

MST TIC 2  
Gestion d'entreprise  
Janvier 2005  
<http://migrationlibre.free.fr>

# La migration vers les logiciels libres est-ce raisonnable pour une entreprise?

Maurice Svay - Jocelyn Aubert  
Laurence Jacquet - Nicolas Gramolini



# Sommaire

Introduction.....	3
Le Logiciel Libre.....	3
L'Open Source.....	3
En quoi cela peut intéresser les entreprises ?.....	4
Le logiciel libre et l'entreprise.....	5
Briser les idées reçues.....	5
Les différentes licences.....	6
Les brevets logiciels.....	7
Qu'est-ce qu'un brevet logiciel?.....	7
Description et objectifs.....	7
Différence entre les brevets et le copyright.....	8
La situation en Europe.....	8
Les brevets logiciels, un frein à l'innovation.....	8
La protection des PME et les brevets croisés.....	8
Le phénomène des brevets "mines".....	8
La position des logiciels libres.....	9
Les étapes vers la migration.....	10
Types de migration.....	10
Modes de migration.....	10
Big Bang (la plus rapide).....	10
Transition progressive par groupes (la meilleure).....	10
Transition individuelle.....	10
Quelques conseils facilitant la migration.....	10
Étapes de la migration.....	12
Exemple de migration d'une entreprise.....	14
Les étapes post-migration.....	14
Les entreprises et les logiciels libres : Exemples .....	16
Retour au logiciel propriétaire après l'essai de l'Open Source: le cas de LDLC.....	16
Présentation de LDLC.....	16
Le site LDLC.com.....	16
Faire baisser le coût total de possession.....	16
Gain de temps.....	16
Gagner en indépendance.....	17
Passage d'un logiciel propriétaire à l'open source: le cas de e-TF1.....	17
Présentation de e-TF1.....	17
Le site TF1.fr.....	17
Industrialiser la production de contenu web.....	17
Anticiper la hausse de fréquentation.....	17
Conclusion.....	18
Annexes.....	19
Les principaux acteurs du libre .....	19
Richard M Stallman.....	19
Eric Raymond.....	19
Bruce Perens.....	19
Tim O'Reilly.....	20
Linus Torvalds.....	20
Tristan Nitot (fr).....	20
Nat Makarevitch (fr).....	20
Wine.....	21
Histoire .....	21
Ce que Wine fait.....	21
Où Wine est bon.....	22
Où Wine n'est pas bon.....	22
Wine – alternatives commerciales.....	23
Wine et Visual Basic.....	23
Migration d'application vers Wine.....	23

## Introduction

Depuis quelques années, on entend de plus en plus parler du "monde du libre", des "logiciels libres". Le prix des licences et des logiciels propriétaires font aujourd'hui parti intégrante des budgets de toutes les entreprises, c'est pourquoi n'importe quelle entreprise doit se poser, à un moment ou un autre, la question de la migration vers les logiciels libres.

Nous vous présenterons donc, tout au long de ce site, des termes plus ou moins techniques pour mieux comprendre le monde du libre, les détails des logiciels libres pour les entreprises, les étapes importantes de la migration vers ces logiciels, puis des exemples d'entreprises qui se sont laissées tentées par les logiciels libres.

Afin de mieux pouvoir aborder la migration vers les logiciels libres, voici quelques définitions de termes importants pour mieux comprendre des termes de logiciels libres, le monde de l'open source...

Pour plus de renseignements sur la naissance du libre, nous vous invitons à consulter un historique d'Unix ([http://enacit1.epfl.ch/guide\\_unix/introd\\_unixG.html#historique](http://enacit1.epfl.ch/guide_unix/introd_unixG.html#historique)).

## Le Logiciel Libre

Définition par Richard M. Stallman:

L'expression logiciel libre correspond au terme anglais free software. Comme ce dernier est ambigu en anglais, free pouvant signifier « libre » ou « gratuit », on voit le terme français « libre » de plus en plus utilisé, même en anglais.

L'expression « logiciel libre » fait référence à la liberté pour tous (simples utilisateurs ou développeurs) d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre libertés pour un individu ayant acquis une version du logiciel, définies par la licence de ce logiciel :

- la liberté d'exécuter le programme, pour tous les usages (liberté 0);
- la liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins (liberté 1); pour cela, l'accès au code source est nécessaire;
- la liberté de redistribuer des copies, donc d'aider son voisin, (liberté 2);
- la liberté d'améliorer le programme et de publier ses améliorations, pour en faire profiter toute la communauté (liberté 3) ; pour cela, l'accès au code source est nécessaire.

Source : [http://fr.wikipedia.org/wiki/Logiciel\\_libre](http://fr.wikipedia.org/wiki/Logiciel_libre)

Les logiciels libres au sens de ces 4 libertés étaient monnaie courante dans les années 60-70. On oppose la notion de "logiciel libre" à celle de "logiciel propriétaire" et non pas de "logiciel commercial" dans la mesure où l'on peut commercialiser des logiciels libres (acheter une mandrake à la fnac par ex...) Le logiciel libre comme on le connaît actuellement a été véhiculé par l'essor des UNIX libres depuis les années 80 (BSD en 80, LINUX en 90). En 1998, le source de netscape a été publié sous licence libre: l'arrivée du logiciel libre sur le bureau, directement pour l'utilisateur final et non plus seulement sur les serveurs.

## L'Open Source

Définition de l'Open Source

Eric Raymond a introduit le concept d'opensource bien après l'apparition de celui de logiciel libre.



L' open source est en fait le logiciel libre vu sous l'angle de l'ingénieur.

L'Open Source Initiative a défini les critères nécessaires afin de pouvoir utiliser l'appellation Open Source. Il y a donc une définition de l'Open Source, et dont voici les critères essentiels:

- Libre redistribution
- Code source disponible
- Travaux dérivés possibles

Le fait de disposer des sources d'un logiciel ne suffit pas à dire qu'il est Open Source™. Dans tous les cas, on se référera à la licence d'utilisation du logiciel.

source : [http://fr.wikipedia.org/wiki/Open\\_source](http://fr.wikipedia.org/wiki/Open_source)

L'open source correspond à une sorte de "label" appliqué à certains logiciels dont font partie les logiciels libres dans la mesure où l'opensource définit des règles moins strictes que celles des logiciels libres tout en s'affranchissant de la connotation libertaire associée à ces derniers et qui rassure les entreprises.

### **En quoi cela peut intéresser les entreprises ?**

L'opensource intéresse de plus en plus les sociétés qui ont souvent l'impression d'être pris en otage par les éditeurs de logiciels traditionnels, notamment à travers l'emploi de formats de fichiers et d'échange fermes et par le non-respect des standards (ex : Oracle) qui empêchent les sociétés de passer à la concurrence.

De plus, les sociétés ont besoin de plus d'interopérabilité, d'ouverture et de respect des standards: ce besoin se fait donc de plus en plus sentir dans les sociétés liées aux solutions propriétaires.

Le développement d'applications Open Source procède par consensus de la communauté, ce qui assure une cohérence de l'architecture et des fonctionnalités offertes par chaque application



# Le logiciel libre et l'entreprise

## Briser les idées reçues

Contrairement aux logiciels propriétaires, les logiciels libres sont souvent gratuits. Cette gratuité entraîne souvent un doute pour les entreprises, qui sont en droit de se demander comment un logiciel gratuit peut permettre de travailler tout aussi bien qu'un logiciel payant.

Cependant, de grosses entreprises se sont lancées dans le développement de logiciels libres ces dernières années (Sun, IBM, Red Hat, Novell..).

Étant donné qu'il est difficile de vendre un produit librement distribuable, les sociétés du logiciel libre rentabilisent leur activité autour du service associé. Ainsi ces sociétés peuvent rassurer leurs clients en leur proposant les mêmes services qu'une société commercialisant des logiciels propriétaires.

