

Référence du développeur Debian

Adam Di Carlo, responsable actuel <aph@debian.org>
Christian Schwarz <schwarz@debian.org>
Ian Jackson <ijackson@gnu.ai.mit.edu>

version française par Antoine Hulin <antoine.hulin@origan.fdn.org>
et les membres de la liste <debian-l10n-french@lists.debian.org>

Version 2.11, 08 avril 2002 (version française 20020315).

Copyright

Copyright ©1998 – 2001 Adam Di Carlo
Copyright ©1997, 1998 Christian Schwarz.

Ce manuel est un logiciel libre ; il peut être redistribué et/ou modifié selon les termes de la licence publique générale du projet GNU (GNU GPL), telle que publiée par la « Free Software Foundation » (version 2 ou toute version postérieure).

Il est distribué dans l'espoir qu'il sera utile, mais *sans aucune garantie*, sans même la garantie implicite d'une possible valeur marchande ou d'une adéquation à un besoin particulier. Consultez la licence publique générale du projet GNU pour plus de détails.

Une copie de la licence publique générale du projet GNU est disponible dans le fichier `/usr/share/common-licenses/GPL` de la distribution Debian GNU/Linux ou sur la toile : la licence publique générale du projet GNU (<http://www.gnu.org/copyleft/gpl.html>). Vous pouvez également l'obtenir en écrivant à la Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Table des matières

1	Introduction	1
1.1	Portée de ce document	1
1.2	Introduction à la version française	2
1.2.1	Avertissement	2
1.2.2	Glossaire	2
2	Devenir responsable Debian	5
2.1	Pour commencer	5
2.2	Devenir responsable Debian	6
2.3	Mentors Debian	8
3	Les charges du responsable Debian	9
3.1	Mise à jour de vos références Debian	9
3.2	Gérer votre clé publique	9
3.3	Prendre des vacances courtoisement	10
3.4	Coordination avec les développeurs amonts	10
3.5	Gestion des bogues bloquants pour la distribution	11
3.6	L'effort d'assurance qualité	11
3.7	Responsables injoignables	11
3.8	Démissionner courtoisement	12

4	Listes de diffusion, serveurs et autres machines	13
4.1	Les listes de diffusion	13
4.2	Les serveurs Debian	14
4.2.1	Le serveur maître	14
4.2.2	Le serveur ftp-master	15
4.2.3	Le serveur WWW	15
4.2.4	Le serveur CVS	15
4.2.5	Les miroirs des serveurs Debian	16
4.3	Les autres machines Debian	16
5	L'archive Debian	17
5.1	Aperçu	17
5.2	Les sections	19
5.3	Les architectures	20
5.4	Les sous-sections	20
5.5	Les paquets	21
5.6	Les répertoires des distributions	21
5.6.1	<i>Stable, testing et unstable</i>	21
5.6.2	<i>Experimental</i>	22
5.7	Les noms de distribution	23
6	La mise à jour d'un paquet	25
6.1	Annoncer un nouveau paquet	25
6.2	Vérifier le paquet avant la mise à jour	26
6.3	Générer le fichier « changes »	26
6.3.1	L'archive des sources amonts	27
6.3.2	Choisir une distribution	27
6.4	Mettre à jour un paquet	28
6.4.1	Installer un paquet sur ftp-master	28

6.4.2	Installer un paquet sur non-US (pandora)	29
6.4.3	Installer un paquet via chiark	30
6.4.4	Installer un paquet via erlangen	30
6.4.5	Les autres serveurs	30
6.5	Annoncer une mise à jour	31
6.6	Notification de l'installation d'un nouveau paquet	31
6.6.1	Le fichier <i>override</i>	31
7	Mise à jour indépendante	33
7.1	Terminologie	33
7.2	Qui peut faire une mise à jour indépendante ?	34
7.3	Quand faire une mise à jour indépendante source ?	34
7.4	Comment faire une mise à jour indépendante source ?	35
7.4.1	Numéro de version pour les mises à jour indépendantes sources	36
7.4.2	Les mises à jour indépendantes sources doivent être mentionnées dans le fichier changelog	36
7.4.3	Mise à jour indépendante source et système de suivi des bogues	37
7.4.4	Fabriquer une mise à jour indépendante source	37
8	Le portage	39
8.1	Être gentil avec les porteurs	39
8.2	Instructions pour les mises à jour des porteurs	40
8.2.1	Numéro de version des mises à jour indépendantes binaires	41
8.2.2	Quand faire une mise à jour indépendante source pour un portage ?	41
8.3	Outils pour les porteurs	42
8.3.1	quinn-diff	42
8.3.2	buildd	42
8.3.3	dpkg-cross	43

9 Déplacer, effacer, renommer, adopter et abandonner des paquets	45
9.1 Déplacer des paquets	45
9.2 Supprimer des paquets	46
9.2.1 Supprimer des paquets dans Incoming	46
9.3 Remplacer ou renommer un paquet	46
9.4 Abandonner un paquet	46
9.5 Adopter un paquet	47
10 Gérer les bogues	49
10.1 Superviser les rapports de bogues	49
10.2 Ouvrir un rapport de bogue	49
10.3 Répondre à des rapports de bogues	50
10.4 Quand les rapports sont fermés par une mise à jour	50
10.5 Les rapports Lintian	51
10.6 Ouvrir un grand nombre de rapports d'un seul coup	51
11 Interaction avec les nouveaux responsables	53
11.1 Parrainer un paquet	53
11.2 Aider un nouveau responsable	54
11.3 Suivre la candidature d'un nouveau responsable	54
12 Aperçu des outils du responsable Debian	55
12.1 dpkg-dev	55
12.2 lintian	55
12.3 debconf	56
12.4 debhelper	56
12.5 debmake	56
12.6 yada	56
12.7 equivs	57
12.8 cvs-buildpackage	57

12.9 dupload	57
12.10 fakeroot	57
12.11 devscripts	57
12.12 debget	58

Chapitre 1

Introduction

1.1 Portée de ce document

Le but de ce document est de donner une vue d'ensemble des procédures à suivre et des ressources mises à la disposition des développeurs Debian.

Les procédures décrites ci-après expliquent entre autre comment devenir responsable Debian ('Devenir responsable Debian' page 5), comment installer des nouveaux paquets dans l'archive ('La mise à jour d'un paquet' page 25), quand et comment faire un portage ou une mise à jour du paquet d'un autre responsable ('Mise à jour indépendante' page 33), comment déplacer, effacer ou abandonner un paquet ('Déplacer, effacer, renommer, adopter et abandonner des paquets' page 45) et comment gérer les bogues ('Gérer les bogues' page 49).

Les ressources présentées dans ce manuel incluent les listes de diffusion et les serveurs ('Listes de diffusion, serveurs et autres machines' page 13), une présentation de la structure de l'archive Debian ('L'archive Debian' page 17), des explications sur les serveurs qui acceptent les mises à jour de paquets ('Installer un paquet sur ftp-master' page 28) et une présentation des outils qui peuvent aider un responsable à améliorer la qualité de ses paquets ('Aperçu des outils du responsable Debian' page 55).

Ce manuel de référence ne présente pas les aspects techniques liés aux paquets Debian. Il ne décrit pas non plus comment les créer. Il ne décrit pas non plus les règles que doivent respecter les paquets Debian. Cette information est disponible dans le Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>).

De plus ce document *n'est pas l'expression d'une politique officielle*. Il contient de la documentation sur le système Debian et des conseils pratiques largement suivis. C'est une sorte de guide pratique.

1.2 Introduction à la version française

1.2.1 Avertissement

Bien que ce document soit en français, l'activité de responsable Debian implique une maîtrise de la langue anglaise. Le projet Debian est un projet international auquel participent des japonais, néo-zélandais, américains, allemands, finlandais, français, espagnols, danois...

En conséquence la langue utilisée dans toutes les listes de diffusion destinées aux développeurs est l'anglais et les rapports de bogue doivent être rédigés en anglais. En fait, sauf exception rare, tout dialogue avec les autres responsables Debian se fera en anglais quelque soit le média.

1.2.2 Glossaire

Cette section liste les termes techniques qui ont un sens spécifique dans le projet Debian. Pour chacune de ces expressions vous trouverez la traduction française utilisée dans ce manuel, une définition et une référence à la section du manuel qui traite de ce sujet.

- *Upload* : mise à jour, téléchargement (parfois livraison). Opération qui consiste à télécharger un nouveau paquet ou une nouvelle version de paquet dans l'archive Debian ('La mise à jour d'un paquet' page 25).
- *Non-maintainer upload (NMU)* : mise à jour indépendante. Téléchargement d'une nouvelle version de paquet dans l'archive Debian par une personne qui n'est pas officiellement responsable de ce paquet ('Mise à jour indépendante' page 33).
- *NMU source* : mise à jour indépendante source. Il s'agit d'une mise à jour indépendante pour un paquet source ('Terminologie' page 33).
- *NMU binaire* : mise à jour indépendante binaire. Mise à jour indépendante pour un paquet binaire ('Terminologie' page 33).
- *Bug Tracking System (BTS)* : système de suivi des bogues. Il s'agit du système utilisé par le projet Debian pour suivre les bogues et leur correction ('Gérer les bogues' page 49).
- *Bug submitter* : rapporteur du bogue. Personne qui envoie un rapport de bogue au système de suivi des bogues ('Ouvrir un rapport de bogue' page 49).
- *Release critical bug* : bogue remettant en cause la distribution. Bogues de gravité *critique*, *grave* et *sérieuse*. Ces bogues ne doivent pas apparaître dans une distribution *stable*. Ils doivent être corrigés ou bien les paquets en cause doivent être supprimés ('Gestion des bogues bloquants pour la distribution' page 11).
- *Unstable* : Nom de la distribution en cours de développement. Cette distribution contient les paquets envoyés par les développeurs. Ceux-ci étant humains, elle est parfois cassée ('*Stable, testing et unstable*' page 21).
- *Testing* : Nom de la distribution en test. Cette distribution reçoit les paquets des développeurs qui ont passé une période de deux semaines dans *unstable* et pour lesquels aucun bogue remettant en cause la distribution (cf. *Release critical bug*) n'a été découvert. Cette distribution n'a pas été testée en profondeur. Elle est cependant sensée être plus stable que *unstable* ('*Stable, testing et unstable*' page 21).

- *Stable* : Nom de la distribution dite stable. Cette distribution a été testée, validée et diffusée (*‘Stable, testing et unstable’* page 21).
- *Debian maintainer* : responsable Debian, développeur Debian (parfois mainteneur). Personne qui fait officiellement partie du projet Debian. Elle a accès aux serveurs Debian et participe au développement — au sens large — de la distribution (*‘Les charges du responsable Debian’* page 9). La plupart des responsables font de la mise en paquet mais il existe d’autres activités telles que documentation, site web, création des cédéroms, administration des serveurs. . .
- *Upstream version* : version amont. Il s’agit du logiciel tel qu’il est fourni par le responsable amont — par opposition à la version fournie par la distribution Debian. (Voir *‘Coordination avec les développeurs amonts’* page 10.)
- *Upstream maintainer* : responsable amont ou développeur amont. Personne(s) responsable(s) du développement ou de la maintenance d’un logiciel. En général le responsable amont n’a rien à voir avec le projet Debian (*‘Coordination avec les développeurs amonts’* page 10).
- *Debian keyring* : porte-clés Debian. Le porte-clés Debian contient les clés publiques de tous les responsables Debian (*‘Gérer votre clé publique’* page 9).
- *Work-needing and prospective packages (WNPP)* : paquets en souffrance et paquets souhaités. La liste des paquets en souffrance indique les paquets qui n’ont plus de responsable. La liste des paquets souhaités indique les logiciels que des utilisateurs aimeraient bien voir dans la distribution (*‘La mise à jour d’un paquet’* page 25).

Chapitre 2

Devenir responsable Debian

2.1 Pour commencer

Vous avez lu toute la documentation, vous comprenez l'intérêt de tout ce qui se trouve dans le paquet exemple `hello` et vous vous apprêtez à mettre en paquet votre logiciel préféré. Comment devenir responsable Debian et intégrer votre travail au projet ?

Si vous ne l'avez pas encore fait vous pouvez commencer par vous inscrire à la liste `<debian-devel@lists.debian.org>`.

Pour cela, envoyez un courrier à l'adresse `<debian-devel-REQUEST@lists.debian.org>` avec le mot `subscribe` dans la ligne *Objet*¹ de votre message. En cas de problème, contactez l'administrateur de la liste `<listmaster@lists.debian.org>`. Vous trouverez plus d'information dans la section 'Les listes de diffusion' page 13.

Vous suivrez les discussions de cette liste (sans poster) pendant quelques temps avant de coder quoi que ce soit et vous informerez la liste de votre intention de travailler sur quelque chose pour éviter de dupliquer le travail d'un autre.

Une autre liste intéressante est `<debian-mentors@lists.debian.org>`. Voir la section 'Mentors Debian' page 8 pour les détails. Le canal IRC `#debian` sur le réseau *Linux People IRC* (c.-à-d. `irc.debian.org`) pourra aussi être utile.

Quand vous avez choisi la manière dont vous contribuerez au projet Debian GNU/Linux, prenez contact avec les responsables Debian qui travaillent sur des tâches similaires. Ainsi vous pouvez apprendre auprès de personnes expérimentées. Si, par exemple, vous voulez mettre en paquet des logiciels existants, trouvez-vous un parrain. Un parrain est une personne qui travaillera sur vos paquets avec vous et les téléchargera dans l'archive Debian une fois qu'il sera satisfait de votre mise en paquet. Pour trouver un parrain, envoyez une demande de parrainage à la liste `<debian-mentors@lists.debian.org>` en vous présentant et en

¹Subject en anglais

décrivant votre paquet (voir ‘Parrainer un paquet’ page 53 pour en savoir plus sur le sujet). Si vous préférez porter Debian sur une architecture ou un noyau alternatif, abonnez vous aux listes dédiées au portage et demandez-y comment démarrer. Finalement, si vous êtes intéressé par la documentation ou l’assurance qualité (QA) vous pouvez contacter les responsables qui travaillent déjà sur ces tâches et proposer des rustines et des améliorations.

2.2 Devenir responsable Debian

Avant de décider de devenir responsable Debian, il vous faudra lire toute la documentation disponible dans le coin du nouveau responsable (<http://www.debian.org/devel/join/newmaint>). Elle décrit toutes les étapes préparatoires qu’il vous faudra franchir avant de déposer votre candidature. Par exemple, avant d’être candidat, il vous faudra lire le contrat social Debian (http://www.debian.org/social_contract). Devenir responsable Debian implique que vous adhérez à ce contrat social et que vous vous engagez à le soutenir ; il est très important que les responsables soient en accord avec les principes fondamentaux qui animent le projet Debian GNU/Linux. Lire le Manifeste GNU (<http://www.gnu.org/gnu/manifesto.html>) est aussi une bonne idée.

