.13 libxslt licenses

Licence for libxslt except libexslt:

Copyright (C) 2001-2002 Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

Licence for libxslt:

Copyright (C) 2001-2002 Thomas Broyer, Charlie Bozeman and Daniel Veillard. All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of the authors shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from him.

16.2.14 gSOAP Public License Version 1.3a

The gSOAP public license is derived from the Mozilla Public License (MPL1.1). The sections that were deleted from the original MPL1.1 text are 1.0.1, 2.1.(c),(d), 2.2.(c),(d), 8.2.(b), 10, and 11. Section 3.8 was added. The modified sections are 2.1.(b), 2.2.(b), 3.2 (simplified), 3.5 (deleted the last sentence), and 3.6 (simplified).

1 DEFINITIONS

1.1. "Contributor" means each entity that creates or contributes to the creation of Modifications.

1.2. "Contributor Version" means the combination of the Original Code, prior Modifications used by a Contributor, and the Modifications made by that particular Contributor.

1.3. "Covered Code" means the Original Code, or Modifications or the combination of the Original Code, and Modifications, in each case including portions thereof.

1.4. "Electronic Distribution Mechanism" means a mechanism generally accepted in the software development community for the electronic transfer of data.

1.5. "Executable" means Covered Code in any form other than Source Code.

1.6. "Initial Developer" means the individual or entity identified as the Initial Developer in the Source Code notice required by Exhibit A.

1.7. "Larger Work" means a work which combines Covered Code or portions thereof with code not governed by the terms of this License.

1.8. "License" means this document.

1.8.1. "Licensable" means having the right to grant, to the maximum extent possible, whether at the time of the initial grant or subsequently acquired, any and all of the rights conveyed herein.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code, or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2 SOURCE CODE LICENSE.

2.1. The Initial Developer Grant.

The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Initial Developer, to make, have made, use and sell ("offer to sell and import") the Original Code, Modifications, or portions thereof, but solely to the extent that any such patent is reasonably necessary to enable You to utilize, alone or in combination with other software, the Original Code, Modifications, or any combination or portions thereof.

(c)

(d)

2.2. Contributor Grant.

Subject to third party intellectual property claims, each Contributor hereby grants You a world-wide, royalty-free, non-exclusive license

(a) under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work; and

(b) under patents now or hereafter owned or controlled by Contributor, to make, have made, use and sell (“offer to sell and import”) the Contributor Version (or portions thereof), but solely to the extent that any such patent is reasonably necessary to enable You to utilize, alone or in combination with other software, the Contributor Version (or portions thereof).

(c)

(d)

3 DISTRIBUTION OBLIGATIONS.

3.1. Application of License.

The Modifications which You create or to which You contribute are governed by the terms of this License, including without limitation Section 2.2. The Source Code version of Covered Code may be distributed only under the terms of this License or a future version of this License released under Section 6.1, and You must include a copy of this License with every copy of the Source Code You distribute. You may not offer or impose any terms on any Source Code version that alters or restricts the applicable version of this License or the recipients’ rights hereunder. However, You may include an additional document offering the additional rights described in Section 3.5.

3.2. Availability of Source Code.

Any Modification created by You will be provided to the Initial Developer in Source Code form and are subject to the terms of the License. 3.3. Description of Modifications.

You must cause all Covered Code to which You contribute to contain a file documenting the changes You made to create that Covered Code and the date of any change. You must include a prominent statement that the Modification is derived, directly or indirectly, from Original Code provided by the Initial Developer and including the name of the Initial Developer in (a) the Source Code, and (b) in any notice in an Executable version or related documentation in which You describe the origin or ownership of the Covered Code.

3.4. Intellectual Property Matters.

(a) Third Party Claims. If Contributor has knowledge that a license under a third party’s intellectual property rights is required to exercise the rights granted by such Contributor under Sections 2.1 or 2.2, Contributor must include a text file with the Source Code distribution titled “LEGAL” which describes the claim and the party making the claim in sufficient detail that a recipient will know whom to contact. If Contributor obtains such knowledge after the Modification is made available as described in Section 3.2, Contributor shall promptly modify the LEGAL file in all copies Contributor makes available thereafter and shall take other steps (such as notifying appropriate mailing lists or newsgroups) reasonably calculated to inform those who received the Covered Code that new knowledge has been obtained.