De plus, un logiciel libre n'est pas forcément gratuit, mais lorsqu'il ne l'est pas, son prix est en général bien plus bas qu'un logiciel propriétaire afin de proposer "la performance au meilleur prix" qui est l'un des principes fondamentaux du monde du libre.

Aussi, le monde du libre n'est pas seulement réservé à de petites entreprises au budget réduits ou à la recherche: de grosses entreprises ont récemment migré leurs systèmes informatiques vers des systèmes libres. On pourra en citer quelques unes telles que Le Monde, les groupes Auchan et Cora, Yahoo et de nombreuses autres entreprises qui utilisent Apache sur leurs serveurs. Ce point sera abordé plus précisément dans la partie "Des exemples d'entreprises".

Suite à toutes ces migrations qui semblent viable, on peut clairement dire que le logiciel libre est une solution possible pour une entreprise, à condition de bien calculer tous les coûts de migration et de formation des utilisateurs et ce que cela peut apporter à l'entreprise.

Une autre idée reçue pourrait être la suivante: si on migre vers les logiciels libres, les applications seront totalement différentes et on ne pourra plus jamais revenir à une solution propriétaire. C'est tout à fait faux: les logiciels libres respectent les systèmes standards de fichiers : ainsi un fichier créé à l'aide d'un logiciel libre sera très souvent compatibles avec d'autres logiciels, ce qui est rarement le cas lorsqu'on utilise un logiciel propriétaire.

Il existe aussi de nombreux logiciels libres tout à fait comparables aux logiciels propriétaires utilisés en entreprise: ces logiciels peuvent remplacer les logiciels propriétaires autant au niveau des serveurs que des machines des utilisateurs (avec des logiciels de base de données ou de traitement de texte par exemple).

Voici une sélection de **10 logiciels** du monde du logiciel libre faite par la société IdealX:

- O.S.: Linux
- Annuaire: OpenLDAP
- Serveur Web: Apache
- Serveur de ressources: Samba
- Base de données: PostgreSQL
- Serveur d'applications: Tomcat
- Suite bureautique: OpenOffice
- Administration: Webmin
- Administration: APT
- Messagerie: Postfix

Ces logiciels libres sont de plus en plus utilisés par les entreprises qui désirent migrer vers le logiciel libre. A cette liste pourra s'ajouter un navigateur web tel que Firefox, une messagerie instantanée comme Gaim ou encore un logiciel de retouche d'images comme The GIMP, comparable à Photoshop.



## Les différentes licences

Une licence définit les droits d'utilisation, de copie, de modification et de distribution d'un logiciel. Les logiciels sans licence sont dits du domaine public. Pour tout les autres logiciels, il faut définir la licence sous laquelle sera distribué le logiciel afin de le protéger.

Voici les droits, les obligations et les libertés des principales licences utilisées pour le développement et la distribution des logiciels libres.

Toutes ces licences sont compatibles OSI (Opensource Initiative)

### GPL

GPL veut dire **GNU General Public License**

- Droit d'Utiliser
- Droit de Modifier
- Droit de Distribuer

**Mais** les droits de l'utilisateur sont inaliénables:

Obligation du passage en GPL:

- Divulgateur obligatoire du source
- Même liberté accordée aux autres
- Éclatement du source impossible

Pour plus d'informations sur cette licence: [www.fsf.org](http://www.fsf.org)

### LGPL

LGPL veut dire **GNU Lesser General Public** (License Licence Publique Générale Limitée)

La LGPL est une **GPL** modifiée, spécialisée pour les bibliothèques de programmation. Globalement, les droits sont les mêmes. Ils ont été reformulés pour le cas spécifique des bibliothèques de programmation.

Cependant, la **FSF** (Free Software Foundation) et le projet **GNU** dissuadent d'utiliser cette licence, car ils estiment que stratégiquement ce n'est plus nécessaire.

### BSD

BSD veut dire **Berkeley Software Distribution License**. Développée par l'université de Berkeley (Californie) depuis 1995.

- Droit d'Utiliser
- Droit de Modifier
- Droit de Distribuer
- **Mais** la source est "propriétarisable" par tous:
  - Pas d'obligation de conservation de la licence (seule une mention indiquant l'auteur original doit persister):
  - Divulgateur non obligatoire du source
  - Résultat: Des utilisateurs lésés

Pour plus d'informations sur cette licence: [www.freebsd.org](http://www.freebsd.org)

## MPL

MPL veut dire Mozilla Public License

- Droit d'Utiliser
- Droit de Modifier
- Droit de Distribuer
- Et l'éditeur original conserve le droit de récupérer les modifications des contributeurs pour les intégrer à des logiciels propriétaires.

Pour plus d'informations sur cette licence: <http://www.mozilla.org/MPL/>

## Apache

Apache vient de « **a patchy** server », en raison du mode de développement par patches (correctifs) de ses premières versions.

- Distribution de versions binaires et de code source
- La distribution de versions binaires est autorisée dans le cas où elle reproduit la notice de copyright
- Dans tous les cas, une distribution doit mentionner qu'il provient d'un logiciel développé par Apache:

« This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>) »

- Licence non copyleftée
- N'oblige pas à publier le code source

Pour plus d'informations sur cette licence: <http://www.apache.org/licenses/>

## CeciLL (fr)

Acronyme pour **Ce(A)C(nrs)I(NRIA)L(ogiciel)L(ibre)**, CeciLL est une Licence libre de droit français compatible avec la licence GNU GPL , elle possède donc les mêmes droits que la licence GNU GPL

Le texte complet et d'avantage d'informations sur cette licence: <http://www.cecill.info/> (fr)

## Les brevets logiciels

### Qu'est-ce qu'un brevet logiciel?

#### *Description et objectifs*

Un brevet est une description textuelle d'une invention destinée à la protéger. Les principaux critères de brevetabilité sont les suivants:

- Nouveauté
- Inventivité
- Application industrielle

Un brevet logiciel est alors l'application de ces critères de brevetabilité au domaine de l'informatique.

Depuis 1985 les logiciels sont brevetables aux États Unis et au Japon, et dans une moindre mesure, depuis 1986 en Europe.

#### *Différence entre les brevets et le copyright*

Les brevets sont issus du monde industriel et les copyrights du monde artistique. Le copyright s'applique en Europe sur le code source d'un logiciel, mais la protection qu'il offre est des plus limitée. En effet, comme le montre la jurisprudence, il est très difficile d'obtenir gain de cause dans un procès pour plagiat, car il faut pouvoir prouver le vol d'un support matériel ou la corruption d'un employé de l'entreprise.

Le brevet quant à lui est déposé dans des registres nationaux et internationaux, ce qui n'est pas le cas pour le copyright. L'un des critères de brevetabilité étant la nouveauté, tout dépôt de brevet est soumis à une recherche d'antériorité.

#### *La situation en Europe*

A l'heure actuelle, les logiciels sont partiellement brevetables en Europe :

- les algorithmes et processus triviaux (éléments de base pour la création d'un logiciel Exemples : if, for, bouton...) soumis au copyright et non brevetables
- les logiciels : copyright et éventuellement brevetables dans certains cas.

Une définition floue : les programmes "ayant des effets techniques" sont "non exclus de la brevetabilité". On considérera donc qu'il s'agit des logiciels en tant que solution technique à un problème technique.

Exemple: un driver d'imprimante, le problème technique étant celui du pilotage du matériel "imprimante".

Actuellement, un projet de loi a été soumis à la Commission européenne pour la brevetabilité totale des logiciels, y compris les algorithmes et les processus triviaux. Ce projet de loi est relativement dangereux pour les logiciels libres, comme nous allons le montrer ci-après, et donc pour leur éventuelle adoption dans les entreprises.