Le processus d’enregistrement a pour but de vérifier votre identité, vos intentions et vos compétences. Le nombre de personnes travaillant pour Debian GNU/Linux a atteint 800 et notre système est utilisé dans plusieurs endroits très importants : nous devons rester attentifs pour éviter un acte malveillant. C’est pourquoi nous contrôlons les nouveaux responsables avant de leur donner un compte sur nos serveurs et les autoriser à ajouter des paquets dans l’archive.

Pour devenir responsable, il faudra montrer que vous pouvez faire du bon travail et être un bon contributeur. Pour cela, vous pourrez proposer des rustines par le système de suivi des bogues (BTS) ou maintenir un paquet parrainé pendant un temps. Nous attendons aussi des contributeurs qu’ils soient intéressés par le projet dans son ensemble et pas uniquement par leurs propres paquets. Si vous pouvez aider d’autres responsables en fournissant des informations sur un bogue ou même avec une rustine, faite-le !

Pour votre candidature, vous devrez être familiarisé avec la philosophie du projet Debian et avec sa documentation technique. Il vous faudra aussi une clé GPG signée par un responsable Debian. Si votre clé GPG n’est pas encore signée, vous devriez essayer de rencontrer un responsable Debian pour le faire. La page de coordination des signatures de clé GPG (<http://nm.debian.org/gpg.php>) devrait vous aider à trouver un responsable Debian près de chez vous. (Si vous ne trouvez pas de responsable près de chez vous, il existe un second moyen pour valider votre identité. Vous pouvez envoyer une photo d’identité signée avec votre clé GPG. Privilégiez tout de même la clé GPG signée pour valider une identité. Reportez-vous à la page d’identification (<http://www.debian.org/devel/join/nm-step2>) pour en savoir plus sur ces deux options.)

Si vous n’avez pas de clé OpenPGP, créez-la. Tout responsable a besoin d’une clé OpenPGP pour signer et vérifier les mises à jour de paquets. Vous lirez la documentation du logiciel de cryptographie que vous utiliserez car elle contient des informations indispensables pour la sécurité de votre clé. Les défaillances de sécurité sont bien plus souvent dues à des erreurs humaines qu’à des problèmes logiciels ou à des techniques

d'espionnage avancées. Voir 'Gérer votre clé publique' page 9 pour plus d'information sur la gestion de votre clé publique.

Debian utilise GNU Privacy Guard (paquet `gnupg` version 1 ou supérieure comme standard de base). Vous pouvez aussi utiliser une autre implémentation d'OpenPGP. OpenPGP est un standard ouvert basé sur la RFC 2440 (<http://www.gnupg.org/rfc2440.html>).

L'algorithme à clé publique recommandé pour les développements Debian est DSA (parfois appelé « DSS » ou « DH\EIGamal »). Vous pouvez aussi utiliser d'autres types de clés. La longueur de votre clé doit être au minimum de 1024 bits ; il n'y a pas de raison d'utiliser une clé plus courte et le faire serait beaucoup moins sûr. Votre clé doit être signée avec votre propre identifiant ; cela évite les falsifications. GNU Privacy Guard le fait automatiquement.

Si votre clé publique n'est pas sur un serveur public tel que `pgp5.ai.mit.edu`, reportez-vous à la documentation disponible localement dans `/usr/share/doc/pgp/keyserv.doc`. Cette documentation explique comment mettre votre clé publique sur un serveur. L'équipe *New maintainer* mettra votre clé publique sur les serveurs de clés si elle n'y est pas déjà.

Pour cause de restriction sur le droit à l'exportation aux États-Unis, certains paquets, dont `gnupg`, sont hébergés sur des serveurs FTP en dehors des États-Unis. Vous trouverez les adresses de ces serveurs dans le fichier <ftp://ftp.debian.org/debian/README.non-US>.

Certains pays limitent l'usage des logiciels de cryptographie. Cela ne devrait cependant pas avoir d'impact sur l'activité d'un responsable de paquet car il peut être tout à fait légal d'utiliser des logiciels de cryptographie pour l'authentification plutôt que pour le cryptage (comme c'est le cas en France). Debian GNU/Linux ne nécessite en aucune manière de cryptographie en tant que cryptage. Si vous vivez dans un pays où l'usage de la cryptographie pour authentification est interdit, contactez-nous pour que nous prenions des dispositions particulières.

Pour faire acte de candidature, il vous faut un responsable Debian qui vérifiera votre candidature (un *avocat*). Après avoir contribué au projet Debian pendant un temps, quand vous choisissez de devenir un responsable Debian officiel, un responsable déjà enregistré avec qui vous aurez travaillé dans les derniers mois devra exprimer que, d'après lui, vous pouvez contribuer avec succès au projet Debian.

Quand vous avez trouvé un *avocat*, quand votre clé GPG est signée et quand vous avez contribué pendant un temps au projet vous êtes prêts pour faire acte de candidature. Il vous suffit pour cela de vous enregistrer sur notre page de candidature (<http://nm.debian.org/newnm.php>). Ensuite, votre avocat devra confirmer votre candidature. Quand il aura accompli cette tâche, un responsable de candidature² sera désigné pour vous accompagner dans le processus d'enregistrement. Vous pouvez toujours consulter le tableau de bord des candidatures (<http://nm.debian.org/>) pour connaître l'état de votre candidature.

Pour en savoir plus, consultez le coin des nouveaux responsables (<http://www.debian.org/devel/join/newmaint>) sur le site Debian. Assurez-vous de bien connaître les étapes nécessaires au processus d'enregistrement avant vous porter candidat. Vous gagnerez beaucoup de temps si vous êtes bien préparé.

²Application manager

2.3 Mentors Debian

La liste de diffusion <debian-mentors@lists.debian.org> a été créée pour les responsables débutants qui cherchent de l'aide pour leur première création de paquet ainsi que pour des problèmes spécifiques aux développeurs. Tout responsable débutant est invité à s'y inscrire (voir 'Les listes de diffusion' page 13 pour plus de détails).

Ceux qui préfèrent une aide personnalisée (via une correspondance privée) doivent aussi poster dans cette liste, un développeur expérimenté se portera volontaire pour les aider.

Chapitre 3

Les charges du responsable Debian

3.1 Mise à jour de vos références Debian

Une base de données LDAP contient de nombreuses informations concernant les développeurs, vous pouvez y accéder à l'adresse <https://db.debian.org/>. Vous pouvez modifier votre mot de passe (ce mot de passe est diffusé sur la plupart des machines auxquelles vous avez accès), votre adresse, votre pays, la latitude et la longitude de l'endroit où vous résidez, vos numéros de téléphone et de fax, votre interpréteur de commande préféré, votre surnom IRC, votre page web, votre adresse de courrier électronique ainsi que l'alias que vous utilisez pour votre courrier électronique chez debian.org. La plupart de ces informations ne sont pas accessibles au public. Pour plus de détails au sujet de cette base de données reportez-vous à sa documentation en ligne disponible à l'adresse <http://db.debian.org/doc-general.html>.

Vous y tiendrez à jour les informations vous concernant.

3.2 Gérer votre clé publique

Soyez très vigilant en utilisant votre clé privée. Ne la placez pas sur un serveur public ou sur des machines multi-utilisateurs telles que `master.debian.org`. Sauvegardez vos clés et gardez-en une copie hors connexion. Lisez la documentation fournie avec votre logiciel et la FAQ PGP (<http://www.cam.ac.uk.pgp.net/pgpnet/pgp-faq/>).

Si vous ajoutez des signatures ou des identifiants à votre clé publique, vous pouvez mettre à jour le porte-clés Debian en envoyant votre clé publique à `keyring.debian.org`. Si vous voulez ajouter ou supprimer une clé, envoyez un courrier à `<keyring-maint@debian.org>`. Les procédures d'extraction de clé décrites dans 'Devenir responsable Debian' page 6 s'appliquent ici.

Vous pouvez trouver une présentation approfondie de la gestion de clé Debian dans la documentation du paquet `debian-keyring`.

3.3 Prendre des vacances courtoisement

La plupart des développeurs prennent des vacances ; généralement, cela signifie qu'ils ne peuvent ni travailler pour Debian ni être joints par courrier électronique si un problème se présentait. Les autres développeurs ont besoin de savoir que vous êtes en vacances ; ainsi ils sauront comment réagir en cas de problème. En général, cela signifie que les autres développeurs sont autorisés à modifier votre paquet (voir 'Mise à jour indépendante' page 33) en cas de gros problème (bogues bloquants pour la distribution, mise à jour de sécurité...) durant vos vacances.

Il y a deux choses à faire pour informer les autres développeurs. Premièrement, envoyer un courrier électronique indiquant vos dates de vacances à <debian-private@lists.debian.org>, vous pouvez aussi donner quelques instructions pour indiquer comment agir si un problème survenait. Notez bien que certaines personnes ne sont pas intéressées par les annonces de vacances et ne veulent pas les lire ; ajoutez '[VAC]' à l'objet de votre courrier pour qu'il soit facilement filtré.

Deuxièmement, il faut mettre à jour la base de données Debian LDAP et vous signaler « en vacances » (l'accès à cette information est réservé aux développeurs). N'oubliez pas de retirer votre indicateur « en vacances » lorsque celles-ci sont terminées !

3.4 Coordination avec les développeurs amonts

Une grosse part de votre travail de responsable Debian sera de rester en contact avec les développeurs amonts. Parfois, les utilisateurs établissent des rapports de bogue qui ne sont pas spécifiques à Debian. Vous devez transmettre ces rapports de bogue aux développeurs amonts pour qu'ils soient corrigés dans les prochaines versions. Il n'est pas de votre responsabilité de corriger les bogues qui ne sont pas spécifiques à Debian. Toutefois, si vous êtes capable de le faire, nous vous encourageons à contribuer au développement amont en proposant une rustine qui corrige ce bogue. Les utilisateurs et responsables Debian proposent souvent des rustines pour corriger des bogues amonts, il vous faudra alors évaluer cette rustine puis la transmettre aux développeurs amonts.

Si vous avez besoin de modifier les sources d'un logiciel pour fabriquer un paquet conforme à la charte Debian, alors vous devriez proposer une rustine aux développeurs amonts pour qu'elle soit incluse dans leur version. Ainsi, vous n'aurez plus besoin de modifier les sources lors des mises à jour amonts suivantes. Quels que soient les changements dont vous avez besoin, il faut toujours essayer de rester dans la lignée des sources amonts.

3.5 Gestion des bogues bloquants pour la distribution

Les bogues bloquants pour la distribution¹ sont les bogues de gravité *critique*, *grave* et *sérieuse*². Ces bogues peuvent retarder la diffusion d'une distribution Debian et/ou peuvent justifier la suppression d'un paquet d'une distribution gelée. C'est pourquoi ces bogues ont besoin d'être corrigés au plus vite. Vous devez être conscient que certains développeurs faisant partie de l'équipe d'assurance qualité Debian (<http://qa.debian.org/>) surveillent ces bogues et essaient de vous aider chaque fois qu'ils le peuvent. Si vous ne pouvez pas corriger un bogue de ce type dans les deux semaines, vous devriez soit demander de l'aide en envoyant un courrier à l'équipe d'assurance qualité (*QA group* <debian-qa@lists.debian.org>), soit expliquer vos difficultés et présenter un plan pour corriger le problème en répondant au rapport de bogue concerné. Si rien n'est fait, des personnes de l'équipe d'assurance qualité pourraient chercher à corriger votre paquet (voir 'Mise à jour indépendante' page 33) après avoir tenté de vous contacter. Ils pourraient attendre moins longtemps que d'habitude pour faire leur mise à jour s'il n'y a pas trace d'activité récente de votre part dans le système de suivi des bogues.

3.6 L'effort d'assurance qualité

Même s'il y a un groupe de personnes dédié à l'assurance qualité, cette activité ne leur est pas réservée. Vous pouvez participer à cet effort en éliminant, autant que faire se peut, tout bogue de vos paquets et en observant les remarques émises par *lintian* (voir 'Les rapports Lintian' page 51). Si cela ne vous semble pas possible, il faudrait songer à abandonner certains de vos paquets (voir 'Abandonner un paquet' page 46). Vous pouvez aussi demander l'aide d'autres personnes pour rattraper votre retard dans la liste des bogues (vous pouvez demander de l'aide sur les listes <debian-qa@lists.debian.org> et <debian-devel@lists.debian.org>).

3.7 Responsables injoignables

Si vous remarqué qu'un paquet manque d'attention, assurez-vous que le responsable est toujours actif et continuera à travailler sur son paquet. Essayez de le contacter.

Si vous n'obtenez pas de réponse après quelques semaines, rassemblez toutes les informations utiles sur ce responsable. Commencez par consulter la base de données des développeurs (<http://db.debian.org>) pour déterminer si le responsable est en vacances et quand il a été vu pour la dernière fois. Listez les paquets importants gérés par ce responsable et les bogues bloquants pour la distribution qui concernent ces paquets.

Envoyez toutes ces informations à <debian-qa@lists.debian.org> pour que l'équipe d'assurance qualité prenne les mesures nécessaires.

¹Traduction de l'anglais *Release critical bug (RCB)*

²Respectivement *critical*, *grave* et *serious* en anglais

3.8 Démissionner courtoisement

Si vous choisissez de quitter le projet Debian, procédez comme suit :

1. Abandonnez tous vos paquets comme décrit dans 'Abandonner un paquet' page 46.
2. Envoyez un courrier électronique à <debian-private@lists.debian.org> indiquant comment vous quittez le projet.
3. Signalez aux responsables du porte-clés Debian que vous quittez le projet en écrivant à <keyring-maint@debian.org>.

Chapitre 4

Listes de diffusion, serveurs et autres machines

Dans ce chapitre vous trouverez une très brève description des listes de diffusion Debian, des principaux serveurs Debian et des autres machines auxquelles vous aurez accès en tant que développeur.