(b) Contributor APIs. If Contributor’s Modifications include an application programming interface and Contributor has knowledge of patent licenses which are reasonably necessary to implement that API, Contributor must also include this information in the LEGAL file.

(c) Representations. Contributor represents that, except as disclosed pursuant to Section 3.4(a) above, Contributor believes that Contributor’s Modifications are Contributor’s original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices. You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients’ rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor.

3.6. Distribution of Executable Versions. You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your

choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient's rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. If you distribute executable versions containing Covered Code as part of a product, you must reproduce the notice in Exhibit B in the documentation and/or other materials provided with the product.

3.7. Larger Works. You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

3.8. Restrictions. You may not remove any product identification, copyright, proprietary notices or labels from gSOAP.

4 INABILITY TO COMPLY DUE TO STATUTE OR REGULATION.

If it is impossible for You to comply with any of the terms of this License with respect to some or all of the Covered Code due to statute, judicial order, or regulation then You must: (a) comply with the terms of this License to the maximum extent possible; and (b) describe the limitations and the code they affect. Such description must be included in the *LEGAL* file described in Section 3.4 and must be included with all distributions of the Source Code. Except to the extent prohibited by statute or regulation, such description must be sufficiently detailed for a recipient of ordinary skill to be able to understand it.

5 APPLICATION OF THIS LICENSE.

This License applies to code to which the Initial Developer has attached the notice in Exhibit A and to related Covered Code.

6 VERSIONS OF THE LICENSE.

6.1. New Versions.

Grantor may publish revised and/or new versions of the License from time to time. Each version will be given a distinguishing version number.

6.2. Effect of New Versions.

Once Covered Code has been published under a particular version of the License, You may always continue to use it under the terms of that version. You may also choose to use such Covered Code under the terms of any subsequent version of the License.

6.3. Derivative Works.

If You create or use a modified version of this License (which you may only do in order to apply it to code which is not already Covered Code governed by this License), You must (a) rename Your license so that the phrase "gSOAP" or any confusingly similar phrase do not appear in your license (except to note that your license differs from this License) and (b) otherwise make it clear that Your version of the license contains terms which differ from the gSOAP Public License. (Filling in the name of the Initial Developer, Original Code or Contributor in the notice described in Exhibit A shall not of themselves be deemed to be modifications of this License.)

7 DISCLAIMER OF WARRANTY.

COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, OF FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS, AND ANY WARRANTY THAT MAY ARISE BY REASON OF TRADE USAGE, CUSTOM, OR COURSE OF DEALING. WITHOUT LIMITING THE FOREGOING, YOU ACKNOWLEDGE THAT THE SOFTWARE IS PROVIDED "AS IS" AND THAT THE AUTHORS DO NOT WARRANT THE SOFTWARE WILL RUN UNINTERRUPTED OR ERROR FREE. LIMITED LIABILITY THE ENTIRE RISK AS TO RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. UNDER NO CIRCUMSTANCES WILL THE AUTHORS BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY KIND OR NATURE WHATSOEVER,

WHETHER BASED ON CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, ARISING OUT OF OR IN ANY WAY RELATED TO THE SOFTWARE, EVEN IF THE AUTHORS HAVE BEEN ADVISED ON THE POSSIBILITY OF SUCH DAMAGE OR IF SUCH DAMAGE COULD HAVE BEEN REASONABLY FORESEEN, AND NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY EXCLUSIVE REMEDY PROVIDED. SUCH LIMITATION ON DAMAGES INCLUDES, BUT IS NOT LIMITED TO, DAMAGES FOR LOSS OF GOODWILL, LOST PROFITS, LOSS OF DATA OR SOFTWARE, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION OR IMPAIRMENT OF OTHER GOODS. IN NO EVENT WILL THE AUTHORS BE LIABLE FOR THE COSTS OF PROCUREMENT OF SUBSTITUTE SOFTWARE OR SERVICES. YOU ACKNOWLEDGE THAT THIS SOFTWARE IS NOT DESIGNED FOR USE IN ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS SUCH AS OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR CONTROL, OR LIFE-CRITICAL APPLICATIONS. THE AUTHORS EXPRESSLY DISCLAIM ANY LIABILITY RESULTING FROM USE OF THE SOFTWARE IN ANY SUCH ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS AND ACCEPTS NO LIABILITY IN RESPECT OF ANY ACTIONS OR CLAIMS BASED ON THE USE OF THE SOFTWARE IN ANY SUCH ON-LINE EQUIPMENT IN HAZARDOUS ENVIRONMENTS BY YOU. FOR PURPOSES OF THIS PARAGRAPH, THE TERM "LIFE-CRITICAL APPLICATION" MEANS AN APPLICATION IN WHICH THE FUNCTIONING OR MALFUNCTIONING OF THE SOFTWARE MAY RESULT DIRECTLY OR INDIRECTLY IN PHYSICAL INJURY OR LOSS OF HUMAN LIFE. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8 TERMINATION.