NB: La situation actuelle au niveau législatif, comme elle a été décrite ici, n'est pas réellement appliquée. En effet l'OEB (office européen des brevets) a autorisé le dépôt de brevets sur les logiciels (de manière totale) depuis plusieurs années et considère donc que cette brevetabilité fait au jour d'aujourd'hui, jurisprudence dans le domaine.

#### **Les brevets logiciels, un frein à l'innovation**

##### *La protection des PME et les brevets croisés*

L'un des principaux objectifs des brevets est la protection de la propriété intellectuelle des entreprises. Or on constate que ce n'est pas le cas en pratique pour l'industrie du logiciel.

De grandes firmes utilisent actuellement d'importants portefeuilles de brevets pour faire pression sur des PME innovantes en les contraignant à des accords de brevets croisés. Un brevet croisé permet à une entreprise d'utiliser un brevet d'une entreprise comme si c'était l'un de ses propres brevets. Ainsi le principe du brevet croisé permet aux grandes entreprises d'enrichir très facilement leur portefeuille de brevets et ne permet pas de protéger les PME.

##### *Le phénomène des brevets "mines"*

La technique du brevet "mine" permet aux entreprises de déposer des brevets sur des innovations sans que celles-ci aient de réalisation industrielle, on dépose un procédé sans le sortir sur le



marché, on évite ainsi les coûts de la réalisation et on rentabilise le dépôt du brevet par les actions en justice.

Ce phénomène permet aussi de contrôler l'innovation technologique et par là même de pouvoir rentabiliser plus longtemps les logiciels actuels tout en empêchant la concurrence d'innover. Les brevets logiciels sont donc en ce sens un frein à l'innovation et ainsi au développement des logiciels libres.

### *La position des logiciels libres*

La menace de la brevetabilité totale en Europe, issue du projet de loi présenté ci-avant, plane sur l'adoption du libre par les entreprises. En effet les brevets logiciels peuvent limiter la "liberté" d'un logiciel libre, en nécessitant le règlement de royalties pour les utilisateurs de ces logiciels. De nombreux exemples aux États Unis peuvent en témoigner.

Ainsi dans les années 80, le logiciel PGP de cryptographie à clef publique a été contraint à être distribué sous une licence non libre par les détenteurs de brevets sur les systèmes à clefs publiques.

Un exemple plus connu est celui du format de compression d'image GIF. En effet, ce format utilise un algorithme de compression, le LZW qui est breveté par la société Unisys. En 1985, Unisys avait communiqué sur le sujet et assurait à tous la gratuité de la compression GIF pour une utilisation non commerciale. Cependant en 1996 la direction d'Unisys changea d'avis et demanda à tous les utilisateurs de GIF de s'acquitter d'une licence de 5000\$. Il s'agissait donc de l'utilisateur lambda, du développeur de logiciels au webmaster gérant un site présentant des images au format GIF. L'utilisation du format GIF fut donc refusée au logiciel libre, aux États Unis tout auteur de logiciel libre permettant de produire des fichiers GIF peut être poursuivi en justice.

Le format MP3 de compression sonore est également breveté par Thomson et le Fraunhofer institute. En 1998, des développeurs américains qui avaient mis à disposition sur internet des logiciels libres permettant de décoder ce format ont été menacés d'atteinte à la propriété intellectuelle et forcés de les retirer. La brevetabilité totale des logiciels en Europe les rendraient donc aussi indisponibles en Europe.

La ville de Munich a entrepris depuis 2003 la migration de l'ensemble de son parc informatique vers les logiciels libres (essentiellement les postes bureautiques vers linux + mozilla et openoffice). Cependant cette migration a été suspendue en septembre 2004 pour cause d'évaluation des risques juridiques et financiers encourus par cette migration dans le cas de l'adoption de la brevetabilité totale en Europe. La bonne nouvelle pour les entreprises décidant de migrer vers les logiciels libres est que la ville de Munich a décidé de poursuivre son processus de migration, créant ainsi un précédent. Le maire de la ville de Munich a réaffirmé son soutien aux solutions libres et espère surtout faire changer d'avis le gouvernement allemand sur la question de la brevetabilité des logiciels.

Pour de plus amples informations à propos des brevets logiciels, nous vous recommandons de lire quelques articles:

- [Brevets logiciels en Europe : un rapide état des lieux](#) sur le site de la [FFII](#)
- [Les brevets logiciels sur désenveloppez](#)
- [Sauver l'Europe des brevets sur les logiciels](#) sur [gnu.org](#)
- [Le maire de Munich rassure les partisans de Linux](#) sur [zdnnet.fr](#)



# Les étapes vers la migration

## Types de migration

On peut différencier deux types de migration :

- La migration partielle, consistant à remplacer uniquement les applications client : bureautique ou développement par exemple.
- La migration totale, consistant à remplacer tous les logiciels propriétaires (application client et serveur : notamment les systèmes d'exploitation) par des logiciels libres.

## Modes de migration

Il existe trois types de mode de migration

### Big Bang (la plus rapide)

Tous les utilisateurs passent au nouveau système le même jour. Dans la pratique, il est vraisemblable que la migration soit planifiée sur une fin de semaine ou durant les vacances. L'avantage réside dans l'absence de structures de double accès et de double maintenance par les équipes techniques. Les inconvénients résident dans un niveau de risque très élevé et dans la très forte mobilisation de ressources durant la migration. Ce schéma semble ne pouvoir être appliqué que dans les petites administrations. Dans la mesure du possible, il faut **éviter la migration BIG BANG**. Ce type de migrations nécessite le contrôle de tant de variables qu'elles échouent en général. Si c'est le cas, ce n'est pas, en général, du fait d'une défaillance de l'Open Source mais de celle du management.

### Transition progressive par groupes (la meilleure)

Les utilisateurs migrent par groupes. On fait migrer en général des groupes fonctionnels complets afin de minimiser les partages de données et les difficultés de travail de groupe. Les risques peuvent être limités et les ressources gérées par l'adéquation de la taille des groupes. Il peut être possible de réaliser un remplacement du matériel par roulement dans le même temps, en effectuant une mise à jour des postes de travail retirés d'un groupe avant de les attribuer au groupe suivant.

### Transition individuelle

Similaire à la transition progressive par groupes mais avec une taille de groupe d'une personne. Cette méthode au goutte-à-goutte mobilise peu de ressources mais elle est peu efficace et sans doute peu adaptée aux grandes administrations. Elle peut cependant être appropriée aux projets pilotes.

## Quelques conseils facilitant la migration

Quelques précautions à appliquer avant la migration, afin de maximiser les chances de réussite :

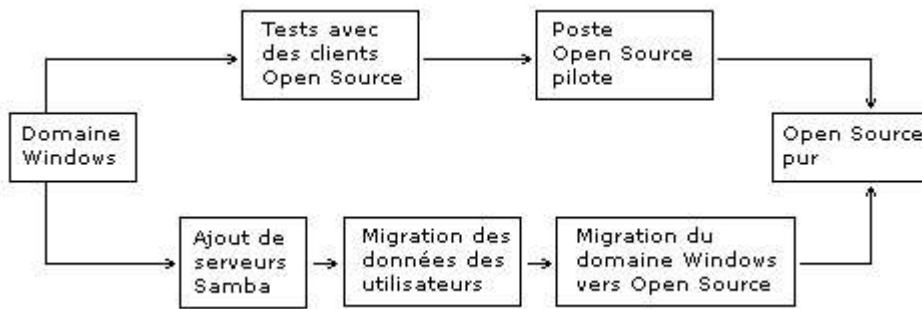
- Imposer que le développement web maison ou par des prestataires produise un contenu visualisable sur tous les navigateurs actuels, en particulier les navigateurs OSS. Cela doit être une bonne pratique dans tous les cas car l'administration ne doit pas nécessiter de logiciel spécifique pour accéder au contenu en ligne. Le respect des standards du web définis par le W3C (World Wide Web Consortium) et l'utilisation d'outils tels que weblint facilitant le contrôle de compatibilité des pages web, permettent d'obtenir des pages web compatibles avec tout navigateur.
- Décourager la prolifération incontrôlée des macros et scripts dans les documents et feuilles de calculs ; chercher d'autres moyens de réaliser la fonctionnalité. Cela aussi doit être une