4.1 Les listes de diffusion

Le serveur de liste de diffusion est `lists.debian.org`. Envoyez un courrier électronique à `debian-foo-REQUEST@lists.debian.org` où `debian-foo` est le nom de la liste, avec le mot `subscribe` dans le champ *Objet*¹ pour vous abonner à la liste ou `unsubscribe` pour vous désabonner. Vous trouverez des instructions plus détaillées pour vous abonner et vous désabonner aux adresses <http://www.debian.org/MailingLists/subscribe>, <ftp://ftp.debian.org/debian/doc/mailling-lists.txt> ou localement dans le fichier `/usr/share/doc/debian/mailling-lists.txt` si vous avez installé le paquet `doc-debian`.

Lorsque vous répondez aux messages d'une liste de diffusion, n'envoyez pas de copie (CC) à l'auteur du courrier à moins qu'il ne l'ait demandé explicitement. Quelqu'un qui envoie un message à une liste de diffusion se doit d'y lire les réponses.

Les principales listes de diffusion Debian sont :

- `<debian-devel@lists.debian.org>` ;
- `<debian-policy@lists.debian.org>` ;
- `<debian-user@lists.debian.org>` ;
- `<debian-private@lists.debian.org>` ;
- `<debian-announce@lists.debian.org>` et
- `<debian-devel-announce@lists.debian.org>`.

¹Subject en anglais

Un développeur devrait, au minimum, être inscrit à la liste <debian-devel-announce@lists.debian.org>.

Il existe d'autres listes de diffusion spécialisées dans différents thèmes. Reportez-vous à la page <http://www.debian.org/MailingLists/subscribe> pour en obtenir la liste complète. Nous déconseillons l'usage des correspondances croisées (envoyer le même message à plusieurs listes).

<debian-private@lists.debian.org> est une liste de diffusion destinée aux discussions privées entre développeurs Debian. Elle doit être utilisée pour tout message qui ne doit pas être publié, quel qu'en soit la raison. C'est une liste à faible trafic et les utilisateurs sont priés de ne l'utiliser qu'en cas de réelle nécessité. De plus, il ne faut *jamaï*s faire suivre un courrier de cette liste à qui que ce soit.

<debian-email@lists.debian.org> est une liste de diffusion fourre-tout. Elle est utilisée pour les correspondances relatives à Debian qu'il serait utile d'archiver, telles que des échanges avec les auteurs amonts à propos de licences, de bogues ou encore des discussions sur le projet avec d'autres personnes.

Comme toujours sur internet, éliminez les lignes inutiles quand vous citez le message auquel vous répondez. Plus généralement, respectez les conventions habituelles lorsque vous envoyez des messages.

Les archives des listes de diffusion sont disponibles en ligne à l'adresse <http://lists.debian.org/>.

4.2 Les serveurs Debian

Les serveurs Debian sont bien connus et hébergent les fonctions critiques du projet Debian. Tout développeur se doit de savoir ce qu'ils sont et à quoi ils servent.

Si vous avez un problème en utilisant un serveur Debian et si vous estimez que les administrateurs système devraient en être avertis, vous trouverez l'adresse d'un responsable pour cette machine à la page <http://db.debian.org/machines.cgi>. Si votre problème n'est pas lié au système (paquet à supprimer ou suggestion pour le site web par exemple), il vous faudra en général ouvrir un rapport de bogue sur un « pseudo-paquet ». Reportez-vous à la section 'Ouvrir un rapport de bogue' page 49 pour connaître la procédure à suivre.

4.2.1 Le serveur maître

master.debian.org est le serveur maître du système de suivi des bogues (BTS²). Si vous envisagez de manipuler les rapports de bogue ou d'en faire une analyse statistique ce sera le bon endroit pour le faire. Informez la liste <debian-devel@lists.debian.org> de votre intention avant d'implémenter quoi que ce soit afin d'éviter un travail en double ou le gaspillage de temps machine.

Tous les développeurs Debian ont un compte sur master.debian.org. Prenez grand soin de votre mot de passe. Évitez les méthodes de connexion et de téléchargement qui envoient votre mot de passe en clair sur internet.

² Système de suivi des bogues se dit *Bug Tracking System* en anglais

Si vous rencontrez un problème avec `master.debian.org` tel qu'un disque plein, une activité suspecte ou autre, envoyez un courrier électronique à `<debian-admin@debian.org>`.

4.2.2 Le serveur ftp-master

Le serveur ftp-master (`ftp-master.debian.org` ou `auric.debian.org`) est le serveur maître de l'archive Debian (exception faite des paquets non-US). En général, les mises à jour de paquets se font sur ce serveur. Reportez-vous à la section 'La mise à jour d'un paquet' page 25 pour en savoir plus.

Les problèmes avec l'archive Debian FTP doivent généralement être rapportés comme bogue contre le pseudo-paquet `ftp.debian.org` ou par courrier électronique à `<ftpmaster@debian.org>` ; reportez-vous à la section 'Déplacer, effacer, renommer, adopter et abandonner des paquets' page 45 pour connaître la procédure à suivre.

4.2.3 Le serveur WWW

Le serveur web principal, `www.debian.org`, est aussi connu sous le nom `klecker.debian.org`. Tous les développeurs ont un compte sur cette machine.

Si vous désirez publier quelques informations spécifiques à Debian sur la toile, vous pouvez le faire en les déposant dans le sous-répertoire `public_html` de votre répertoire personnel sur `klecker.debian.org`. Toute information déposée à cet endroit est accessible à l'adresse `http://people.debian.org/~user-id/`. Vous devriez préférer ce serveur aux autres car les répertoires `public_html` du serveur `klecker.debian.org` sont sauvegardés. Ce n'est pas le cas sur les autres serveurs. N'utilisez pas ces serveurs pour publier des informations sans rapport avec le projet Debian avant d'en avoir obtenu la permission. Pour toute question supplémentaire, adressez-vous à la liste `<debian-devel@lists.debian.org>`.

Si vous rencontrez un problème avec un serveur web Debian, vous devez généralement envoyer un rapport de bogue contre le pseudo-paquet `www.debian.org`. Vérifiez d'abord sur le système de suivi des bogues (<http://bugs.debian.org/www.debian.org>) que personne ne l'a déjà rapporté avant vous.

4.2.4 Le serveur CVS

Le serveur `cvs.debian.org` est aussi connu sous le nom `klecker.debian.org` (déjà présenté plus haut). Si vous avez besoin d'un serveur CVS accessible par tous pour, par exemple, coordonner le travail de plusieurs développeurs sur un paquet, vous pouvez demander un espace sur ce serveur.

Le serveur `cvs.debian.org` autorise les accès CVS locaux, les accès en lecture seule pour les connexions client-serveur anonymes et les accès client-serveur complets pour les connexions ssh. L'espace CVS peut aussi être consulté par la toile à l'adresse <http://cvs.debian.org/>.

Pour obtenir un espace CVS, envoyez une demande à l'adresse `<debian-admin@debian.org>` en précisant le nom de l'espace, le propriétaire du répertoire racine et pourquoi vous en avez besoin.

4.2.5 Les miroirs des serveurs Debian

Les serveurs web et FTP Debian sont dupliqués sur d'autres serveurs. Évitez de charger les serveurs originaux. Idéalement, les serveurs originaux se contentent de copier leur contenu sur les miroirs de premier niveau et tous les utilisateurs consultent les miroirs. Ainsi le projet Debian étale ces besoins en bande passante sur plusieurs serveurs et plusieurs réseaux. Notez que la duplication des données sur les miroirs est déclenchée par le serveur maître. Ceci nous garantit que les miroirs sont aussi à jour que possible.

La liste des miroirs FTP (et en général HTTP) se trouve à l'adresse <http://www.debian.org/distrib/ftplist>. Vous trouverez plus d'information sur les miroirs Debian à l'adresse <http://www.debian.org/mirror/>. Cette page contient des informations et des outils qui pourront vous aider si vous avez l'intention de mettre en place votre propre miroir, que ce soit pour un usage interne ou pour un accès public.

Les miroirs sont en général mis en oeuvre par des tiers qui veulent aider Debian. C'est pourquoi les développeurs n'ont en général pas de compte sur ces machines.

4.3 Les autres machines Debian

Vous aurez peut-être accès à d'autres machines Debian. Vous pouvez utiliser ces machines pour des tâches en rapport avec le projet Debian. Soyez gentils avec les administrateurs et ne consommez pas des montagnes d'espace disque, de bande passante ou de temps machine sans en avoir d'abord reçu l'autorisation. En général, ces machines sont gérés par des volontaires et sont utilisées pour des activités de portage.

Vous trouverez une liste des autres machines mises à la disposition des développeurs Debian à l'adresse <http://db.debian.org/machines.cgi>.

Chapitre 5

L'archive Debian

5.1 Aperçu

La distribution Debian GNU/Linux est composée d'un grand nombre de paquets Debian (fichiers `.deb` : à peu près 6800) et de quelques autres fichiers (documentation, images des disquettes d'installation, etc.).

Voici un exemple d'arborescence pour une archive Debian complète :

```
dists/stable/main/  
dists/stable/main/binary-all/  
dists/stable/main/binary-all/admin/  
dists/stable/main/binary-all/base/  
dists/stable/main/binary-all/comm/  
dists/stable/main/binary-all/devel/  
...  
dists/stable/main/binary-i386/  
dists/stable/main/binary-i386/admin/  
dists/stable/main/binary-i386/base/  
...  
dists/stable/main/binary-m68k/  
dists/stable/main/binary-m68k/admin/  
dists/stable/main/binary-m68k/base/  
...  
dists/stable/main/source/  
dists/stable/main/source/admin/  
dists/stable/main/source/base/  
...  
dists/stable/main/disks-i386/
```

```
dists/stable/main/disks-m68k/
...

dists/stable/contrib/
dists/stable/contrib/binary-all/
dists/stable/contrib/binary-i386/
dists/stable/contrib/binary-m68k/
...
dists/stable/contrib/source/

dists/stable/non-free/
dists/stable/non-free/binary-all/
dists/stable/non-free/binary-i386/
dists/stable/non-free/binary-m68k/
...
dists/stable/non-free/source/

dists/testing/
dists/testing/main/
...
dists/testing/contrib/
...
dists/testing/non-free/
...

dists/unstable
dists/unstable/main/
...
dists/unstable/contrib/
...
dists/unstable/non-free/
...

pool/
pool/a/
pool/a/apt/
...
pool/b/
pool/b/bash/
...
pool/liba/
pool/liba/libalias-perl/
```

```
...  
pool/m/  
pool/m/mailx/  
...
```

Comme vous pouvez le voir, le répertoire racine contient deux répertoires, `dists/` et `pool/`. Le second est un ensemble de répertoires où sont stockés les paquets. Ceux-ci sont manipulés grâce à la base de données de l'archive et aux logiciels qui l'accompagnent. Le premier répertoire contient les distributions *stable*, *testing* et *unstable*. Le découpage en sous-répertoires est identique d'un répertoire de distribution à l'autre ; nous nous contenterons donc d'exposer ce découpage pour la distribution *stable*. Les fichiers `Packages` et `Sources` qui se trouvent dans les répertoires de distribution peuvent faire référence à des fichiers du répertoire `pool/`.

Le répertoire `dists/stable` contient trois répertoires nommés `main`, `contrib` et `non-free`.

Dans chacune de ces sections se trouve un répertoire contenant les paquets sources (`source/`), un répertoire pour chaque architecture supportée (`binary-i386/`, `binary-m68k/`, etc) et un répertoire pour les paquets indépendants de l'architecture (`binary-all/`).

La section *main* contient d'autres répertoires destinés aux images de disquettes et à quelques documentations essentielles pour installer la distribution Debian sur une architecture particulière (`disk-i386/`, `disk-m68k/`, etc).

Les répertoires `binary-*/` et `source/` sont à nouveau divisés en *sous-sections*.

5.2 Les sections

La section *main* constitue la **distribution Debian GNU/Linux officielle**. Elle est officielle parce qu'elle est conforme à toutes nos recommandations. Les deux autres sections divergent de ces recommandations à différents degrés, elles ne font donc pas officiellement partie de Debian GNU/Linux.

Chaque paquet de la section *main* doit être conforme aux Debian Free Software Guidelines (http://www.debian.org/social_contract#guidelines) et à toutes les autres recommandations décrites dans le Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>). Les DFSG¹ constituent notre définition de *logiciel libre*. Reportez-vous au *Debian Policy Manual* pour en savoir plus.

Les paquets de la section *contrib* doivent être conformes aux DFSG mais ne respectent pas d'autres contraintes. Ils peuvent par exemple dépendre de paquets de la section *non-free*.

Les paquets qui ne sont pas conformes aux DFSG sont rangés dans la section *non-free*. Bien que nous supportions l'usage de ces paquets et qu'ils bénéficient de nos infrastructures (système de suivi des bogues, listes de diffusion, etc), ces paquets *non-free* ne font pas partie de la distribution Debian.

¹*Debian Free Software Guidelines*

Le *Debian Policy Manual* donne des définitions plus précises pour ces trois sections. Les paragraphes précédents ne constituent qu'une introduction.

La séparation de l'archive en trois sections est importante pour toute personne qui désire distribuer Debian, que ce soit par serveur FTP ou sur cédérom. Il suffit de distribuer les sections *main* et *contrib* pour éviter tout problème légal. Certains paquets de la section *non-free* interdisent leur distribution à titre commercial par exemple.

D'un autre côté, un distributeur de cédérom pourra facilement vérifier individuellement les licences des paquets de la section *non-free* et ajouter autant de paquets qu'il est autorisé à le faire sur ses cédéroms (dans la mesure où cela varie énormément d'un distributeur à l'autre ce travail ne peut être fait par les développeurs Debian).

5.3 Les architectures

À ses débuts, le noyau Linux existait uniquement pour les architectures Intel x86 ; il en était de même pour Debian. Linux devenant de plus en plus populaire, il a été porté vers d'autres architectures.

Le noyau 2.0 existe pour les architectures Intel x86, DEC Alpha, SPARC, Motorola, 680x0 (Atari, Amiga, et Macintosh), MIPS et PowerPC. Le noyau 2.2 supporte encore plus d'architectures dont ARM et UltraSPARC. Puisque Linux supporte ces architectures, Debian a décidé qu'il devait les supporter aussi. C'est pourquoi plusieurs portages sont en cours ; en fait il y a aussi des portages vers d'autres noyaux. À côté d'*i386* (notre nom pour Intel x86), nous avons, au moment où j'écris *m68k*, *alpha*, *powerpc*, *sparc*, *hard-i386*, et *arm*.

Debian GNU/Linux 1.3 est disponible uniquement pour *i386*. Debian 2.0 supporte les architectures *i386* et *m68k*. Debian 2.1 supporte les architectures *i386*, *m68k*, *alpha* et *sparc*. Debian 2.2 ajoute le support de l'architecture *powerpc*.