8.1.

This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2.

8.3.

If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9 LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY'S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10 U.S. GOVERNMENT END USERS.

11 MISCELLANEOUS.

12 RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

EXHIBIT A.

“The contents of this file are subject to the gSOAP Public License Version 1.3 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.cs.fsu.edu/~engelen/soaplicense.html>. Software distributed under the License is distributed on an “AS IS” basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code of the gSOAP Software is: stdsoap.h, stdsoap2.h, stdsoap.c, stdsoap2.c, stdsoap.cpp, stdsoap2.cpp, soapcpp2.h, soapcpp2.c, soapcpp2_lex.l, soapcpp2_yacc.y, error2.h, error2.c, symbol2.c, init2.c, soapdoc2.html, and soapdoc2.pdf, httpget.h, httpget.c, stl.h, stldeque.h, stllist.h, stlvector.h, stlset.h.

The Initial Developer of the Original Code is Robert A. van Engelen. Portions created by Robert A. van Engelen are Copyright (C) 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved.

Contributor(s): “_____”. [Note: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

EXHIBIT B.

“Part of the software embedded in this product is gSOAP software. Portions created by gSOAP are Copyright (C) 2001-2004 Robert A. van Engelen, Genivia inc. All Rights Reserved. THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY GENIVIA INC AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

16.2.15 Chromium licenses

16.2.15.1 Main license

Copyright (c) 2002, Stanford University All rights reserved.

Some portions of Chromium are copyrighted by individual organizations. Please see the files COPYRIGHT.LLNL and COPYRIGHT.REDHAT for more information.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Stanford University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

16.2.15.2 COPYRIGHT.LLNL file

This Chromium distribution contains information and code which is covered under the following notice:

Copyright (c) 2002, The Regents of the University of California. Produced at the Lawrence Livermore National Laboratory For details, contact: Randall Frank (rjfrank@llnl.gov). UCRL-CODE-2002-058 All rights reserved.

This file is part of Chromium. For details, see accompanying documentation.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer (as noted below) in the documentation and/or other materials provided with the distribution.

Neither the name of the UC/LLNL nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OF THE UNIVERSITY OF CALIFORNIA, THE U.S. DEPARTMENT OF ENERGY OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Additional BSD Notice

1. This notice is required to be provided under our contract with the U.S. Department of Energy (DOE). This work was produced at the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48 with the DOE.

2. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately-owned rights.

3. Also, reference herein to any specific commercial products, process, or services by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of

California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

16.2.15.3 COPYRIGHT.REDHAT file

This Chromium distribution contains information and code which is covered under the following notice:

Copyright 2001,2002 Red Hat Inc., Durham, North Carolina.
All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation on the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL RED HAT AND/OR THEIR SUPPLIERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

16.2.16 curl license

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1996 - 2009, Daniel Stenberg, daniel@haxx.se.
All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

16.2.17 libgd license

Portions copyright 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002 by Cold Spring Harbor Laboratory. Funded under Grant P41-RR02188 by the National Institutes of Health.

Portions copyright 1996, 1997, 1998, 1999, 2000, 2001, 2002 by Boutell.Com, Inc.

Portions relating to GD2 format copyright 1999, 2000, 2001, 2002 Philip Warner.

Portions relating to PNG copyright 1999, 2000, 2001, 2002 Greg Roelofs.

Portions relating to gdttf.c copyright 1999, 2000, 2001, 2002 John Ellson (ellson@lucent.com).

Portions relating to gdft.c copyright 2001, 2002 John Ellson (ellson@lucent.com).

Portions copyright 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007 Pierre-Alain Joye (pierre@libgd.org).