- bonne pratique car l'utilisation de ceux-ci est un biais classique d'infection par les virus. De même, les macros peuvent facilement être utilisées pour dérober des informations et détourner des documents (par exemple, elles permettent que le document affiche des éléments différents à l'écran et à l'impression).
- Imposer l'utilisation des formats de fichiers ouverts standard, par exemple PostScript et PDF. Il existe un certain débat sur le caractère ouvert des standards PostScript et PDF. Il s'agit plus d'un débat sur la définition stricte et en particulier sur le « qui contrôle le standard ? ». En réalité, il s'agit des seuls formats standard largement utilisés actuellement, dont les définitions soient publiques et qui puissent être utilisés sans restriction. Des tentatives pour créer de véritables standards ouverts fondés sur XML sont apparues et OpenOffice.org en est une implantation. Cependant le seul critère technique XML ne rend pas le format ouvert. En particulier, il faut éviter d'utiliser des formats de fichiers propriétaires pour les fichiers qui ne sont prévus que pour la lecture et non la modification par le destinataire. À nouveau, il s'agit d'une bonne pratique car ceux-ci sont un vecteur de diffusion des virus. L'utilisation de tels formats verrouille l'administration avec un éditeur pour une durée considérable. De plus, ces formats peuvent contenir des quantités considérables de méta-données y compris, en particulier, du texte supprimé préalablement qui peut se révéler embarrassant pour l'administration s'il est vu par des tiers ; il n'est pas difficile d'accéder à ces méta-données.
  - Lors de la rédaction de documents à plusieurs, utiliser le format le plus petit dénominateur commun. Par exemple, utiliser le format MS-Word 97 de préférence au format MS-Windows 2000. Cela augmente la probabilité de participation par des applications OSS.
  - Utiliser des protocoles standard. Les protocoles standard ouverts sont ceux définis comme libres de brevets et disposant d'une implantation OSS. Différents ensembles de standards nationaux existent tels que E-gif au Royaume-Uni, OSOSS aux Pays Bas et SAGA en Allemagne. L'objectif et le contenu de ces différents cadres sont assez variables mais en général ils sont imposés.
  - Développer des systèmes fondés au minimum sur un modèle 3-tiers dans lequel le code applicatif est indépendant de l'interface homme-machine et des méthodes d'accès aux données. Par exemple, si possible, ayez une interface par navigateur qui fonctionne avec un navigateur OSS. Le développement d'applications selon ce principe modulaire en facilite la migration élément par élément. Cela ne réduira pas seulement l'échelle de toutes les phases de migration mais aussi le risque de défaillance. Les applications à client monolithique traditionnel sont notoirement difficiles à maîtriser.
  - Imposer que tout nouveau développement soit portable. Cela implique l'utilisation de langages portables standardisés tels que le C ANSI, Java, Python et Perl ainsi que l'utilisation exclusive de bibliothèques et boîtes à outils I.H.M. multiplates-formes telles que wxWindows et FOX toolkit. Éviter les langages et API spécifiques. Éviter de développer des applications nécessitant la présence d'autres applications propriétaires.
  - Éloigner les utilisateurs des outils de courriel qui utilisent des formats de stockage de courriel propriétaires et communiquent avec les serveurs à l'aide de protocoles propriétaires. De nombreuses applications stockent le courriel à l'aide d'IMAP. Si possible, trouver un moyen de stocker les carnets d'adresses et les informations calendaires dans un format ouvert.



## Étapes de la migration

Schéma général :



Dans cette partie, nous détaillerons les étapes d'une migration totale, c'est-à-dire, changer le système d'exploitation ainsi que les applications. Bien sûr, il existe un grand nombre de combinaisons en fonction des conditions initiales.

Avant tout, il faut procéder à une phase de collecte d'information (ou justement les conditions initiales) : il est nécessaire de former, ou recruter une équipe compétente d'informaticiens chargée de la migration. Cette équipe aura pour première mission, de déterminer la faisabilité de la migration (vérification de la portabilité des documents existants, la conversion des formats, le calcul des coûts, l'estimation de la durée de travail...), d'observer l'architecture du système, les protocoles et standards utilisés ainsi que l'environnement physique (bande passante, réseau, localisation). Puis de déterminer le besoin de l'entreprise, c'est-à-dire, quels logiciels sont les plus adaptés aux besoins de l'entreprise, et également aux capacités de l'équipe informatique en place (pré requis (langue) et particularités) et enfin évaluer le produit Open Source afin de s'assurer qu'il assure toutes les fonctionnalités nécessaires. S'il existe une application essentielle à l'entreprise n'ayant pas d'équivalent Open Source, il existe une solution : Wine , qui permet d'exécuter des applications Windows sur un système Unix.

Avant d'effectuer quoique ce soit, sur les stations de travail et/ou sur les serveurs, il est indispensable d'effectuer de réelles sauvegardes, afin de restaurer le système en cas de problème majeur.

Enfin, il convient de proposer un plan minutieux expliquant les démarches successives de mise en place jusqu'aux conditions finales.

La première étape réelle de la migration, consiste en une migration partielle, pour « habituer » les futurs utilisateurs, et ainsi, ne pas les déstabiliser totalement. Ainsi, les applications qui seront utilisées sur le nouveau système libre (tout au moins, les applications libres existantes sur un système d'exploitation propriétaire), sont installées sur le système d'exploitation connu et maîtrisé. Ce contact initial permet une prise en main plus aisée.

Les acteurs de la migration, ont tout intérêt d'utiliser cette approche de la migration, car le facteur humain est le facteur le plus important. Ne pas tenir compte des appréhensions, des questions des utilisateurs rime forcément avec échec. De plus, cette approche, permet de détecter des incompatibilités entre format de fichiers par exemple.

Liste non exhaustive de logiciels libres existants sur système propriétaire (Windows)

Logiciel	Avant	Après
<b>Bureautique</b>	Microsoft Office 590,67 €* Inclus dans Windows	Open Office.org
<b>Navigateur Web</b>	Microsoft Internet Explorer Inclus dans Windows	Mozilla Firefox
<b>Client e-mail</b>	Microsoft Outlook Express Inclus dans Office	Mozilla Thunderbird
<b>Compilateur C/C++</b>	Borland C++ Builder 3 339 €* Inclus dans Windows	Dev-C++
<b>Création/Retouche d'images</b>	Adobe Photoshop 1059 €* Inclus dans Windows	The Gimp
<b>Serveur Web</b>	Microsoft IIS 3004 € avec Windows Server 2003*	Apache

\* : tarif professionnel HT à l'unité (mise à jour non comprise)

En parallèle, à ce changement du côté utilisateur, un changement du côté serveur, peut-être envisagé, par exemple le serveur DHCP, le serveur de noms DNS ou encore les serveurs de bases de données. L'ajout d'un serveur Samba (paquetage serveur de fichiers et d'impression pour les systèmes OSS).

Cette opération sera totalement transparente pour les utilisateurs, à condition qu'elle soit effectuée d'une manière correcte.