Vous trouverez des informations destinées aux développeurs à propos d'un portage particulier sur la page Portage Debian (<http://www.debian.org/ports/>).

5.4 Les sous-sections

Les sections *main*, *contrib* et *non-free* sont divisées en sous-sections pour simplifier le processus d'installation et la maintenance de l'archive Debian. Ces sous-sections ne sont pas définies formellement, à l'exception peut-être de la sous-section *base*. Elles existent pour simplifier l'organisation des paquets et la navigation dans la liste des paquets disponibles. Reportez-vous à la distribution courante pour connaître la liste des sous-sections disponibles.

Notez qu'avec l'introduction de la nouvelle organisation de l'archive (voir le répertoire racine *pool/*), les sous-sections matérialisées par des répertoires pourraient disparaître à l'avenir. Elles seront cependant conservées dans le champ *Section* de l'en-tête des paquets.

5.5 Les paquets

Il existe deux types de paquets Debian : les paquets sources et les paquets binaires.

Les paquets sources sont constitués de deux ou trois fichiers :

- un fichier `.dsc` et un fichier `.tar.gz` ou,
- un fichier `.dsc`, un fichier `.orig.tar.gz` et un fichier `.diff.gz`.

Si un paquet est développé spécifiquement pour le projet Debian et n'est pas distribué en dehors de Debian, il n'y a qu'un fichier `.tar.gz` qui contient les sources du programme. Si un paquet est distribué ailleurs, le fichier `.orig.tar.gz` contient ce que l'on appelle *code source amont*, c'est-à-dire le code source distribué par le *mainteneur amont* (il s'agit souvent de l'auteur du logiciel). Dans ce cas, le fichier `.diff.gz` contient les modifications faites par le responsable Debian.

Le fichier `.dsc` liste tous les fichiers sources avec leurs sommes de contrôle (`md5sums`) et quelques informations supplémentaires concernant le paquet (responsable, version, etc).

5.6 Les répertoires des distributions

L'organisation des répertoires présentée précédemment est elle-même englobée par les *répertoires des distributions*. Chaque distribution est incluse dans le répertoire `pool` à la racine de l'archive Debian.

En résumant, une archive Debian a un répertoire racine sur un serveur FTP. Par exemple, sur le site miroir ftp.us.debian.org, l'archive Debian se trouve dans `/debian` ce qui est un emplacement courant. Un autre emplacement courant est `/pub/debian`.

Une distribution est composée de paquets Debian sources et binaires et des fichiers `Sources` et `Packages` correspondants. Ces derniers contiennent la description de tous les paquets. Les premiers sont conservés dans le répertoire `pool/` tandis que les seconds sont conservés dans le répertoire `dists/` de l'archive (pour des questions de compatibilité descendante)

5.6.1 *Stable, testing et unstable*

Il y a toujours une distribution appelée *stable* (dans le répertoire `dists/stable`), une distribution appelée *testing* (dans le répertoire `dists/testing`) et une distribution appelée *unstable* (dans le répertoire `dists/unstable`). Ceci reflète le processus de développement du projet Debian.

Les développements se font sur la distribution *unstable*² (c'est pourquoi elle est aussi appelée *distribution de développement*). Chaque développeur Debian peut modifier ses paquets à tout moment dans cette distribution. Ainsi son contenu change tous les jours. Comme aucun effort particulier n'est fait pour s'assurer que tout fonctionne correctement dans cette distribution, elle est parfois *instable*.

²*unstable* signifie « instable »

Les paquets sont copiés de *unstable* vers *testing* s'ils satisfont certains critères. Pour entrer dans la distribution *testing* un paquet doit faire partie de l'archive depuis deux semaines et ne doit pas avoir de bogue bloquant pour la distribution (*RC bug*). Passé cette période, le paquet sera installé dans *testing* dès que les paquets dont il dépend y seront. Ce processus est automatique. Vous pouvez consulter quelques notes sur ce système ainsi que les `update_excuses` (qui indiquent quels paquets sont candidats, lesquels ne le sont pas et pourquoi) à l'adresse <http://ftp-master.debian.org/testing/>.

Après une période de développement, quand le responsable de distribution³ le juge opportun, la distribution *testing* est gelée, ce qui signifie que les conditions à remplir pour qu'un paquet passe de *unstable* à *testing* sont durcies. Les paquets trop bogués sont supprimés et les seules mises à jours autorisées concernent les corrections de bogues. Après quelques temps, selon l'avancement, la distribution entre dans une phase de « gel complet » où les seules modifications acceptées concernent la procédure d'installation. Cette phase s'appelle un « cycle de test » et cela peut durer jusqu'à deux semaines. Il peut y avoir plusieurs cycles de tests avant que le responsable de distribution ne la déclare prête pour la diffusion. À la fin du dernier cycle de test, la distribution *frozen* est renommée *stable*, remplaçant l'ancienne distribution *stable* qui est enlevée à cette occasion (elle peut être retrouvée à l'adresse `archive.debian.org`).

Ce cycle de développement est basé sur l'idée que la distribution *instable* devient *stable* après une période de test (*testing*). Une distribution contient inévitablement des bogues, même si elle est classée *stable*. C'est pourquoi les distributions stables sont mises à jour de temps en temps. Les corrections introduites sont testées avec une grande attention et ajoutées individuellement à l'archive pour diminuer les risques d'introduire de nouveaux bogues. Vous pouvez trouver des paquets proposés pour les mises à jour de *stable* dans le répertoire `proposed-updates`. De temps en temps, les paquets de ce répertoire qui ne présentent pas de problème sont installés dans la distribution *stable* et le numéro de révision de cette distribution est incrémenté (« 1.3 » devient « 1.3r1 », « 2.0r2 » devient « 2.0r3 » et ainsi de suite).

Notez que pendant la période de gel les développements continuent sur la distribution instable car cette distribution reste en place.

5.6.2 *Experimental*

La distribution *experimental* est une distribution particulière. Ce n'est pas une distribution à part entière comme le sont *stable* et *unstable*. Elle est prévue pour servir de plate-forme de développement pour les projets expérimentaux qui ont de grandes chances de détruire le système ou bien pour des logiciels qui sont vraiment trop instables pour être inclus dans la distribution *unstable* (mais pour qui une mise en paquet est justifiée). Les utilisateurs qui téléchargent et installent des paquets depuis *experimental* sont prévenus : on ne peut pas faire confiance à la distribution *experimental*.

Si un logiciel risque de causer des dégâts importants, il sera sûrement préférable de le mettre dans la distribution *experimental*. Un système de fichier compressé, par exemple, devrait probablement aller dans *experimental*.

³Release manager

Une nouvelle version amont qui ajoute de nouvelles fonctions tout en supprimant de nombreuses autres ne devra pas être téléchargée dans l'archive Debian, elle pourra cependant être téléchargée dans *experimental*. Une nouvelle version non finalisée d'un logiciel qui utilise une méthode de configuration complètement différente pourrait aller dans *experimental* au gré du responsable. Si vous travaillez sur un cas de mise à jour complexe ou incompatible vous pouvez aussi utiliser *experimental* comme plate-forme d'intégration et ainsi fournir un accès aux testeurs.

Quelques logiciels expérimentaux peuvent aller dans *unstable*, avec un avertissement dans la description mais ce n'est pas recommandé car les paquets de *unstable* se propagent dans *testing* et aboutissent dans *stable*.

Un nouveau logiciel qui ne risque pas d'endommager le système ira directement dans *unstable*.

Une alternative à *experimental* consiste à utiliser vos pages personnelles sur le serveur `people.debian.org` (`klecker.debian.org`).

5.7 Les noms de distribution

Chaque distribution Debian diffusée a un nom : Debian 1.1 s'appelle « buz » ; Debian 1.2, « rex » ; Debian 1.3 « bo » ; Debian 2.0, « hamm » ; Debian 2.1, « slink » ; Debian 2.2 « potato » et Debian 3.0 « woody ». Il y a aussi une pseudo-distribution nommée « sid », il s'agit de la distribution *unstable* ; comme les paquets vont de *unstable* à *testing* quand ils approchent de la stabilité, la distribution « sid » n'est jamais diffusée. En plus du contenu habituel d'une distribution Debian, « sid » contient des paquets pour des architectures qui ne sont pas encore officiellement supportées ou pour lesquelles la distribution n'a pas encore été diffusée. Ces architectures doivent rejoindre les architectures supportées à une date future.

Comme Debian est un projet de développement ouvert (i.e. tout le monde peut participer et suivre les développements) même les distributions *unstable* et *testing* sont disponibles sur les serveurs HTTP et FTP de Debian.

Si nous avons nommé le répertoire qui contient la prochaine version à diffuser « testing », il aurait fallu le renommer « stable » au moment de la diffusion, ce qui aurait forcé les miroirs FTP à re-télécharger la distribution complète (qui est plutôt volumineuse).

D'un autre côté, si une distribution s'appelle *Debian-x.y* dès le départ, des personnes pourraient s'imaginer que la distribution Debian *x.y* est disponible. (Cela s'est produit par le passé, un distributeur avait gravé des cédéroms Debian 1.0 en utilisant une version de développement pré-1.0. C'est pour cette raison que la première version officielle était la version 1.1 et non la 1.0.)

En conséquence, les noms de répertoire des distributions dans l'archive sont déterminés par leurs noms de code et non par leur statut (exemple : *slink*). Ces noms sont identiques pendant la période de développement et une fois la distribution diffusée ; des liens symboliques, qui peuvent être modifiés facilement, indiquent la distribution stable actuelle.

Tout ceci explique pourquoi les répertoires des distributions sont nommés à partir des noms de code des distributions alors que *stable*, *testing* et *unstable* sont des liens symboliques qui pointent vers les répertoires appropriés.

Chapitre 6

La mise à jour d'un paquet

6.1 Annoncer un nouveau paquet

Si vous voulez créer un nouveau paquet pour la distribution Debian vous devriez commencer par consulter la liste des paquets en souffrance et paquets souhaités (<http://www.debian.org/devel/wnpp/>). Vous pourrez ainsi vérifier que personne ne travaille déjà sur ce paquet et éviter un double travail. Consultez aussi cette page si vous voulez en savoir plus.

Supposons que personne ne travaille sur le paquet que vous visez, vous devez alors envoyer un rapport de bogue (voir 'Ouvrir un rapport de bogue' page 49) concernant le pseudo-paquet `wnpp`. Ce courrier devra décrire le paquet que vous projetez de créer, la licence de ce paquet et l'URL à laquelle le source peut être téléchargé. Cette liste n'est pas limitative.

Le sujet de votre rapport de bogue devra être `<ITP1 : NomDuPaquet — courte description>`, en remplaçant `NomDuPaquet` par le nom du paquet. La gravité du bogue sera `whishlist`. Si vous le jugez nécessaire, envoyez une copie à `<debian-devel@lists.debian.org>` en mettant cette adresse dans le champ `X-Debbugs-CC` : de l'en-tête du message. N'utilisez pas le champ `CC` : car de cette manière le sujet du message ne contiendrait pas le numéro du bogue.

Il faudra aussi ajouter une entrée `Closes` : `bug#nnnnn` dans le fichier `changelog` du nouveau paquet. Cette indication fermera automatiquement le rapport de bogue à l'installation du nouveau paquet sur les serveurs d'archivage (voir 'Quand les rapports sont fermés par une mise à jour' page 50).

Il y a plusieurs raisons qui nous poussent à demander aux responsables d'annoncer leurs intentions :

- Les responsables ont ainsi la possibilité de puiser dans l'expérience des autres responsables et cela leur permet de savoir si une autre personne travaille déjà dessus.
- D'autres personnes qui envisagent de travailler sur le même paquet apprendront ainsi qu'il existe déjà un volontaire, l'effort peut alors être partagé.

¹*Intent To Package* : intention de mise en paquet

- Cela permet aux autres responsables d'en savoir plus sur le nouveau paquet que ne le permettent la description d'une ligne et l'habituelle entrée « *Initial release* » que l'on trouve dans les fichiers *changelog* qui sont postés sur `debian-devel-changes`. C'est une information utile pour les gens qui utilisent la distribution *unstable* et qui sont nos premiers testeurs. Nous devons faciliter la tâche de ces gens. Avec ces annonces, les développeurs Debian et toutes les autres personnes intéressées peuvent se faire une meilleure idée des évolutions et nouveautés du projet.

6.2 Vérifier le paquet avant la mise à jour

Avant de mettre à jour votre paquet vous ferez quelques tests de base. Vous devrez au moins faire les tests suivants (il vous faut une ancienne version du paquet) :

- Installez le paquet et vérifiez que le logiciel fonctionne. Si le paquet existait déjà dans une version plus ancienne, faites une mise à jour.
- Exécutez `lintian` sur votre paquet. Vous pouvez exécuter `lintian` comme suit : `lintian -v package-version.changes`. Ce programme fera une vérification sur les paquets source et binaire. Si vous ne comprenez pas les messages générés par `lintian` essayez l'option `-i`. Cette option rendra `lintian` beaucoup plus bavard dans sa description du problème. En principe, un paquet pour lequel `lintian` génère des erreurs (elles commencent par E) *ne doit pas* être installé dans l'archive. Pour en savoir plus sur `lintian` reportez-vous à la section `lintian` 'lintian' page 55.
- Faites régresser le paquet vers sa version précédente si elle existe — cela permet de tester les scripts `postrm` et `prerm`.
- Désinstallez le paquet et réinstallez-le.

6.3 Générer le fichier « changes »

Chaque nouvelle version d'un paquet installé sur les archives FTP Debian doit être accompagnée d'un fichier `.changes`. Ce fichier explique à l'administrateur de l'archive Debian ce qu'il doit faire du paquet. Il est en général créé par `dpkg-genchanges` au cours du processus de fabrication du paquet.

Le fichier `changes` est un fichier de contrôle qui contient les champs suivants :

- Format
- Date
- Source
- Binary
- Architecture
- Version
- Distribution
- Urgency
- Maintainer
- Description

- Changes
- Files

Tous ces champs sont obligatoires pour une installation sur les serveurs Debian. Vous pouvez consulter la liste des champs de contrôle dans le Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>) pour connaître les valeurs que prennent ces champs. Vous pouvez fermer un rapport de bogue automatiquement avec le champ `Description` (voir 'Quand les rapports sont fermés par une mise à jour' page 50).