Portions relating to JPEG and to color quantization copyright 2000, 2001, 2002, Doug Becker and copyright (C) 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is based in part on the work of the Independent JPEG Group. See the file README-JPEG.TXT for more information.

Portions relating to WBMP copyright 2000, 2001, 2002 Maurice Szmurlo and Johan Van den Brande.

Permission has been granted to copy, distribute and modify gd in any context without fee, including a commercial application, provided that this notice is present in user-accessible supporting documentation.

This does not affect your ownership of the derived work itself, and the intent is to assure proper credit for the authors of gd, not to interfere with your productive use of gd. If you have questions, ask. "Derived works" includes all programs that utilize the library. Credit must be given in user-accessible documentation.

This software is provided "AS IS." The copyright holders disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability and fitness for a particular purpose, with respect to this code and accompanying documentation.

Although their code does not appear in gd, the authors wish to thank David Koblas, David Rowley, and Hutchison Avenue Software Corporation for their prior contributions.

16.2.18 BSD license from Intel

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Intel Corporation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

16.2.19 libjpeg License

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2010, Thomas G. Lane, Guido Vollbeding. All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltmain.sh). Another support script, install-sh, is copyright by X Consortium but is also freely distributable.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

16.2.20 x86 SIMD extension for IJG JPEG library license

Copyright 2009 Pierre Ossman <ossman@cendio.se> for Cendio AB

Copyright 2010 D. R. Commander

Based on

x86 SIMD extension for IJG JPEG library - version 1.02

Copyright (C) 1999-2006, MIYASAKA Masaru.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

16.2.21 FreeBSD license

The compilation of software known as FreeBSD is distributed under the following terms:

Copyright (c) 1992-2009 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

16.2.22 NetBSD license

Copyright (c) 1992, 1993 The Regents of the University of California. All rights reserved.

This software was developed by the Computer Systems Engineering group at Lawrence Berkeley Laboratory under DARPA contract BG 91-66 and contributed to Berkeley.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

17 Informations sur la confidentialité de VirtualBox

Politique de confidentialité version 5, 13 déc. 2012

La politique de confidentialité d'Oracle publiée sur <http://www.oracle.com/html/privacy.html> s'applique à vos données personnelles recueillies et utilisées par Oracle. Les informations suivantes décrivent en détails les informations qui sont échangées entre l'application VirtualBox et Oracle et celles recueillies par le site Internet virtualbox.org.

À§ 1 virtualbox.org. Le site Internet “virtualbox.org” enregistre les informations d'utilisation anonymes telles que votre adresse IP, votre situation géographique, votre navigateur, l'endroit d'où vous naviguez, la durée de votre visite, le nombre de pages vues pendant votre visite (collectivement, “données anonymes”). En outre, mais seulement si vous choisissez de vous enregistrer, le chasseur de bogues du site Internet et les forums stockent les données que vous choisissez de révéler lors de l'enregistrement telles que votre nom d'utilisateur et vos informations de contact.

À§ 2 Cookies. Le site Internet virtualbox.org, le chasseur de bogues et le forum utilisent des cookies pour identifier et savoir les navigateurs internet qui nous visitent et, si vous vous êtes enregistré, pour faciliter votre identification. La plupart des navigateurs vous permettent de refuser les cookies. Si vous pouvez visiter le site Internet sans cookies, l'écriture dans le chasseur de bogues et dans les services de forum ne fonctionnera probablement pas sans eux.

À§ 3 Processus d'enregistrement de VirtualBox. Il se peut que l'application VirtualBox demande à l'utilisateur, à titre facultatif, de s'enregistrer auprès d'Oracle. Si vous choisissez de vous enregistrer, votre nom, votre adresse électronique, votre pays et votre société seront envoyés à Oracle et stockées avec l'adresse IP de la personne ainsi que la version du produit et la plateforme utilisée.

À§ 4 Notifications de mise à jour. Il se peut que l'application VirtualBox contacte Oracle pour savoir si une nouvelle version de VirtualBox a été publiée et pour avertir l'utilisateur si tel est le cas. Dans ce processus, des données anonymes telles que votre adresse IP et un numéro de comptage non caractéristique, la version du produit et la plateforme utilisée sont envoyées pour que le serveur sache si une mise à jour est disponible. Par défaut, cette vérification a lieu une fois par jour. Pour modifier cet intervalle ou désactiver ces vérifications, allez dans les préférences de VirtualBox.