Après une période d'adaptation aux nouveaux logiciels, où les acteurs de la migration, auront très certainement, besoin d'intervenir auprès des utilisateurs pour les assister et les conseiller dans la découverte de l'interface et des nouvelles fonctions offertes par les logiciels Open Source nouvellement installés. L'équipe découvrira à coup sûr des points obscurs pour les utilisateurs, qui n'avaient pas été envisagés lors de l'étude de faisabilité.

Dès que les utilisateurs semblent maîtriser parfaitement les nouveaux outils, ce qui peut prendre plus ou moins de temps, selon leur rapport à l'informatique ; un informaticien aura plus d'aisance à acquérir la maîtrise d'un nouvel outil informatique qu'un agent de saisie par exemple ; la dernière phase de la migration peut-être enclenchée : le changement de système d'exploitation. Ce changement peut-être considéré comme le plus déstabilisant; l'utilisateur perd alors tout repère dans son environnement de travail, même si désormais, les systèmes d'exploitation libres adoptent de plus en plus, d'interfaces quasi-similaires à celle bien connue du système de Microsoft.

Un accompagnement plus ou moins important reste nécessaire, afin de rassurer les utilisateurs, car ce sont les premiers concernés par la migration, et sa réussite dépend de leur satisfaction et leur aptitude face au nouveau système.



## Exemple de migration d'une entreprise

Prenons l'exemple d'une entreprise utilisant un système d'exploitation ainsi que des logiciels propriétaires, désirant migrer vers l'Open Source de façon totale. Le tableau ci-dessous, montre la configuration de départ et l'équivalent Open Source proposé.

Usage	Logiciel avant migration	Logiciel après migration	Commentaires
<b>Stations</b>			
<b>Système d'exploitation</b>	Microsoft Windows 2000	RedHat v.8.0	<ul style="list-style-type: none"> <li>• Système simple d'utilisation</li> <li>• Interfaces disponibles (KDE, Gnome) proches de celle de Windows</li> </ul>
<b>Bureautique</b>	Microsoft Office	OpenOffice.org	<ul style="list-style-type: none"> <li>• fonctionne sous Windows pour migration partielle</li> <li>• meilleures interprétations des formats de fichiers Microsoft</li> <li>• bonne alternative à Ms-Office</li> </ul>
<b>P.A.O. (Publication Assistée par Ordinateur)</b>	FrameMaker, Pagemaker, QuarkXpress	Scribus / OpenOffice.org	si OpenOffice.org est déjà installé, pourquoi installer un autre logiciel ?
<b>Génération de fichiers PDF</b>	Adobe Acrobat	OpenOffice.org	
<b>Courriel</b>	Microsoft Outlook	Evolution / Thunderbird	<ul style="list-style-type: none"> <li>• interface similaire à celle de Outlook</li> <li>• fonctionnalités très utiles (dossiers virtuels, anti-spam, contrôle de virus)</li> </ul>
<b>Navigateur Web</b>	Microsoft Internet Explorer	Mozilla Firefox	<ul style="list-style-type: none"> <li>• fonctionne sous Windows pour migration partielle</li> <li>• meilleure interprétation des standards du Web</li> <li>• fonctionnalités très utiles (navigation par onglet, blocage des pop-up)</li> </ul>
<b>Antivirus</b>	Tout antivirus propriétaire	aucun	Un système Open Source correctement configuré craint peu les virus, de plus un agent de transport de courriel correctement configuré bloque la plupart des menaces
<b>Pare-feu</b>	Tout pare-feu propriétaire	aucun	Tout système d'exploitation Open Source dispose de pare-feux par filtrage de paquets internes (ex: iptables, ipfilter ou Packetfilter)
<b>Bases de données</b>	Microsoft Access / Microsoft SQL Server	MySQL / PostgreSQL	Attention, tout de même aux pertes encourues, dues à l'absence de certaines fonctions dans MySQL comme par exemple les procédures stockées.
<b>Serveurs</b>			
<b>Courriel</b>	Agent de transport (MTA - Mail Transport Agent)	Exim / Postfix	
<b>Courriel</b>	Agent d'accès au courriel (MAA - Mail Access Agent)	Courier-IMAP / Cyrus	
<b>Services Web</b>	Microsoft IIS	Apache	
<b>DNS</b>	Microsoft DNS	Bind / MyDNS	
<b>Travail de groupe</b>	Microsoft Exchange	Evolution	

## Les étapes post-migration

Une fois, la migration effectuée, et lorsque le système en place fonctionne parfaitement, les administrateurs du système peuvent alors, lorsque cela s'avère nécessaire et de façon quasi-

identique que sur un système propriétaire, effectuer des mises à jour. Seule différence, ces mises à jour, peuvent provenir soit de l'éditeur et ou du groupe de développeur, mais également de techniciens de l'entreprise, désireuse de modifier le programme aux besoins réels de l'entreprise.

L'utilisation de l'Open Source, peut, au premier abord, susciter une inquiétude : manque de fiabilité, de documentation, de support technique ; mais il faut savoir que le fait que le code soit libre permet à quiconque d'améliorer ou de corriger un quelconque « bug », ce qui permet une excellente réactivité face aux failles de sécurité, au nécessité de mise à jour.

Pour ce qui est de la documentation, il est vrai qu'il est plus difficile de trouver un livre sur un logiciel libre que sur son équivalent propriétaire, qu'importe, il existe toujours un site de documentation disponible sur Internet ; seul défaut : très souvent ceux-ci sont destinés à un public anglophone.

Le support technique des logiciels libres est en passe de devenir aussi présent que pour les logiciels propriétaires ; de nombreux éditeurs éditant des logiciels développent pour les entreprises des services techniques permettant ainsi d'obtenir une aide technique.



## Les entreprises et les logiciels libres : Exemples

De grandes entreprises participent au développement des logiciels libres, soit en sponsorisant des projets, soit en rémunérant les développeurs de ces projets.

On peut citer:

- IBM : Participe activement au développement de GNU/Linux
- Sun Microsystems : est à l'origine de la suite bureautique OpenOffice.org
- Novell : a fait l'acquisition de plusieurs sociétés spécialisées comme Ximian et Suse Linux
- Apple : a bâti son dernier système (Mac OS X) à partir de Darwin
- est parti de KHTML pour créer le moteur Webcore qui est utilisé dans le navigateur Safari
- Microsoft : a diffusé quelques projets (WiX, FlexWiki, WTL) en source ouverte

D'autres entreprises utilisent les logiciels libres. On peut citer:

- Yahoo! : Le premier portail du Web fonctionne grâce à PHP
- Hachette Multimédia : La dernière encyclopédie sur CD-Rom utilisent des technologies de la Fondation Mozilla
- Le Monde: La publication du journal en ligne s'appuie sur des systèmes libres et des technologies de la Fondation Mozilla
- Auchan

## Retour au logiciel propriétaire après l'essai de l'Open Source: le cas de LDLC

### Présentation de LDLC

Le groupe LDLC a été créé en 1996 par Laurent de La Clergerie. Il compte 180 salariés et devrait générer 158 millions d'euros de chiffre d'affaire pour 2004. Les activités du groupe se répartissent entre la vente de matériels et logiciels informatiques aux particuliers et aux entreprises, l'importation et l'information en ligne.

### Le site LDLC.com

Le site LDLC.com vend du matériel et des logiciels informatiques sur Internet. Lancé en 1997, il fait figure de pionnier et reste un des premiers sites marchands dans le secteur high tech. En 2003, après 6 six ans d'existence, le site fait table rase et décide remplacer son infrastructure open source par des logiciels Microsoft.