6.3.1 L'archive des sources amonts

La première fois qu'un paquet est installé dans l'archive pour une version amont donnée, le fichier `tar` de cette version amont doit être téléchargé et mentionné dans le fichier `.changes`. Par la suite, ce fichier `tar` sera utilisé pour générer les fichiers `diff` et `.dsc` et il ne sera pas nécessaire de le télécharger à nouveau.

Par défaut, `dpkg-genchanges` et `dpkg-buildpackage` incluront le fichier `tar` amont si et seulement si le numéro de révision du paquet source est 0 ou 1, ce qui indique une nouvelle version amont. Ce comportement peut être modifié en utilisant `-sa` pour l'inclure systématiquement ou `-sd` pour ne jamais l'inclure.

Si la mise à jour ne contient pas le fichier `tar` des sources originaux, `dpkg-source` doit, pour construire les fichiers `.dsc` et `diff` de la mise à jour, utiliser un fichier `tar` identique à l'octet près à celui sur le serveur. Si pour une raison ou pour une autre il y a une différence, la nouvelle version de ce fichier doit à nouveau être incluse dans la mise à jour (en utilisant l'option `-sa` par exemple).

6.3.2 Choisir une distribution

Le champ `Distribution`, qui provient de la première ligne du fichier `debian/changelog`, indique à quelle distribution le paquet est destiné.

Il y a trois valeurs possibles pour ce champ : *stable*, *unstable* et *experimental*. En temps normal, les paquets sont téléchargés dans *unstable*.

Vous devriez éviter de combiner *stable* avec d'autres cibles à cause des problèmes potentiels de dépendance de bibliothèque (pour votre paquet et pour les paquets fabriqués par le démon de compilation pour les autres architectures). Se reporter à 'Mettre à jour un paquet de la distribution *stable*' de la présente page pour savoir quand et comment faire une mise à jour de *stable*.

Notez bien que combiner *experimental* avec quelque distribution que ce soit n'a pas de sens.

Mettre à jour un paquet de la distribution *stable*

Livrer un paquet pour la distribution *stable* signifie que le paquet sera dirigé vers le répertoire `proposed-updates` des archives Debian pour y être testé avant d'être effectivement inclus dans *stable*.

Une livraison pour la distribution *stable* requière des soins supplémentaires. Un paquet de cette distribution ne devrait être mis à jour que dans les cas suivants :

- un problème de sécurité (un avis de sécurité Debian²),
- un problème fonctionnel vraiment critique,
- un paquet devenu ininstallable,
- un paquet indisponible pour une architecture.

Il est fortement déconseillé de changer quoi que ce soit si ce n'est pas important car même une modification triviale peut causer un bogue plus tard. Livrer une nouvelle version amont d'un logiciel pour corriger un problème de sécurité est désapprouvé ; dans la plupart des cas la bonne solution consiste à prendre la rustine correspondante de la nouvelle version amont et à l'appliquer à l'ancienne (faire un portage (*backport*) de la rustine.

Les paquets livrés pour *stable* doivent être compilés avec la distribution *stable* pour que leurs dépendances se limitent aux bibliothèques (et autres paquets) disponibles dans *stable* ; un paquet livré pour la distribution *stable* qui dépend d'une bibliothèque qui n'est disponible que dans *unstable* sera rejeté. Modifier les dépendances d'autres paquets (en manipulant le champ `Provides` ou les fichiers `shlibs`) et, peut-être, rendre ces paquets ininstallables, est fortement déconseillé.

L'équipe responsable de la distribution³ (joignable à l'adresse `<debian-release@lists.debian.org>`) évaluera régulièrement le contenu de *proposed-updates* et décidera si votre paquet peut être inclus dans la distribution *stable*. Soyez précis (et, si nécessaire, généreux) quand vous décrivez, dans le fichier `changelog`, vos changements pour une livraison vers *stable* sinon le paquet ne sera pas considéré.

6.4 Mettre à jour un paquet

6.4.1 Installer un paquet sur `ftp-master`

Pour installer un paquet vous avez besoin d'un compte sur `ftp-master`, ce que vous devez avoir en tant que développeur Debian. Si vous utilisez `scp` ou `rsync` pour télécharger vos paquets, placez-les dans `/org/ftp.debian.org/incoming/`. Si vous utilisez un FTP anonyme, placez-les dans `/pub/UploadQueue/`.

Note : ne téléchargez pas de paquet soumis aux lois de contrôle des exportations américaines sur le serveur `ftp-master`, ni sur les serveurs `chiark` et `erlangen`. Cet interdit couvre à peu près tous les logiciels de cryptographie ainsi que quelques logiciels liés aux logiciels de cryptographie comme les clients de courrier électronique qui utilisent le cryptage et l'authentification PGP. Ces logiciels doivent être téléchargés sur le serveur `non-us`. Reportez-vous à la section 'Installer un paquet sur non-US (pandora)' page suivante pour en savoir plus. Si vous ne savez pas si votre paquet est soumis aux lois de contrôle des exportations américaines posez la question sur la liste `<debian-devel@lists.debian.org>`.

²Debian security advisory.

³the Release team

Le paquet `dupload` pourra vous faciliter le travail lors du téléchargement. Ce programme, bien pratique, est configuré par défaut pour se connecter par FTP sur les serveurs `ftp-master`, `chiark` et `erlangen`. Il peut aussi être configuré pour utiliser `ssh` ou `rsync`. Se reporter à `dupload(1)` et `dupload(5)` pour en savoir plus.

Après avoir téléchargé votre paquet, vous pouvez vérifier ce qu'en fera le logiciel de maintenance de l'archive en exécutant `dinstall` sur votre fichier `.changes` :

```
dinstall -n foo.changes
```

6.4.2 Installer un paquet sur non-US (pandora)

Comme nous l'avons dit un peu plus haut, les programmes soumis au contrôle des exportations américain ne doivent pas être installés sur `ftp-master`. Utilisez `scp` ou `rsync` pour copier votre paquet sur non-us.debian.org. Placer les fichiers dans le répertoire `/org/non-us.debian.org/incoming/`. Par défaut, vous pouvez utiliser le même *login/mot de passe* que pour `ftp-master`. Si vous utilisez une connexion FTP anonyme pour le téléchargement, placez vos fichiers dans le répertoire `/pub/UploadQueue/`.

Le programme `dupload` est, lui aussi, capable de télécharger sur `non-us` ; consultez la documentation de ce programme pour en savoir plus.

Tout comme sur `ftp-master`, vous pouvez vérifier votre téléchargement avec :

```
dinstall -n foo.changes
```

Attention, les personnes résidant aux États-Unis et les citoyens américains sont soumis à des restrictions sur l'exportation des logiciels de cryptographie. Au moment où j'écris, les citoyens américains peuvent exporter quelques logiciels de cryptographie soumis à une obligation de déclaration auprès du département du commerce américain.

Le *Debian Policy Manual* n'empêche pas les résidents et citoyens américains de livrer des paquets sur `non-US` mais ils devront être vigilants en le faisant. Nous recommandons aux responsables concernés de prendre toutes les précautions nécessaires, y compris la consultation d'un juriste, pour s'assurer qu'ils n'enfreignent pas une loi américaine en livrant un paquet sur `non-US`.

Pour les paquets des sections *non-US/main* et *non-US/contrib* les responsables devraient au moins suivre la procédure décrite par le gouvernement américain (<http://www.bxa.doc.gov/Encryption/PubAvailEncSource.html>). Les responsables de paquets *non-US/non-free* devront en plus consulter les règles de déclaration d'exportation pour les logiciels commerciaux (<http://www.bxa.doc.gov/Encryption/>).

Cette section a pour seul but d'informer, elle n'a pas valeur de conseil juridique. Une fois encore, nous recommandons aux résidents et citoyens américains de consulter un juriste avant de livrer un paquet sur `non-US`.

6.4.3 Installer un paquet via `chiark`

Si votre connexion vers `ftp-master` est lente, vous avez plusieurs possibilités. L'une d'elles consiste à télécharger vos fichiers dans `Incoming` en passant par le serveur `chiark` en Europe. Pour les détails consultez <ftp://ftp.chiark.greenend.org.uk/pub/debian/private/project/README.how-to-upload>

Attention, ne téléchargez pas de paquet contenant un logiciel soumis aux lois de contrôle des exportations américaines sur `chiark`. Les paquets téléchargés sur ce serveur sont redirigés vers `ftp-master`, les indications de la section 'Installer un paquet sur `ftp-master`' page 28 sont applicables ici aussi.

Le programme `dupload` est capable de télécharger sur `chiark`; consultez la documentation de ce programme pour en savoir plus.

6.4.4 Installer un paquet via `erlangen`

Vous pouvez aussi accéder à un serveur situé en Allemagne : il vous suffit d'ouvrir une connexion anonyme sur :

```
ftp://ftp.uni-erlangen.de/pub/Linux/debian/UploadQueue/.
```

Le téléchargement fait sur ce serveur doit être aussi complet que s'il était fait dans le répertoire `Incoming` sur `ftp-master` : il doit comporter le fichier `.changes` et tous les fichiers mentionnés dans ce dernier. Le serveur vérifie que le fichier `.changes` est bien signé avec la clé PGP d'un développeur Debian pour éviter que des faux paquets n'atteignent `ftp-master`. Vérifiez bien que le champ `Maintainer` du fichier `.changes` contient *votre* adresse électronique. De même que sur `ftp-master`, cette adresse est ensuite utilisée pour toutes les réponses.

Il n'est pas nécessaire de déplacer votre fichier dans un autre répertoire après le téléchargement, contrairement au serveur `chiark`. Vous devriez ensuite recevoir un courrier du serveur expliquant ce qu'il a fait de votre paquet. Normalement, il devrait avoir été déplacé vers `ftp-master`; vous serez informé par le même biais si une erreur s'est produite au cours du processus.

Attention, ne téléchargez pas de paquet contenant un logiciel soumis aux lois de contrôle des exportations américaines sur `erlangen`. Les paquets téléchargés sur ce serveur sont redirigés vers `ftp-master`, les indications de la section 'Installer un paquet sur `ftp-master`' page 28 sont applicables ici aussi.

Le programme `dupload` est capable de télécharger sur `erlangen`; consultez la documentation de ce programme pour en savoir plus.

6.4.5 Les autres serveurs

Un autre serveur est disponible aux États-Unis ; c'est un bon point de repli quand il est difficile de joindre `ftp-master`. Livrez vos paquets à l'adresse <ftp://samosa.debian.org/pub/UploadQueue/> comme vous le feriez sur `erlangen`.

Il existe aussi un serveur au Japon : téléchargez vos paquet par FTP anonyme sur <ftp://master.debian.or.jp/pub/Incoming/upload/>.

6.5 Annoncer une mise à jour

Quand un paquet est mis à jour, une annonce doit être envoyée sur l'une des listes « debian-changes ». Ceci est maintenant géré automatiquement par le logiciel de gestion de l'archive quand il est exécuté (en principe une fois par jour). Vous devez juste utiliser une version récente de `dpkg-dev` ($\geq 1.4.1.2$). Le courrier généré par le logiciel de maintenance de l'archive contiendra le fichier `.changes` signé que vous avez livré avec votre paquet. Précédemment, cette charge revenait à `dupload` ; vérifiez que vous avez bien configuré `dupload` pour qu'il n'envoie pas ces annonces (cherchez `dinstall_runs` dans la documentation de `dupload`).

Si un paquet est mis à jour avec un champ `Distribution` : à *stable*, l'annonce est envoyée sur la liste : `<debian-changes@lists.debian.org>`.

S'il est mis à jour avec un champ `Distribution` : à *unstable* ou *experimental*, l'annonce est envoyée sur la liste `<debian-devel-changes@lists.debian.org>`.

Le programme `dupload` est suffisamment intelligent pour déterminer où devra aller l'annonce et pour envoyer le courrier sur la bonne liste. Voir 'dupload' page 57.

6.6 Notification de l'installation d'un nouveau paquet

Les administrateurs de l'archive Debian sont responsables de l'installation des mises à jour. La plupart des mises à jour sont gérées quotidiennement par le logiciel de gestion de l'archive `katie`. Les mises à jour de paquets sur la distribution *unstable*, en particulier, sont installés ainsi. Dans les autres cas et notamment dans le cas d'un nouveau paquet, celui-ci sera installé manuellement. Il peut s'écouler jusqu'à un mois entre le téléchargement d'un paquet vers un serveur et son installation effective. Soyez patients.

Dans tous les cas vous recevrez un accusé de réception par courrier électronique indiquant que votre paquet a été installé et quels rapports de bogues ont été clos. Lisez attentivement ce courrier et vérifiez que tous les rapports de bogues que vous vouliez clore sont bien dans cette liste.

L'accusé de réception indique aussi la section dans laquelle le paquet a été installé. S'il ne s'agit pas de votre choix, vous recevrez un second courrier qui vous informera de cette différence (voir plus loin).

6.6.1 Le fichier *override*

Les champs `Section` et `Priority` du fichier `debian/control` n'indiquent ni où le paquet sera installé dans l'archive Debian ni sa priorité. Afin de conserver la cohérence de l'archive, ce sont les administrateurs qui contrôlent ces champs. Les valeurs du fichier `debian/control` sont juste des indications.

Les administrateurs de l'archive indiquent les sections et priorités des paquets dans le fichier *override*. Si ce fichier *override* et le fichier `debian/control` de votre paquet diffèrent, vous en serez informé par courrier électronique quand votre paquet sera installé dans l'archive. Vous pourrez corriger votre fichier *debian/control* avant votre prochain téléchargement ou alors vous aurez envie de modifier le fichier *override*.

Pour modifier la section dans laquelle un paquet est archivé, vous devez d'abord vérifier que fichier `debian/control` est correct. Ensuite, envoyez un courrier à `<override-change@debian.org>` ou un rapport de bogue sur le pseudo paquet `ftp.debian.org` demandant la modification de la section ou de la priorité de votre paquet. Exposez bien les raisons qui vous amènent à demander ces changements.

Pour en savoir plus sur le fichier *override*, reportez-vous à `dpkg-scanpackages(8)`, `/usr/share/doc/debian/bug-log-mailserver.txt` et `/usr/share/doc/debian/bug-maint-info.txt`.

Chapitre 7

Mise à jour indépendante

Dans certaines circonstances il est nécessaire qu'une personne autre que le responsable d'un paquet fasse une mise à jour de ce paquet. Ce type de mise à jour est désigné en anglais par l'expression *non-maintainer upload (NMU)*. Dans le présent document, nous traduisons librement cette expression par « mise à jour indépendante ».