À§ 5 Utilisation des informations personnelles. Oracle peut utiliser ces données anonymes et personnelles recueillies par les moyens ci-dessus à des fins statistiques ou pour vous informer automatiquement des nouveaux messages liés au vôtre sur le chasseur de bogues et le forum, pour administrer le site Internet et pour vous contacter du fait de problèmes techniques. Oracle peut aussi vous informer de nouvelles versions de produits liés à VirtualBox.

Vos données personnelles ne seront en aucun cas transmises à un tiers sans votre consentement, sauf si cela est demandé à Oracle par une loi ou dans le cadre de procédures légales.

À§ 6 Mises à jour. Oracle peut mettre à jour sa politique de confidentialité n'importe quand en en publiant une nouvelle version sur <http://www.oracle.com/html/privacy.html> et les informations de confidentialité seront mises à jour dans la documentation fournie avec l'application VirtualBox. Vous devriez vérifier ces ressources de temps en temps pour vous assurer que les changements vous conviennent.

Glossary

A

ACPI Advanced Configuration and Power Interface, an industry specification for BIOS and hardware extensions to configure PC hardware and perform power management. Windows 2000 and higher as well as Linux 2.4 and higher support ACPI. Windows can only enable or disable ACPI support at installation time.

AHCI Advanced Host Controller Interface, the interface that supports SATA devices such as hard disks. See chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85.

AMD-V The hardware virtualization features built into modern AMD processors. See chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201.

API Application Programming Interface.

APIC Advanced Programmable Interrupt Controller, a newer version of the original PC PIC (programmable interrupt controller). Most modern CPUs contain an on-chip APIC (“local APIC”). Many systems also contain an I/O APIC (input output APIC) as a separate chip which provides more than 16 IRQs. Windows 2000 and higher use a different kernel if they detect an I/O APIC during installation. Therefore an I/O APIC must not be removed after installation.

ATA Advanced Technology Attachment, an industry standard for hard disk interfaces (synonymous with IDE). See chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85.

B

BIOS Basic Input/Output System, the firmware built into most personal computers which is responsible of initializing the hardware after the computer has been turned on and then booting an operating system. VirtualBox ships with its own virtual BIOS that runs when a virtual machine is started.

C

COM Microsoft Component Object Model, a programming infrastructure for modular software. COM allows applications to provide application programming interfaces which can be accessed from various other programming languages and applications. VirtualBox makes use of COM both internally and externally to provide a comprehensive API to 3rd party developers.

D

DHCP Dynamic Host Configuration Protocol. This allows a networking device in a network to acquire its IP address (and other networking details) automatically, in order to avoid having to configure all devices in a network with fixed IP addresses. VirtualBox has a built-in DHCP server that delivers an IP addresses to a virtual machine when networking is configured to NAT; see chapitre 6, *Le réseau virtuel*, page 98.

DKMS Dynamic Kernel Module Support. A framework that simplifies installing and updating external kernel modules on Linux machines; see chapitre 2.3.2, *Le module noyau de VirtualBox*, page 40.

E

EFI Extensible Firmware Interface, a firmware built into computers which is designed to replace the aging BIOS. Originally designed by Intel, most modern operating systems can now boot on computers which have EFI instead of a BIOS built into them; see chapitre 3.12, *Autre firmware (EFI)*, page 62.

EHCI Enhanced Host Controller Interface, the interface that implements the USB 2.0 standard.

G

GUI Graphical User Interface. Commonly used as an antonym to a “command line interface”, in the context of VirtualBox, we sometimes refer to the main graphical VirtualBox program as the “GUI”, to differentiate it from the VBoxManage interface.

GUID See UUID.

I

IDE Integrated Drive Electronics, an industry standard for hard disk interfaces. See chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85.

I/O APIC See APIC.

iSCSI Internet SCSI; see chapitre 5.10, *Serveurs iSCSI*, page 97.

M

MAC Media Access Control, a part of an Ethernet network card. A MAC address is a 6-byte number which identifies a network card. It is typically written in hexadecimal notation where the bytes are separated by colons, such as 00:17:3A:5E:CB:08.