D'après Laurent de La Clergerie, directeur général de LDLC, les objectifs de ce changement étaient multiples:

- faire baisser le coût total
- gain de temps
- gagner en indépendance

### Faire baisser le coût total de possession

D'après le directeur général, le coût de l'infrastructure logicielle (coût des licences et du support) est inférieur au coût d'une solution analogue mais open source. Même si les licences sont plus onéreuses, un expert Microsoft reviendrait entre 20% et 30% et moins cher qu'un expert open source. Il explique que cet écart proviendrait de faible nombres d'expert de cette dernière catégorie et que la loi du marché fait que leur tarifs sont plus élevés.

### Gain de temps

En utilisant les logiciels propriétaires, le service informatique du site n'a pas perdu de temps « à chercher des solutions, à vérifier leur fiabilité, à développer et à intégrer des briques ». Il démontre également que la migration a été très rapide: il a suffi de 10 mois (moins d'un an) pour que le



système d'information repose sur des logiciels Microsoft et les développements spécifiques ont été achevés en 3 mois à l'aide de Visual Basic.

### **Gagner en indépendance**

En utilisant des briques logiciels Microsoft lors du développement, les connaissances du système ne sont plus concentrées dans l'équipe de développement. Les informaticiens peuvent gérer le système sans pour autant avoir participé au développement logiciel.

Il faut noter que l'utilisation de ces solutions a été rendue possible par les moyens dont disposait la société, largement supérieurs aux débuts. De plus, le l'open source n'a pas été complètement abandonné, notamment dans pour la répartition des charges entre les différents serveur.

## **Passage d'un logiciel propriétaire à l'open source: le cas de e-TF1**

### **Présentation de e-TF1**

La société e-TF1 est la branche de TF1 dédiée à la création de sites web et de portails internet. Cette activité a été en 1998 et reste détenue à 100% par TF1.

### **Le site TF1.fr**

En 2000, le site a effectué une mise à plat de son système de publication pour industrialiser la production de contenu web et anticiper la hausse de fréquentation. Le site TF1.fr est l'un des sites d'information et de divertissement les plus visités en France.

### **Industrialiser la production de contenu web**

Avant la remise à plat, la production de contenu web était faite à l'aide d'outils comme Dreamweaver ou Frontpage. Une solution s'articulant autour de Vignette (logiciel adossé à TCL) et Oracle a permis d'améliorer le workflow et a ouvert la voie à des publications transversales. Cette solution, bien que permettant de mieux répondre aux besoins pour des actions événementielles, imposait des contraintes de structuration trop fortes.

Pour le lancement de nouveaux services transactionnels (en opposition à rédactionnels), e-TF1 a opté pour un système d'exploitation GNU/Linux et le langage de script open source PHP. Ainsi, la mise en place de services payant a pu être mise en oeuvre grâce au développement d'un système de gestion d'accès aux comptes utilisateurs.

### **Anticiper la hausse de fréquentation**

Le système Vignette permettait déjà de gérer un cache et amortir la hausse de fréquentation. Pour être adopté, la solution Linux/PHP a du passer des tests de montée en charge qui se sont révélés probants.

Sources:

- [http://www.microsoft.com/france/temoignages/info.asp?mar=/france/temoignages/2004/p\\_ldlc\\_0504.html](http://www.microsoft.com/france/temoignages/info.asp?mar=/france/temoignages/2004/p_ldlc_0504.html)
- <http://linuxfr.org/>
- [http://solutions.journaldunet.com/itws/030619\\_it\\_etf1.shtml](http://solutions.journaldunet.com/itws/030619_it_etf1.shtml)



## Conclusion

Nous avons vu que les logiciels libres répondent à de réels besoins en entreprise. Les échanges croissants avec d'autres entités font qu'il devient nécessaire d'adopter des solutions capables d'interagir avec d'autres solutions informatiques sans problèmes. La maîtrise totale et l'interopérabilité sont alors des critères importants que satisfont pleinement les logiciels libres grâce à leurs multiples avantages.

Cependant, leur adoption rencontre des obstacles que sont la méconnaissance des solutions, des idées reçues et les brevets logiciels. Une fois cette étape franchie, la migration peut commencer en suivant plusieurs rythmes, le plus adapté étant généralement la transition progressive.

Les changements effectués sont d'ailleurs facilités et accélérés si l'entreprise a pris soin d'utiliser des technologies standardisées.

Des exemples concrets montrent que l'utilisation de logiciels libres en entreprise est une réalité, rendue possible même dans les grandes structures

Alors que la majorité des entreprises privées commencent à se poser la question de la migration, certaines ont déjà franchi ce cap et l'impulsion d'un mouvement de plus grande ampleur viendra peut être des organismes publics qui choisissent de plus en plus ces solutions.



## Annexes

### Les principaux acteurs du libre

#### Richard M Stallman



Richard Matthew Stallman (né à Manhattan, le 16 mars 1953), aussi connu sous les initiales RMS, est le fondateur du mouvement pour les logiciels libres. Il est célèbre pour avoir initié le projet GNU et avoir rédigé (avec Eben Moglen) la première licence permettant de réglementer la distribution d'un logiciel avec son code source, modifiable (un logiciel libre) : la licence publique générale GNU (GNU GPL). Il a inventé le concept de Copyleft (« gauche d'auteur »).

Il fut chercheur informaticien au Massachusetts institute of technology, puis a créé la Free software foundation (FSF) en 1983. « Free » est à comprendre dans le sens de « libre » et non de « gratuit ». Il a également écrit l'éditeur de texte GNU Emacs, le compilateur C Gcc, et le débogueur Gdb.

La biographie complète sur [http://fr.wikipedia.org/wiki/Richard\\_Stallman](http://fr.wikipedia.org/wiki/Richard_Stallman)

#### Eric Raymond



Eric Steven Raymond, connu également sous ses initiales ESR, est un hacker célèbre à qui l'on doit notamment la popularisation du terme open source par opposition à free software. Cette divergence terminologique a pour but de souligner une divergence de vues idéologique d'avec Richard M. Stallman et la FSF. Alors que ces derniers vantent les mérites plutôt éthiques et philosophiques des logiciels libres, ESR vante la qualité des logiciels développés de manière libre d'un pur point de vue de la technique informatique.

Ses productions informatiques à proprement parler sont moins connues que celles d'autres hackers célèbres tels que Richard M. Stallman ou Linus Torvalds, ce qui explique qu'il soit parfois contesté dans le milieu libre.

La biographie complète sur [http://fr.wikipedia.org/wiki/Eric\\_Steven\\_Raymond](http://fr.wikipedia.org/wiki/Eric_Steven_Raymond)

#### Bruce Perens



Bruce Perens est un avocat de Linux et du logiciel open source depuis longtemps. Jusqu'en 1997 Bruce était à la tête du projet Debian, un effort bénévole pour créer une distribution de Linux ne comprenant que du logiciel open source. Pendant son travail sur le projet Debian, Bruce a contribué à la maturation du projet social Debian (Debian Social Contract), un ensemble de conditions sous lesquelles un logiciel peut être considéré comme relevant d'une licence suffisamment libre pour être inclus dans la distribution Debian. Le contrat social Debian est l'ancêtre direct de l'actuelle définition de l'Open Source (Open Source Definition ou OSD).

Après s'être retiré de l'intendance de Debian, Bruce continua ses efforts d'évangélisation en faveur de l'Open Source en créant et dirigeant Software in the Public Interest, et en démarrant, avec Eric Raymond, l'Open Source Initiative.