Ces mises à jour indépendantes font partie du travail normal des porteurs qui compilent les paquets pour d'autres architectures (voir 'Le portage' page 39). Nous faisons aussi des mises à jour indépendantes quand un développeur Debian corrige le paquet d'un autre développeur pour éliminer un trou de sécurité ou un bogue paralysant. Cela se produit plus particulièrement en période de gel de la distribution de développement ou quand le responsable officiel du paquet ne peut pas fournir une correction dans un délai raisonnable.

Ce chapitre contient des informations qui vous expliqueront quand et comment faire des mises à jour indépendantes. Une distinction fondamentale doit être faite entre les mises à jour indépendantes sources et les mises à jour indépendantes binaires. Elle est explicitée dans la section suivante.

7.1 Terminologie

Deux nouvelles expressions sont introduites dans cette section : « mise à jour indépendante source » et « mise à jour indépendante binaire ». Ces expressions ont une définition précise dans le monde Debian. Elles correspondent toutes deux au même type d'activité ; elles impliquent toutes deux qu'une personne fait une mise à jour d'un paquet alors qu'elle n'est pas officiellement responsable de ce paquet. C'est pourquoi nous qualifions ces mises à jours d'*indépendantes*¹.

Une mise à jour indépendante source est une livraison de paquet faite par une personne qui n'est pas le responsable officiel de ce paquet avec pour objectif de corriger un bogue dans le paquet. Une mise à jour

¹Contrairement à ce que pourrait laisser entendre cette traduction de *non-maintainer upload*, il n'est pas question d'agir sans prévenir le responsable au préalable (voir 'Comment faire une mise à jour indépendante source ?' page 35).

indépendante source implique toujours une modification des sources du paquet, même s'il ne s'agit que d'un changement dans le fichier `debian/changelog`. Ce changement peut tout aussi bien concerner la partie amont du source que la partie spécifique à Debian. Une mise à jour indépendante source peut aussi inclure des paquets spécifiques à une architecture, un fichier *diff* modifié ou, plus rarement, des nouveaux sources amonts.

Une mise à jour indépendante binaire est la recompilation et l'archivage d'un paquet pour une architecture donnée. Il s'agit souvent du résultat d'un effort de portage. Une mise à jour indépendante binaire est la livraison d'un paquet compilé (souvent pour une autre architecture) à condition que cette compilation n'ait pas nécessité de modifications des sources. Dans de nombreux cas, les porteurs sont obligés de modifier les sources pour les rendre compilables sur leur architecture cible ; il s'agira alors d'une mise à jour indépendante source et non d'une mise à jour indépendante binaire. Comme vous pouvez le remarquer, nous ne faisons pas de distinction entre les mises à jour indépendantes faites par des porteurs et les autres mises à jour indépendantes.

Les mises à jour indépendantes sources et binaires sont toutes deux couvertes par l'expression « mise à jour indépendante » (NMU²). Pourtant cela conduit souvent à des confusions car beaucoup associent « mise à jour indépendante » et « mise à jour indépendante source ». Il faut donc rester vigilant. Dans ce chapitre, si nous utilisons « mise à jour indépendante » seul, il s'agit des deux types de livraisons.

7.2 Qui peut faire une mise à jour indépendante ?

Seuls les responsables Debian officiels peuvent faire des mises à jour indépendantes. Un responsable officiel est une personne dont la clé est dans le porte-clés Debian. Toute personne est invitée à télécharger les paquets sources pour corriger des bogues ; au lieu de faire des mises à jour indépendantes, ils pourront soumettre les rustines qui le méritent au système de suivi des bogues. Les responsables apprécient presque toujours les rustines et les rapports de bogue soignés.

7.3 Quand faire une mise à jour indépendante source ?

Les recommandations pour déterminer quand faire une mise à jour indépendante source dépendent de la distribution visée (i.e. stable, instable ou expérimentale). Les porteurs, ayant une activité particulière, obéissent à des règles légèrement différentes (voir 'Quand faire une mise à jour indépendante source pour un portage ?' page 41).

Quand une faille de sécurité est détectée, un paquet corrigé doit être livré le plus tôt possible. Dans ce cas, un membre de l'équipe de sécurité Debian³ entrera en contact avec le responsable du paquet pour s'assurer qu'un paquet corrigé sera livré dans un délai raisonnable (moins de 48 heures). Si le mainteneur ne peut

²Non-maintainer upload

³Debian Security officer

fournir une mise à jour suffisamment vite ou s'il ne peut être joint à temps, l'équipe de sécurité pourra corriger le paquet (i.e. faire une mise à jour indépendante source).

Pendant le cycle de mise au point (*release cycle*, voir '*Stable, testing et unstable*' page 21), les livraisons qui corrigent les bogues de gravité *sérieuse* (i.e. *serious*) et supérieures sont encouragées et acceptées. Même pendant cette période, vous devrez tenter d'entrer en contact avec le responsable du paquet ; il pourrait bien être sur le point de livrer un paquet corrigé lui aussi. Comme pour n'importe quelle mise à jour indépendante source, les recommandations de la section 'Comment faire une mise à jour indépendante source ?' de la présente page doivent être respectées.

Les mises à jour indépendantes sont aussi acceptable pour la distribution instable mais seulement en dernier ressort ou avec une autorisation. Essayez d'abord ce qui suit et si cela ne fonctionne pas il est probablement correct de faire une mise à jour indépendante (*NMU*) :

- Vérifiez que le bogue est bien référencé dans le système de suivi des bogues. S'il n'y est pas, faites un rapport de bogue.
- Envoyez un courrier au responsable du paquet et proposez votre aide pour corriger le bogue. Donnez-lui quelques jours.
- Lancez-vous. Corrigez le bogue et envoyez votre rustine au système de suivi des bogues. Construisez le paquet et testez-le comme décrit dans la section 'Vérifier le paquet avant la mise à jour' page 26. Utilisez le paquet chez vous.
- Attendez une réponse pendant quelques semaines.
- Envoyez un courrier au responsable lui demandant son accord pour faire une mise à jour indépendante (*NMU*).
- Vérifiez une nouvelle fois que votre modification n'a pas d'effet de bord inattendu et qu'elle est aussi minimaliste et peu intrusive que possible.
- Attendez une réponse une semaine encore.
- Faites votre mise à jour indépendante comme décrit dans la section 'Comment faire une mise à jour indépendante source ?' de la présente page.

7.4 Comment faire une mise à jour indépendante source ?

Les règles qui suivent s'appliquent aux porteurs tant qu'ils jouent le double rôle de correcteur de bogue et de porteur. Si un porteur doit modifier le paquet source, cette mise à jour est automatiquement une mise à jour indépendante source et est soumise aux règles qui suivent. Si un porteur construit un paquet binaire recompilé, les règles sont différentes (voir 'Instructions pour les mises à jour des porteurs' page 40).

Tout d'abord il est capital que ces mises à jour indépendantes soient aussi peu intrusives que possible. Ne faites pas de ménage, ne modifiez pas le nom des modules ou des fichiers, ne déplacez pas les répertoires ; plus généralement ne corrigez pas ce qui n'est pas cassé. Faites une rustine aussi petite que possible. Si certaines choses froissent votre sens de l'esthétique, parlez-en au responsable du paquet, au responsable amont ou soumettez un rapport de bogue. Quoiqu'il en soit, les changements esthétiques *ne doivent pas* être faits lors d'une mise à jour indépendante.

7.4.1 Numéro de version pour les mises à jour indépendantes sources

Chaque fois que vous modifiez un paquet, le numéro de version de ce paquet doit changer, même pour la plus triviale des modifications. Notre système de gestion de paquets s'appuie sur ces numéros de version.

Si vous faites une mise à jour indépendante (*NMU*), vous devez ajouter un numéro de version mineur à la partie *debian-revision* du numéro de version (la partie qui suit le dernier trait d'union). Ce numéro supplémentaire débutera à « 1 ». Prenons pour exemple le paquet « foo » qui porte le numéro de version 1.1-3. Dans l'archive, le fichier de contrôle du paquet source serait `foo_1.1-3.dsc`. La version amont est « 1.1 » et la révision Debian est « 3 ». La mise à jour indépendante suivante ajouterait le numéro de version mineur « .1 » au numéro de révision Debian ; le nouveau fichier de contrôle du paquet source serait alors `foo_1.1-3.1.dsc`.

Le numéro de révision mineur est nécessaire pour éviter de prendre un numéro de version au responsable officiel du paquet, ce qui pourrait perturber son travail. Cela a aussi l'avantage de montrer clairement que le paquet n'a pas été livré par le responsable officiel.

S'il n'y a pas de partie *debian-revision* dans le numéro de version du paquet, il faut en créer une en démarrant à « 0.1 ». S'il est absolument nécessaire qu'une personne qui n'est pas responsable d'un paquet fasse une livraison basée sur une nouvelle version amont, cette personne doit choisir « 0.1 » comme numéro de révision Debian. Le mainteneur du paquet doit, lui, démarrer sa numérotation à « 1 ». Notez que pour faire cela vous devrez utiliser `dpkg-buildpackage` avec l'option `-sa` pour le forcer à utiliser le nouveau paquet source (par défaut il reconnaît les numéros de révisions « 0 » ou « 1 » — il n'est pas suffisamment intelligent pour reconnaître « 0.1 »).

Attention, les porteurs qui ne font que recompiler les paquets pour d'autres architectures n'ont pas besoin de renuméroter les paquets. Les porteurs ne doivent utiliser de nouveaux numéros de version que s'ils modifient les paquets sources qu'ils recompilent, c'est-à-dire s'ils font une mise à jour indépendante source et non une mise à jour indépendante binaire.

7.4.2 Les mises à jour indépendantes sources doivent être mentionnées dans le fichier changelog

Une personne qui fait une mise à jour indépendante source doit ajouter une entrée dans le fichier `changelog` qui indique les bogues corrigés et qui précise pourquoi cette mise à jour était nécessaire. Cette entrée comportera l'adresse de la personne ayant fait la mise à jour ainsi que la version livrée.

Par convention, dans le cas d'une mise à jour indépendante source (*NMU*), l'entrée du fichier `changelog` débute par la ligne

* Non-maintainer upload

7.4.3 Mise à jour indépendante source et système de suivi des bogues

Un développeur qui n'est pas responsable d'un paquet doit faire aussi peu de modifications que possible et doit toujours envoyer ses modifications au système de suivi des bogues au format diff unifié (`diff -u`).

En cas de simple recompilation du paquet, le processus diffère suivant que vous agissez en tant que porteur ou pas. Si vous faites une mise à jour indépendante qui ne nécessite rien de plus qu'une recompilation et n'agissez pas en temps que porteur (i.e. une nouvelle bibliothèque partagée est disponible, un bogue a été corrigé dans `debhelper`), vous devez tout de même ajouter une entrée dans le fichier *changelog* ; il y aura donc une modification. Si vous êtes porteur, vous êtes probablement en train de faire une mise à jour indépendante binaire. (Note : ce système ne tient pas compte des porteurs qui font des recompilations — tenez cela pour une faiblesse de notre système de gestion des paquets.)

Si la mise à jour indépendante source (*source NMU*) corrige des bogues, ceux-ci doivent être marqués *fixed* (corrigé) dans le système de suivi des bogues plutôt que *close*. Par convention, seul le responsable du paquet et la personne qui a ouvert le rapport de bogue peuvent clore ce rapport. Heureusement, le système d'archivage Debian reconnaît les mises à jours indépendantes et positionne correctement le statut des bogues à *fixed* si la personne qui fait la mise à jour a listé tous les bogues dans le fichier *changelog* en utilisant la syntaxe `Closes : bug#nnnnn` (voir 'Quand les rapports sont fermés par une mise à jour' page 50 pour en savoir plus sur la fermeture de bogue par le fichier *changelog*). Ce passage au statut *fixed* assure que chacun sait que le bogue est corrigé par une mise à jour indépendante tout en laissant le rapport de bogue ouvert jusqu'à ce que le responsable du paquet incorpore les modifications de cette mise à jour dans la version officielle du paquet.

Après avoir fait une mise à jour indépendante, il vous faudra aussi ouvrir un nouveau rapport de bogue qui inclura une rustine contenant toutes les modifications que vous avez faites. Le responsable officiel pourra choisir d'appliquer la rustine, il pourra aussi employer une autre méthode pour régler le problème. Certains bogues sont corrigés dans la version amont, ce qui est une bonne raison pour annuler les modifications d'une mise à jour indépendante. Si le responsable choisit de mettre à jour le paquet plutôt que d'utiliser les rustines de la mise à jour indépendante, il devra s'assurer que cette nouvelle version corrige effectivement chacun des bogues corrigés dans la mise à jour indépendante.

De plus, le responsable officiel devrait *toujours* conserver les entrées documentant une mise à jour indépendante dans le fichier *changelog*.

7.4.4 Fabriquer une mise à jour indépendante source

Les paquets faisant l'objet d'une mise à jour indépendante source sont construits comme les autres. Sélectionnez une distribution en utilisant les règles décrites dans la section 'Choisir une distribution' page 27 et construisez un fichier `.changes` classique avec tout ce qui l'accompagne conformément à la description 'Mettre à jour un paquet' page 28.

En fait, toutes les prescriptions de la section 'La mise à jour d'un paquet' page 25 sont applicables ici, y compris l'obligation d'annoncer la mise à jour sur la liste idoine.

Vérifiez que vous n'avez pas modifié la valeur du champ `maintainer` dans le fichier `debian/control`. Votre nom, mentionné dans l'entrée du fichier `debian/changelog` concernant la mise à jour, sera utilisé pour signer le fichier `.changes`.

Chapitre 8

Le portage

Debian supporte un nombre croissant d'architectures. Même si vous n'êtes pas un porteur et même si vous n'utilisez qu'une architecture, il est de votre responsabilité de développeur d'être attentif aux questions de portabilité. C'est pourquoi il est important que vous lisiez ce chapitre même si vous n'êtes pas un porteur.

Porter un paquet consiste à faire un paquet binaire pour une architecture différente de celle du paquet binaire fait par le responsable du paquet. C'est une activité remarquable et essentielle. En fait, les porteurs sont à l'origine de la plupart des compilations de paquet Debian. Pour un paquet binaire *i386*, par exemple, il faut compter une recompilation pour chaque autre architecture, soit un total de 12 recompilations.

8.1 Être gentil avec les porteurs

Les porteurs ont une tâche remarquable et difficile car ils doivent gérer un grand nombre de paquets. Idéalement, tout paquet source devrait compiler sans modification. Malheureusement, c'est rarement le cas. Cette section contient une liste d'erreurs commises régulièrement par les responsables Debian — problèmes courants qui bloquent souvent les porteurs et compliquent inutilement leur travail.