MSI Message Signalled Interrupts, as supported by modern chipsets such as the ICH9; see chapitre 3.4.1, *Onglet “Carte mère”*, page 52. As opposed to traditional pin-based interrupts, with MSI, a small amount of data can accompany the actual interrupt message. This reduces the amount of hardware pins required, allows for more interrupts and better performance.

N

NAT Network Address Translation. A technique to share networking interfaces by which an interface modifies the source and/or target IP addresses of network packets according to specific rules. Commonly employed by routers and firewalls to shield an internal network from the Internet, VirtualBox can use NAT to easily share a host's physical networking hardware with its virtual machines. See chapitre 6.3, *Network Address Translation (NAT)*, page 100.

O

OVF Open Virtualization Format, a cross-platform industry standard to exchange virtual appliances between virtualization products; see chapitre 1.14, *Importer et exporter des machines virtuelles*, page 33.

P

PAE Physical Address Extension. This allows accessing more than 4 GB of RAM even in 32-bit environments; see chapitre 3.3.2, *Onglet “Avancé”*, page 51.

PIC See APIC.

PXE Preboot Execution Environment, an industry standard for booting PC systems from remote network locations. It includes DHCP for IP configuration and TFTP for file transfer. Using UNDI, a hardware independent driver stack for accessing the network card from bootstrap code is available.

R

RDP Remote Desktop Protocol, a protocol developed by Microsoft as an extension to the ITU T.128 and T.124 video conferencing protocol. With RDP, a PC system can be controlled from a remote location using a network connection over which data is transferred in both directions. Typically graphics updates and audio are sent from the remote machine and keyboard and mouse input events are sent from the client. A VirtualBox extension package by Oracle provides VRDP, an enhanced implementation of the relevant standards which is largely compatible with Microsoft's RDP implementation. See chapitre 7.1, *Affichage distant (VRDP support)*, page 109 for details.

S

SAS Serial Attached SCSI, an industry standard for hard disk interfaces. See chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85.

SATA Serial ATA, an industry standard for hard disk interfaces. See chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85.

SCSI Small Computer System Interface. An industry standard for data transfer between devices, especially for storage. See chapitre 5.1, *Les contrôleurs de disque dur : IDE, SATA (AHCI), SCSI, SAS*, page 85.

SMP Symmetrical Multiprocessing, meaning that the resources of a computer are shared between several processors. These can either be several processor chips or, as is more common with modern hardware, multiple CPU cores in one processor.

SSD Solid-state drive, uses microchips for storing data in a computer system. Compared to classical hard-disks they are having no mechanical components like spinning disks.

T

TAR A widely used file format for archiving. Originally, this stood for “Tape ARchive” and was already supported by very early Unix versions for backing up data on tape. The file format is still widely used today, for example, with OVF archives (with an .ova file extension); see chapitre 1.14, *Importer et exporter des machines virtuelles*, page 33.

U

UUID A Universally Unique Identifier – often also called GUID (Globally Unique Identifier) – is a string of numbers and letters which can be computed dynamically and is guaranteed to be unique. Generally, it is used as a global handle to identify entities. VirtualBox makes use of UUIDs to identify VMs, Virtual Disk Images (VDI files) and other entities.

V

VM Virtual Machine – a virtual computer that VirtualBox allows you to run on top of your actual hardware. See chapitre 1.2, *Un peu de terminologie*, page 12 for details.

VMM Virtual Machine Manager – the component of VirtualBox that controls VM execution. See chapitre 10.2, *Exécutables et composants de VirtualBox*, page 199 for a list of VirtualBox components.

VRDE VirtualBox Remote Desktop Extension. This interface is built into VirtualBox to allow VirtualBox extension packages to supply remote access to virtual machines. A VirtualBox extension package by Oracle provides VRDP support; see chapitre 7.1, *Affichage distant (VRDP support)*, page 109 for details.

VRDP See RDP.

VT-x The hardware virtualization features built into modern Intel processors. See chapitre 10.3, *Virtualisation matérielle vs. logicielle*, page 201.

X

XML The eXtensible Markup Language, a metastandard for all kinds of textual information. XML only specifies how data in the document is organized generally and does not prescribe how to semantically organize content.

XPCOM Mozilla Cross Platform Component Object Model, a programming infrastructure developed by the Mozilla browser project which is similar to Microsoft COM and allows applications to provide a modular programming interface. VirtualBox makes use of XPCOM on Linux both internally and externally to provide a comprehensive API to third-party developers.