Lorsqu'il n'évangélise pas activement pour le logiciel Open Source, Bruce travaille pour les studios d'animation Pixar.

source: [www.oreilly.fr](http://www.oreilly.fr)



## Tim O'Reilly



Tim O'Reilly est le fondateur et CEO d'O'Reilly & Associates, Inc., l'éditeur dont les ouvrages sont considérés comme des références en matière d'Open Source telles que Perl, Linux, Apache, et l'infrastructure de l'Internet. Tim a convoqué le premier «Sommet de l'Open Source» afin de rassembler les leaders de communautés Open Source majeures, et a été un promoteur actif du mouvement Open Source en écrivant, discutant et donnant des conférences. Il est aussi membre du conseil d'administration de l'Internet Society.

source: [www.oreilly.fr](http://www.oreilly.fr)

## Linus Torvalds



Linus Benedict Torvalds est à l'origine du noyau Linux et coordonne aujourd'hui son développement. Inspiré par le système Minix développé par Andrew S. Tanenbaum, il eut l'idée de créer un système d'exploitation qui puisse fonctionner sur son ordinateur personnel. Torvalds a développé les premières versions du noyau sur son temps libre et sur le matériel qu'il possédait alors, un PC de la série 386 durant l'année 1991.

Linus Torvalds est né à Helsinki, en Finlande, le 28 décembre 1969. Il fait partie de la communauté des Finlandssvensk, une population représentant 6% des habitants de la Finlande, parlant le suédois. Il a étudié à l'université d'Helsinki. Il vit actuellement à Santa Clara en Californie (USA) avec sa femme Tove et ses deux filles. Il est employé de l'Open Source Development Lab (OSDL).

Sa mascotte est un manchot, surnommée Tux, largement adoptée par la communauté des linuxiens.

La biographie complète sur [http://fr.wikipedia.org/wiki/Linus\\_Torvalds](http://fr.wikipedia.org/wiki/Linus_Torvalds)

## Tristan Nitot (fr)



Tristan Nitot a publié son premier site web public en 1996, avec une parodie des pages perso de l'époque racontant la vie et l'oeuvre de Virtual Bob, son fils, qui ne devait naître que quelques mois plus tard.

Passionné par Internet depuis les premières versions de Mosaic et de Navigator, de Netscape Communications, il a rejoint cette entreprise début 1997 pour s'occuper successivement des relations techniques avec la presse puis le marketing produit au niveau France et Europe du sud. Il est maintenant en charge des relations avec les développeurs Web pour AOL et Netscape pour l'Europe. Tristan a compris l'intérêt des standards du Web quand il a ouvert avec un éditeur de texte une page qu'il avait composé quelques mois plus tôt avec un éditeur WYSIWYG. Sous le choc, il a fait vœu de chasteté (pour le code HTML/CSS uniquement) et se refuse à utiliser la mise en page en tableaux et toute balise propriétaire.

Contributeur au navigateur Open-Source Mozilla, il tente désespérément d'occuper son temps libre en maintenant le StandBlog, un site sur les standards du W3C, et en élevant deux enfants, Robin et Philippine.

Source: <http://www.nitot.com/>

## Nat Makarevitch (fr)



Nat Makarevitch est né en 1967.

Acteur et promoteur du logiciel libre dans le monde francophone depuis 1991 : co-fondateur du forum [fr.comp.os.linux](http://fr.comp.os.linux), fondateur de [linux-france.org](http://linux-france.org) et d'[ikarios.com](http://ikarios.com)...

Consultant indépendant puis éditeur (éditions O'Reilly).

Co-fonde en 2000 la SSII 'IDEALX', spécialiste du logiciel open source. Y exerce la fonction de directeur technique et R&D.

Connaissances techniques en matière de système comme de développement.

Sa biographie est consultable sur : <http://makarevitch.org/>.

## Wine

Wine est l'acronyme récursif de « Wine Is Not an Emulator » (ou l'acronyme de WINDows Emulator...) et tous les détails

### Histoire

Le développement de Wine a commencé autour de 1993 par Bob Amstadt qui utilisait GNU/Linux et MS-Windows sur la même machine. Les logiciels GNU/Linux étaient arrivés à un point permettant de satisfaire de nombreux besoins mais il avait certains jeux dont il était fou qui n'étaient disponibles que sous MS-Windows.

Fatigué de redémarrer uniquement pour jouer, il commença à travailler sur le moyen d'intercepter les appels système des jeux utilisés pour les faire correspondre à ceux de l'environnement X sous GNU/Linux.

D'autres entendirent parler de ce travail et vinrent à la rescousse, jusqu'à ce que Wine soit capable d'exécuter les jeux auxquels ils souhaitaient jouer.

Autour de 1995, certains tentèrent d'exécuter d'autres programmes, dont Quicken et la suite MSOffice et ceux-ci devinrent le centre d'intérêt principal. Un groupe distinct fit sécession, continuant le support des techniques graphiques complexes utilisées dans les jeux mais rarement nécessaires dans les applications bureautique. Le projet prit alors une approche plus formalisée, avec une équipe de développeurs et un chef de projet. Depuis 2000; le projet a été repris de manière plus systématique avec un chef de projet et une équipe support basés aux États-Unis, deux petites équipes de développement au Canada et des développeurs dans la plupart des pays européens. Des éditeurs majeurs contribuent aussi, IBM par exemple.

Des techniques d'identification des appels systèmes par les programmes furent créées. Dans de nombreux cas, le développement d'une petite quantité de code permettait le fonctionnement d'une application. Il fut trouvé que les programmes préparent souvent l'appel à une interface particulière sans réaliser l'appel lui-même. Le code fut alors écrit pour permettre aux programmes de continuer à faire ces appels préparatoires sans déclencher d'erreur intermédiaire, ainsi que le code des appels eux-mêmes.

La première utilisation commerciale de Wine fut réalisée par Corel qui réalisa un gros travail de support de Wine et l'utilisa pour produire une version native GNU/Linux de WordPerfect 8. D'autres entreprises ont depuis utilisé Wine pour produire une version GNU/Linux de leurs produits avec le minimum d'efforts et de modifications, l'une des dernières étant Xilinx qui réalise des paquetages de CAO spécialisée en électronique. Le projet Ximian Mono prévoit d'utiliser Wine pour permettre aux applications .NET écrites pour MS-Windows de fonctionner sans réécriture (voir <http://appde.winehq.com/> pour le détail du niveau de support de diverses applications).

Récemment, une équipe de développeurs MS-Windows expérimentés a commencé la réalisation d'une suite de programmes de test systématique des plus de 12 000 appels système actuellement présents dans l'ensemble des bibliothèques MS-Windows.

Actuellement, Wine se compose d'environ 750 000 lignes de code « C » implantant autour de 90% des appels des spécifications courantes de MS-Windows telles que ECMA-234 et Open32. Les appels non publiquement documentés sont plus difficiles à implanter mais des progrès sont en cours.

Certaines entreprises qui travaillent sur Wine développent du code pour des fonctions particulières initialement propriétaires. Elles font cela pour se fondre elles-mêmes ainsi que leur travail dans le flux principal du projet. Elles insèrent leur code dans ce flux lorsqu'elles disposent d'une source de revenus alternative suffisante. Les supports d'OLE et de ActiveX entrent dans cette catégorie.

### Ce que Wine fait

Wine intercepte tous les appels système MS-Windows et MS-DOS ainsi que les interruptions BIOS 4 pour tenter de les faire correspondre dans l'environnement X GNU/Linux. Les instructions natives du processeur sont exécutées comme elles l'auraient été dans l'environnement MS-Windows et c'est pourquoi MS-Wine n'est pas un émulateur intégral.

Wine n'est pas lié à l'architecture Intel x86 - il existe par exemple une version pour Alpha de DEC/Compaq, mais le besoin et l'utilisation sérieuse n'existent que sur x86. Il ne permet pas aux programmes MS-Windows x86 de s'exécuter sur une autre architecture telle que PowerPC ni

SPARC, quoique Wine puisse se compiler et s'exécuter sur les deux.

Toutes les interfaces de l'environnement MS-Windows ne peuvent avoir de correspondance dans l'environnement X GNU/Linux. Certaines n'ont simplement pas d'équivalent. Cela veut dire que dans certains cas une quantité significative de code doit être écrite pour permettre la correspondance. Par exemple, il y a des problèmes avec les curseurs les plus complexes utilisés par certains programmes MSWindows.