Ici, le mot d'ordre est de répondre rapidement aux rapports de bogues et remarques soulevées par les porteurs. Traitez-les courtoisement, comme s'ils étaient co-responsables de vos paquets (ce qu'ils sont d'une certaine manière). Merci pour votre indulgence envers des rapports de bogues succincts ou peu clairs ; faites de votre mieux pour éliminer le problème.

Les problèmes les plus couramment rencontrés par les porteurs sont causés par des erreurs de mise en paquet dans le paquet source. Voici une *checklist* de choses auxquelles vous devez être attentif :

1. Vérifiez que les champs `Build-Depends` et `Build-Depends-Indep` du fichier `debian/control` sont corrects. Le meilleur moyen pour vérifier cela est d'utiliser le paquet `debootstrap` pour créer un environnement *unstable chrooté*. Dans cet environnement *chrooté* il faudra installer le paquet `build-essential` et tous les paquets mentionnés dans les champs `Build-Depends` et

Build-Depends-Indep. Ensuite, vous essayerez de fabriquer votre paquet dans cet environnement.

Consultez le Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>) pour en savoir plus sur les dépendances de fabrication.

2. Ne choisissez pas d'autre valeur que *all* ou *any* pour le champ architecture sans avoir de bonnes raisons pour le faire. Trop souvent, les développeurs ne respectent pas les instructions du Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>). Choisir la valeur « i386 » est la plupart du temps incorrect.
3. Vérifiez que votre paquet source est bon. Faites `dpkg-source -x package.dsc` pour vous assurer que le paquet se désarchive correctement. En utilisant le résultat de ce test, construisez votre paquet binaire à l'aide de la commande `dpkg-buildpackage`.
4. Vérifiez que les fichiers `debian/files` et `debian/substvars` ne sont pas dans votre paquet source. Ils doivent être effacés par la cible *clean* de `debian/rules`.
5. Assurez-vous que vous ne vous appuyez pas sur des éléments de configuration ou des logiciels installés ou modifiés localement. Par exemple, vous ne devriez jamais appeler des programmes du répertoire `/usr/local/bin` ou de répertoires équivalents. Essayez de ne pas vous appuyer sur des logiciels configurés de manière spéciale. Essayez de construire votre paquet sur une autre machine, même s'il s'agit de la même architecture.
6. Ne vous appuyez pas sur une installation préexistante de votre paquet (un sous-cas de la remarque précédente).
7. Si possible, ne vous appuyez pas sur une particularité présente dans un compilateur précis ou dans certaine version d'un compilateur. Si vous ne pouvez pas faire autrement, assurez-vous que les dépendances de fabrication reflètent bien cette restriction. Dans ce cas, vous cherchez sûrement les problèmes car quelques architectures pourraient choisir un compilateur différent.
8. Vérifiez que votre `debian/rules` distingue les cibles *binary-arch* et *binary-indep* comme l'exige le *Debian packaging manual*. Vérifiez que ces cibles sont indépendantes l'une de l'autre, c'est à dire qu'il n'est pas nécessaire d'invoquer l'une de ces cibles avant d'invoquer l'autre. Pour vérifier cela, essayez d'exécuter `dpkg-buildpackage -b`.

8.2 Instructions pour les mises à jour des porteurs

Si le paquet se construit tel quel sur l'architecture que vous visez, vous avez de la chance et votre travail est facile. Cette section s'applique dans ce cas ; elle décrit comment construire et installer correctement votre mise à jour indépendante binaire dans l'archive Debian. Si vous devez modifier le paquet pour le rendre compilable sur votre architecture cible vous devez faire une mise à jour des sources, consultez la section 'Comment faire une mise à jour indépendante source ?' page 35.

Pour une mise à jour indépendante binaire, ne faites pas de changement dans les sources. Vous n'avez pas besoin de modifier les fichiers du paquet source (cela inclut `debian/changelog`).

La manière d'invoquer `dpkg-buildpackage` est la suivante : `dpkg-buildpackage -B -eadresse-porteur`. Bien sûr, remplacez *adresse-porteur* par votre adresse électronique. Cette commande construira les portions du paquet qui dépendent de l'architecture, en utilisant la cible *binary-arch* de `debian/rules`.

8.2.1 Numéro de version des mises à jour indépendantes binaires

Parfois il est nécessaire de recompiler des paquets quand d'autres paquets, tels que les bibliothèques, ont été mis à jour. Dans ce cas, vous devez changer le numéro de version pour que le système de comparaison des numéros de version fonctionne correctement. Ce type de mise à jour reste une mise à jour indépendante binaire — il n'est pas nécessaire de reconsidérer le statut des paquets binaires des autres architectures pour les marquer périmés ou à recompiler.

Ces recompilations nécessitent des numéros de version « magiques » pour que le système de maintenance de l'archive comprennent que, bien qu'il y ait une nouvelle version, il n'y a pas eu de modification des sources. Si vous ne faites pas cela correctement, les administrateurs de l'archive rejeteront votre mise à jour (car il n'y aura pas de code source associé).

Cette magie associée à une mise à jour par recompilation est déclenchée en utilisant un troisième nombre dans la partie `debian` du numéro de version. Si, par exemple, la dernière version du paquet que vous recompilez était « 2.9-3 », votre mise à jour portera le numéro « 2.9-3.0.1 ». Si cette version était « 3.4-2.1 » votre mise à jour portera le numéro « 3.4-2.1.1 ».

8.2.2 Quand faire une mise à jour indépendante source pour un portage ?

Les porteurs qui font des mises à jour indépendantes sources suivent généralement les instructions de la section 'Mise à jour indépendante' page 33 tout comme les non-porteurs. Les délais d'attente sont cependant plus courts car les porteurs doivent manipuler un grand nombre de paquets. À nouveau, la situation diffère selon la distribution visée.

Si vous êtes porteur et faites une mise à jour pour *unstable*, les instructions précédentes sont applicables à deux différences près. Tout d'abord, le temps d'attente raisonnable — délai entre le moment où vous envoyez un rapport au système de suivi des bogues et le moment où vous pouvez faire une mise à jour indépendante (*NMU*) — est de sept jours. Ce délai peut être raccourci si le problème est crucial et met l'effort de portage en difficulté : c'est à la discrétion de l'équipe de portage. (Souvenez-vous, il ne s'agit pas d'un règlement mais de recommandations communément acceptées).

Deuxième différence, les porteurs qui font des mises à jour indépendantes sources doivent choisir une gravité *sérieuse* (i.e. *serious*) ou supérieure quand ils envoient leur rapport au système de suivi des bogues. Ceci assure qu'un paquet source unique permet de produire un paquet binaire pour chaque architecture supportée au moment de la sortie de la distribution. Il est très important d'avoir un paquet source et un paquet binaire pour toutes les architectures pour être conforme à plusieurs licences.

Les porteurs doivent éviter d'implémenter des contournements pour des bogues de l'environnement de compilation, du noyau ou de la `libc`. Quelques fois ces contournements sont inévitables. Si vous devez faire

quelque chose de ce genre, marquez proprement vos modifications avec des `#ifdef` et documentez votre contournement pour que l'on sache le retirer une fois que le problème aura disparu.

Les porteurs peuvent aussi avoir une adresse où ils publient le résultat de leur travail pendant le délai d'attente. Ainsi d'autres personnes peuvent bénéficier du travail du porteur même pendant ce délai. Bien sûr ces adresses n'ont rien d'officiel alors soyez sur vos gardes si vous les utilisez.

8.3 Outils pour les porteurs

Il y a plusieurs outils pour le travail de portage. Cette section contient une courte description de ces outils ; reportez-vous à la documentation de leurs paquets ou aux références citées ci-après pour une description complète.

8.3.1 `quinn-diff`

`quinn-diff` est utilisé pour localiser les différences d'une architecture à l'autre. Il pourrait vous dire, par exemple, quels paquets de l'architecture *X* doivent être portés sur l'architecture *Y*.

8.3.2 `buildd`

Le système `buildd` est un système distribué pour la compilation d'une distribution. Il est habituellement utilisé en conjonction avec des automates de compilation ; ce sont des machines « esclaves » qui récupèrent des paquets sources et tentent de les compiler. Il est aussi possible d'interagir par courrier électronique avec ce système. Cette interface est utilisée par les porteurs pour récupérer un paquet source (en général un paquet qui ne peut être compilé automatiquement) et travailler dessus.

`buildd` n'est pas disponible sous forme de paquet ; pourtant, la plupart des équipes de porteurs l'utilisent aujourd'hui ou ont prévu de l'utiliser bientôt. Il regroupe un ensemble de composants très utiles, continuellement utilisés mais non encore mis en paquet, tels que `andrea`, `sbuild` et `wanna-build`.

Une partie des informations produites par `buildd` — utiles pour les porteurs — est disponible sur la toile à l'adresse <http://buildd.debian.org/>. Ces informations incluent les résultats produits toutes les nuits par `andrea` (dépendances des sources) et `quinn-diff` (paquets à recompiler).

Nous sommes très enthousiasmés par ce système car il a de nombreux usages potentiels. Des groupes de développeurs indépendants peuvent utiliser ce système pour créer différentes saveurs de la Debian — qui peuvent être ou ne pas être intéressantes pour tous (une version de Debian compilée avec des vérifications relatives à `gcc`). Ce système nous permettra aussi de recompiler rapidement toute une distribution.

8.3.3 `dpkg-cross`

`dpkg-cross` est un outil qui installe les bibliothèques et les entêtes nécessaires à une compilation croisée¹ d'une manière similaire à `dpkg`. De plus `dpkg-buildpackage` et `dpkg-shlibdeps` ont été améliorés pour supporter les compilations croisées.

¹*cross-compilation*

Chapitre 9

Déplacer, effacer, renommer, adopter et abandonner des paquets

Certaines manipulations de l'archive ne sont pas accessibles avec le processus de mise à jour automatisé. Elles sont appliquées manuellement par les développeurs. Ce chapitre décrit ce qu'il faut faire dans ces situations.

9.1 Déplacer des paquets

Parfois un paquet pourra changer de section. Une nouvelle version d'un paquet de la section `non-free` pourrait par exemple être distribuée sous licence LGPL GNU ; dans ce cas le paquet doit être déplacé dans la section `main` ou `contrib`¹.

Si vous avez besoin de modifier la section de l'un de vos paquets, modifiez les informations de contrôle du paquet pour le placer dans la section désirée et téléchargez à nouveau votre paquet dans l'archive. Reportez-vous au Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>) pour en savoir plus. Lisez attentivement le rapport d'installation qui vous sera envoyé lors de l'archivage de votre paquet. Si pour une raison ou une autre le paquet est toujours à son ancien emplacement, envoyez un rapport de bogue sur le pseudo-paquet `ftp.debian.org` demandant la suppression dudit paquet. Décrivez précisément vos manipulations car il pourrait s'agir d'un bogue dans le logiciel de gestion de l'archive.

Si vous avez besoin de modifier la sous-section de l'un de vos paquets (`devel` ou `admin` par exemple) la procédure est légèrement différente. Modifiez la sous-section dans le fichier de contrôle de votre paquet et téléchargez-le. Il vous faudra ensuite demander la modification du fichier *override* comme décrit dans la section 'Le fichier *override*' page 31.

¹Reportez-vous au Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>) pour savoir dans quelle section un paquet doit être classé.

9.2 Supprimer des paquets

Si pour une raison ou une autre vous avez besoin de supprimer un paquet de l'archive (disons qu'il s'agit d'une vieille bibliothèque devenue inutile, que l'on conservait pour des raisons de compatibilité), il vous faudra envoyer un rapport de bogue concernant le pseudo-paquet `ftp.debian.org` et demandant sa suppression. N'oubliez pas de préciser de quelle distribution le paquet doit être supprimé.

Si vous ne savez pas bien si un paquet peut être supprimé, demandez l'avis des autres développeurs sur la liste `<debian-devel@lists.debian.org>`. Le programme `apt-cache` du paquet `apt` pourra aussi vous être utile. La commande `apt-cache showpkg package` vous indiquera, entre autres, les paquets qui dépendent de `package`.

9.2.1 Supprimer des paquets dans Incoming

Si vous décidez de supprimer un paquet de la section Incoming, il serait gentil de l'annoncer sur la liste de diffusion appropriée (`<debian-changes@lists.debian.org>` ou `<debian-devel-changes@lists.debian.org>`). Ce n'est pas obligatoire.

9.3 Remplacer ou renommer un paquet

Il peut vous arriver de vous tromper en nommant un paquet et avoir besoin de le renommer. Dans ce cas, vous devrez intervenir en deux étapes. D'abord, modifiez votre fichier `debian/control` pour que votre paquet renommé remplace et entre en conflit avec le nom de paquet que vous voulez remplacer. Reportez-vous au Debian Policy Manual (<http://www.debian.org/doc/debian-policy/>) pour les détails. Une fois que votre paquet est installé dans l'archive, faites un rapport de bogue concernant le pseudo-paquet `ftp.debian.org` et demandant la suppression du paquet mal nommé. l'archive, faites un rapport de bogue concernant le pseudo-paquet `ftp.debian.org` et demandant la suppression du paquet mal nommé.

9.4 Abandonner un paquet

Si vous ne pouvez plus maintenir un paquet, vous devez en informer les autres et faire le nécessaire pour qu'il soit marqué *abandonné* (i.e. *orphaned*). Vous devriez aussi remplacer votre nom par Debian QA Group `<packages@qa.debian.org>` dans le champ `maintainer` du paquet et faire un rapport de bogue sur le pseudo-paquet `wnpp`. Le titre de votre rapport de bogue devra être « O^2 : `paquet --- courte description` » pour indiquer que le paquet est orphelin. La gravité du bogue sera *normale*. Si vous le jugez nécessaire, envoyez une copie à `<debian-devel@lists.debian.org>` en mettant

²*Orphaned* : abandonné.

cette adresse dans le champ X-Debbugs-CC : de l'en-tête du message. N'utilisez pas le champ CC : car de cette manière le sujet du message ne contiendra pas le numéro du bogue.

Si le paquet est particulièrement important pour la distribution il vous faudra faire un rapport de bogue sur le pseudo-paquet wnpp, titrer « RFA³ : *paquet --- courte description* » et lui affecter la gravité *importante*. Envoyez une copie de votre rapport de bogue à la liste `debian-devel` comme décrit précédemment.

Reportez-vous à la page WNPP⁴ <http://www.debian.org/devel/wnpp/> pour en savoir plus.

9.5 Adopter un paquet

Une liste des paquets en attente de nouveau responsable est disponible à la page Work-Needing and Prospective Packages (<http://www.debian.org/devel/wnpp/>). Si vous voulez prendre en charge un paquet de cette liste reportez-vous à la page susmentionnée pour connaître la procédure à suivre.

Prendre un paquet parce qu'il vous semble que celui-ci est négligé n'est pas correct — ce serait une prise d'otage. Vous pouvez prendre contact avec le responsable actuel et lui demander si vous pouvez prendre en charge ce paquet. Vous ne pouvez le faire sans son accord. Qu'il vous ignore n'est pas une raison suffisante pour le faire. Si vous avez le sentiment qu'un responsable est parti sans prévenir depuis un moment, demandez-le sur la liste `<debian-private@lists.debian.org>`.

Si vous reprenez un vieux paquet, vous voudrez sûrement que le système de suivi des bogues indique que vous êtes le responsable du paquet. Cela se produira automatiquement une fois que vous aurez installé une nouvelle version du paquet dans l'archive avec le champ `Maintainer` à jour. Cela peut prendre quelques heures après le téléchargement. Si vous pensez ne pas avoir de mise à jour à faire pour un moment, envoyez un courrier électronique à `<override-change@debian.org>` pour pouvoir recevoir les rapports de bogue.

³*Request For Adoption* : offre d'adoption.

⁴*Work-needing and prospective packages* : paquets en souffrance et paquets souhaités.

Chapitre 10

Gérer les bogues

10.1 Superviser les rapports de bogues

Si vous voulez être un bon responsable, consultez régulièrement la page du système de suivi des bogues (<http://www.debian.org/Bugs/>). Cette page contient tous les rapports de bogue qui concernent vos paquets.

Les responsables interagissent avec le système de suivi des bogues en utilisant l'adresse électronique `bugs.debian.org`. Vous trouverez une documentation sur les commandes disponibles à l'adresse <http://www.debian.org/Bugs/> ou, si vous avez installé le paquet `doc-debian`, dans les fichiers locaux `/usr/share/doc/debian/bug-*`.

Certains trouvent utile de recevoir régulièrement une synthèse des rapports de bogues ouverts. Si vous voulez recevoir une synthèse hebdomadaire relevant tous les rapports de bogue ouverts pour vos paquets, vous pouvez configurer `cron` comme suit :

```
# Synthèse hebdomadaire des rapports de bogue qui me concernent
0 17 * * fri    echo "index maint address" | mail request@bugs.debian.org
```

Remplacez *address* par votre adresse officielle de responsable Debian.

10.2 Ouvrir un rapport de bogue

En tant que responsable Debian, vous trouvez des bogues dans d'autres paquets. En tant que responsable Debian, vous trouvez des bogues dans d'autres paquets, ou bien vous voulez réassigner à d'autres paquets des rapports de bogue concernant vos paquets. La page d'aide du système de suivi des bogues (<http://www.debian.org/Bugs/server-control.html>) vous explique comment procéder.

Nous vous encourageons à ouvrir des rapports de bogue s'il y a des problèmes. Essayez de soumettre ces rapports depuis un compte utilisateur où vous pouvez recevoir du courrier. Ne les envoyez pas en tant que *root*.

Vérifiez que le bogue n'a pas encore été déclaré. Essayez de faire un travail propre en déclarant le bogue et en l'envoyant à la bonne adresse.

Pour faire encore mieux, vous pouvez consulter d'autres paquets, grouper les bogues qui ont été rapportés plusieurs fois ou affecter la gravité *corrigé* (i.e. *fixed*) à un rapport dont le bogue a été corrigé. Attention, quand vous n'êtes ni le rapporteur d'un bogue ni le responsable du paquet vous ne devez pas clore le rapport (à moins que vous n'ayez l'autorisation du responsable).

10.3 Répondre à des rapports de bogues

Assurez-vous que toutes vos discussions concernant les bogues sont envoyées au rapporteur du bogue et au bogue lui-même (<123@bugs.debian.org> par exemple).

Vous ne devez *jamais* fermer un rapport de bogue en envoyant la commande *close* à l'adresse :

```
<control@bugs.debian.org>
```

Si vous le faites, le rapporteur n'aura aucun retour sur la clôture de son rapport.

10.4 Quand les rapports sont fermés par une mise à jour

Si vous corrigez un bogue dans vos paquets, il est de votre responsabilité de fermer le rapport de bogue associé. Pourtant vous ne devez pas le fermer avant que le paquet n'ait été accepté dans l'archive Debian. C'est pourquoi, vous pouvez et vous devez clore le rapport dans le système de suivi des bogues une fois que vous avez reçu l'avis indiquant que votre nouveau paquet a été installé dans l'archive.

Si vous utilisez une version récente de `dpkg-dev` et que vous remplissez convenablement le fichier `changelog`, le logiciel de maintenance de l'archive fermera les rapports de bogue automatiquement. Tout ce que vous avez à faire est de respecter la syntaxe suivante dans votre fichier `debian/changelog` :

```
acme-cannon (3.1415) unstable; urgency=low

* Frobbed with options (closes: Bug#98339)
* Added safety to prevent operator dismemberment, closes: bug#98765,
  bug#98713, #98714.
* Added manpage. Closes: #98725.
```

Techniquement parlant, l'expression régulière suivante est utilisée :

```
/closes:\s*(?:bug)?#\s*\d+(?:,\s*(?:bug)?#\s*\d+)*\s*/ig
```

L'auteur préfère la syntaxe (`closes : Bug#XXX`), car elle est facilement repérable dans le fichier `changelog`.

Si vous voulez fermer un rapport de bogue manuellement — à l'ancienne — il suffit d'envoyer le fichier `.changes` à l'adresse `<XXX-done@bugs.debian.org>` où `XXX` est le numéro du bogue.

10.5 Les rapports Lintian

Vous devriez récupérer la dernière version de `lintian` depuis *unstable* régulièrement et vérifier tous vos paquets. Vous pouvez aussi chercher votre adresse électronique dans la page de rapport lintian (<http://lintian.debian.org/>). Cette page, mise à jour automatiquement, contient les rapports lintian concernant la dernière version de la distribution (en général *unstable*) générés avec la dernière version de `lintian`.

10.6 Ouvrir un grand nombre de rapports d'un seul coup

Ouvrir un grand nombre de rapports pour le même problème sur un grand nombre de paquet — plus de dix — est une pratique que nous déconseillons. Prenez toutes les mesures possibles pour éviter cette situation. Si le problème peut être détecté automatiquement par exemple, ajoutez un nouveau test dans le paquet `lintian` pour générer une erreur ou un avertissement.

Si vous ouvrez plus de dix rapports sur le même sujet, il est préférable d'indiquer votre intention sur la liste `<debian-devel@lists.debian.org>`. Cela donnera à d'autres développeurs la possibilité de vérifier que le problème existe vraiment. De plus, cela permet d'éviter que plusieurs responsables ne rédigent les mêmes rapports de bogue simultanément.

Quand vous envoyez un grand nombre de rapports sur le même sujet, vous devriez les envoyer à `<maintonly@bugs.debian.org>` pour qu'ils ne soient pas redirigés vers les listes de diffusions.

Chapitre 11

Interaction avec les nouveaux responsables

Ce chapitre décrit les procédures que doivent suivre les responsables Debian quand ils ont affaire à un futur responsable Debian.

11.1 Parrainer un paquet

Parrainer un paquet signifie télécharger un paquet dans l'archive Debian pour un responsable qui n'est pas capable de le faire lui-même : un futur responsable. Parrainer un paquet signifie aussi assumer la responsabilité de ce parrainage.

Les nouveaux responsables éprouvent souvent quelques difficultés pour créer des paquets Debian — ce qui est bien compréhensible. C'est là que le parrain intervient. Il doit vérifier que le paquet est suffisamment bon pour intégrer la distribution. Notez que si le paquet est nouveau les administrateurs FTP devront eux aussi inspecter ce paquet avant de le laisser entrer.

Parrainer un paquet en se contentant de recompiler le paquet ou de signer le téléchargement **n'est franchement pas recommandé**. Vous devez reconstruire le paquet source comme vous le feriez avec vos propres paquets. N'oubliez pas qu'il sera possible de voir que vous avez téléchargé un paquet même si vous avez laissé le nom du futur responsable dans les fichiers de contrôle et changelog.

Si vous êtes responsable de candidature¹ pour un développeur, vous pouvez aussi être son parrain. De cette manière, vous aurez la possibilité de vérifier vous-même comment ce candidat accomplit la partie « Tâches et compétences »² de sa candidature.

¹Application manager

²«Tasks and Skills»

11.2 Aider un nouveau responsable

Consultez la page Aider un futur responsable (<http://www.debian.org/devel/join/nm-advocate>) sur le site du projet Debian.

11.3 Suivre la candidature d'un nouveau responsable

Consultez la page Aide mémoire du responsable de candidature (<http://www.debian.org/devel/join/nm-amchecklist>) sur le site du projet Debian.

Chapitre 12

Aperçu des outils du responsable Debian

Cette section contient un aperçu rapide des outils dont dispose le responsable. Ces outils sont faits pour améliorer le confort des responsables et pour libérer leur temps pour des tâches cruciales.

Certaines personnes préfèrent utiliser des outils de haut niveau, d'autres pas. Debian n'a pas de position officielle sur la question ; tout outil conviendra du moment qu'il fait le boulot. C'est pourquoi cette section n'a pas été conçue pour indiquer à chacun quel outil il devrait utiliser ou comment il devrait faire pour gérer sa charge de responsable Debian. Elle n'est pas non plus destinée à favoriser l'usage d'un outil aux dépens d'un autre.

La plupart des descriptions de ces outils proviennent des descriptions de leurs paquets. Vous trouverez plus d'information dans les documentations de ces paquets.

12.1 `dpkg-dev`

Le paquet `dpkg-dev` contient les outils (y compris `dpkg-source`) nécessaires pour déballer, fabriquer et livrer des paquets Debian source. Ces utilitaires fournissent les fonctionnalités de bas niveau indispensables pour créer et manipuler les paquets ; en tant que tels, ils sont indispensables à tout responsable Debian.

12.2 `lintian`

`lintian` dissèque les paquets pour y repérer des bogues et des manquements aux règles de développement. Il contient des tests automatisés pour vérifier de nombreuses règles et quelques erreurs courantes. L'utilisation de `lintian` a déjà été discutée dans 'Vérifier le paquet avant la mise à jour' page 26 et 'Les rapports Lintian' page 51.

12.3 debconf

Le paquet `debconf` fournit une interface consistante pour configurer les paquets interactivement. Il est indépendant de l'interface et permet une configuration en mode texte, par une interface HTML ou par boîtes de dialogues. D'autres types d'interface peuvent être ajoutés sous forme de modules.

Beaucoup pensent que ce système devrait être utilisé pour tout paquet nécessitant une configuration interactive. `debconf` n'est pas requis par le *Debian Policy Manual* pour le moment ; cela pourra changer dans le futur.

12.4 debhelper

Le paquet `debhelper` regroupe un ensemble de programmes qui peuvent être utilisés dans `debian/rules` pour automatiser les tâches courantes relatives à la fabrication des paquets Debian binaires. Ce paquet contient des utilitaires pour installer différents fichiers, les compresser, ajuster leurs droits et intégrer votre paquet dans le système de menu Debian.

Au contraire de `debmake`, `debhelper` est divisé en plusieurs petits utilitaires qui agissent de manière cohérente. Ce découpage permet un contrôle des opérations plus fin que `debmake`.

12.5 debmake

`debmake` — un précurseur de `debhelper` — est un assistant moins modulaire pour manipuler le fichier `debian/rules`. Il inclut deux programmes principaux : `deb-make`, utile au développeur Debian pour convertir un paquet source normal (non-Debian) en paquet source Debian, et `debstd` qui regroupe le type de fonction que l'on trouve dans `debhelper`.

Le consensus actuel est que l'usage de `debmake` est déconseillé au profit de `debhelper` mais ce n'est pas une erreur d'utiliser `debmake`.

12.6 yada

Le paquet `yada` est un nouvel assistant pour la création de paquets qui a une approche légèrement différente. Il utilise un fichier `debian/packages` pour générer d'autres fichiers nécessaires dans le sous-répertoire `debian/`.

Remarque : `yada` est encore jeune et probablement moins robuste que ses aînés.

12.7 `equivs`

`equivs` est un autre paquet pour faire des paquets. Il est souvent conseillé pour un usage local, si vous avez besoin de faire un paquet pour satisfaire des dépendances. Il est aussi parfois utilisé pour faire des « méta-paquets » qui sont des paquets dont l'unique objet est de dépendre d'autres paquets.

12.8 `cvs-buildpackage`

Le paquet `cvs-buildpackage` permet de mettre à jour ou récupérer des paquets sources dans un référentiel CVS, il permet de fabriquer un paquet Debian depuis le référentiel CVS et il assiste le développeur lors de l'intégration de modifications amont dans le référentiel.

Ce paquet fournit l'infrastructure facilitant l'utilisation de CVS pour le responsable. Il permet de conserver des branches CVS distinctes pour les distributions *stable*, *unstable* et probablement *experimental*.

12.9 `dupload`

Le paquet `dupload` contient un script du même nom pour mettre à jour des paquets dans l'archive Debian, tracer ces mises à jour et les annoncer par courrier électronique automatiquement. Vous pouvez le configurer pour faire des mises à jour à d'autres endroits et avec d'autres méthodes.

Note : l'annonce d'une mise à jour est maintenant prise en charge par le logiciel de gestion de l'archive. `dupload` doit être configuré pour ne plus envoyer de courrier (voir 'Annoncer une mise à jour' page 31).

12.10 `fakeroot`

`fakeroot` simule les privilèges *root*. Cela permet de fabriquer un paquet sans être *root* (en général les paquets installent des fichiers appartenant à *root*). Si vous avez installé `fakeroot` vous pouvez par exemple écrire `dpkg-buildpackage -rfakeroot` en tant qu'utilisateur.

12.11 `devscripts`

Le paquet `devscripts` contient quelques scripts et outils que vous trouverez peut-être utiles pour maintenir vos paquets Debian. Parmi ces scripts, vous trouverez `debchange` qui manipule votre fichier `debian/changelog` depuis la ligne de commande et `debuild` qui est construit au-dessus de `dpkg-buildpackage`.

12.12 `debget`

Le paquet `debget` contient un script qui peut être utile pour télécharger des paquets depuis l'archive Debian. Vous pouvez par exemple l'utiliser pour télécharger des paquets sources.