Le système X-Window ne peut pas gérer plus de deux couleurs dans un curseur, ce qui oblige Wine à faire des hypothèses sur les couleurs à utiliser, parfois avec un résultat inutilisable. Wine est en réalité composé de deux produits ; Wine lui-même qui permet l'exécution de programmes MS-Windows et Winelib une bibliothèque de compilation destinée à la production de programmes natifs GNU/Linux (c'est celle-ci que Corel a utilisé pour la version GNU/Linux de WordPerfect). Winelib peut être utilisée pour exécuter sur un matériel non-x86 des programmes dont le code source est disponible, quoique des problèmes spécifiques existent encore pour d'autres architectures (notamment concernant l'ordre des octets).

### Où Wine est bon

Le support des programmes MS-Windows 3.x/95/98/Me/NT est disponible (moins complet pour MSWindows NT). De nombreux programmes pour MS-Windows 2000 fonctionneront, sauf s'ils utilisent de nouvelles interfaces spécialisées introduites par ce dernier. Peu de travail a été fait pour le support des programmes spécifiques MS-Windows XP mais il en existe très peu.

Wine supporte l'essentiel des interfaces documentées de MS-Windows, cependant pas toujours de manière aussi complète que l'on pourrait le souhaiter. Voir <http://www.winehq.com/?page=status> pour connaître l'état actuel du support dans Wine.

Les programmes qui s'exécutent en isolation ou utilisent uniquement des interfaces de communication externes doivent fonctionner. Chaque programme doit être testé individuellement car l'interaction des interfaces et paramètres utilisés peut poser problème.

Il a été rapporté l'utilisation avec succès de certains compilateurs et environnements de développements.

### Où Wine n'est pas bon

Certaines zones spécifiques sont incomplètes, l'échange dynamique de données (DDE) par exemple ; cependant, de nombreux programmes font des appels DDE sans utiliser ceux-ci en réalité et ainsi fonctionnent très bien. OpenGL et autres logiciels graphiques à hautes performances ont aussi des problèmes. L'implantation des listes de contrôle d'accès (ACL - comme dans MS-Windows NT) existe en partie mais n'est pas encore intégrée avec les ACL du système d'exploitation sous-jacent. La technologie des pilotes VxD, introduite par MS-Windows 98, est une zone difficile. Elle nécessite l'accès au matériel et à l'intérieur du noyau d'une manière qu'aucun système multi-utilisateur sérieux ne peut permettre. Des techniques existent pour produire des résultats équivalents mais elles impliquent une quantité de travail et leur fonctionnement n'est pas garanti. Dans certains cas, les éditeurs peuvent être convaincus de produire une version GNU/Linux utilisant les interfaces normales. Ce type d'accès étant abandonné par Microsoft (les architectures de type MS-Windows NT ne le permettent pas), cela cessera progressivement d'être un problème.

Certains programmes MS-Windows tentent de manipuler directement les périphériques (notamment les ports série). Cela n'est pas autorisé sous GNU/Linux ni sous Unix. Cela ne s'applique en principe qu'aux paquetages de communication tels que Procomm ainsi qu'aux programmes issus du monde MS-DOS pour lequel il était nécessaire de procéder ainsi.

Le rendu de certaines images graphiques n'est pas encore satisfaisant (en particulier celui des polices TrueType) mais ce but est activement poursuivi.

L'autre zone de difficulté est le logiciel produit par Microsoft lui-même, car celui-ci tend à utiliser des interfaces non documentées. Quoiqu'il soit possible de découvrir ce qui se passe, les développeurs doivent être prudents car les lois sur la rétro-ingénierie sont très strictes dans certains pays. Les États-Unis par exemple interdisent celle-ci dans tous les cas et de nombreux pays occidentaux ne l'autorisent que pour établir une compatibilité. Ainsi, le travail sur cette partie restera toujours assez lent.

L'exécution de logiciels d'installation, en particulier, a été problématique mais de récents travaux ont résolu l'essentiel des difficultés et le travail continue. Certaines de celles-ci sont causées par



les développeurs de paquetages qui n'appliquent pas les techniques recommandées. L'accès à la base de registre en est un exemple. Wine maintient sa base de registre dans un format différent de celui de MSWindows pour en faciliter la récupération. Tant que les interfaces documentées sont utilisées, cela n'a pas d'importance, mais parfois, les développeurs tentent d'y accéder directement, au risque de corrompre une base de registre réelle sous MS-Windows, avec pour résultat un fonctionnement impossible sous Wine.

Wine est parfois critiqué pour ses faibles performances mais cela est souvent dû à la quantité de code de déverminage. Il est possible de compiler Wine sans celui-ci mais cela doit être réalisé avec prudence car alors les éventuels problèmes ne pourront être diagnostiqués sans une nouvelle recompilation.

### **Wine - alternatives commerciales**

Comme mentionné plus haut, des versions étendues de Wine sont disponibles comme produits commerciaux afin d'aider au développement du tronc principal de celui-ci. Les deux entreprises qui font cela sont Transgaming et CodeWeavers. Transgaming travaille essentiellement sur l'amélioration des interfaces graphiques et sonores et son produit est destiné au marché du jeu. CodeWeavers travaille sur les applications bureautiques principales et édite un produit, CrossOver Office qui supporte MS-Office et Lotus Notes.

### **Wine et Visual Basic**

Il a été rapporté un fonctionnement de Visual Basic 3 mais aucun détail n'est disponible. Visual Basic 6 ne s'installe actuellement pas. Le travail est en cours pour régler cela, mais il est trop tôt pour dire si le résultat sera un succès complet ou non.

Aucune autre version n'a été testée.

### **Migration d'application vers Wine**

Voici une liste des lignes directrices de gestion du processus de migration d'applications sous GNU/Linux avec Wine :

1. Contrôler les conditions de licence : certaines entreprises ont réalisé des licences qui interdisent l'exécution de leur application hors du système d'exploitation cible. Oracle par exemple faisait cela et Microsoft commence à le faire pour les composants qui peuvent être téléchargés gratuitement ; faire une liste à part des programmes qui entrent dans cette catégorie ;
2. Obtenir une copie de chaque application à migrer ; les licences site peuvent ne pas permettre les copies pour tests ;
3. Configurer une machine avec la dernière version de Wine ;
4. Tester chaque programme de la liste de test, noter tous les problèmes rencontrés ainsi que la phase de survenance (installation, initialisation ou exécution) et valider leur impact sur une sélection représentative d'utilisateurs finals ; noter aussi les problèmes de performances ;
5. Pour chaque programme de la liste de problèmes, contrôler tout d'abord l'existence éventuelle d'une version GNU/Linux et tester alors celle-ci aussitôt que possible ; dans le cas contraire, contacter l'éditeur pour lui suggérer d'en réaliser une, par exemple avec Winlib (il faudra négocier séparément avec chaque éditeur) ;
6. Si l'éditeur refuse de coopérer, il faudra trouver une application alternative ou abandonner le projet ;
7. Utiliser la liste des DLL et appels système manquants nécessaires pour en déterminer un coût d'implantation ; tester à nouveau chaque programme avec les dernières versions de Wine/Winlib jusqu'à disparition de tous les problèmes (parfois, les correctifs posent des problèmes à des programmes qui fonctionnaient correctement) ;
8. Wine est normalement compilé avec la trace de déverminage, ce qui impacte négativement les performances (en particulier pour l'interaction écran) ; tester à nouveau tous les programmes qui fonctionnent avec des problèmes de performances sur une version de Wine compilée sans la trace de déverminage ; si les performances restent insuffisantes, un travail de développement sera nécessaire.