KDE

Published : 2017-06-26
License : GPLv2+

KDE DU POINT DE VUE D'UN DÉVELOPPEUR

1. AVEZ-VOUS BESOIN DE CE LIVRE ?
2. LA PHILOSOPHIE DE KDE
3. COMMENT OBTENIR DE L'AIDE

1. AVEZ-VOUS BESOIN DE CE LIVRE ?

Vous devriez lire ce livre si vous voulez développer pour KDE. Nous utilisons le terme *développement* très largement pour couvrir tout ce qui peut conduire à un changement dans le code source, ce qui inclut :

- Soumettre une correction de bogue
- Écrire une nouvelle application optimisée par la technologie KDE
- Contribuer à un projet existant
- Ajouter de la fonctionnalité aux bibliothèques de développement de KDE

Dans ce livre, nous vous livrerons les bases dont vous avez besoin pour être un développeur productif. Nous décrirons les outils que vous devrez installer, montrer comment lire la documentation (et écrire la vôtre propre, une fois que vous aurez créé la nouvelle fonctionnalité !) et comment obtenir de l'aide par d'autres moyens. Nous vous présenterons la communauté KDE, qui est essentielle pour comprendre KDE parce que nous sommes un projet « open source », libre (gratuit).

Les utilisateurs finaux du logiciel n'ont **PAS** besoin de ce livre ! Cependant, ils pourraient le trouver intéressant pour les aider à comprendre comment les logiciels complexes et riches en fonctionnalités qu'ils utilisent ont vu le jour.

2. LA PHILOSOPHIE DE KDE

Le succès de KDE repose sur une vue globale, que nous avons trouvée à la fois pratique et motivante. Les éléments de cette philosophie de développement comprennent :

L'utilisation des outils disponibles plutôt que de ré-inventer ceux existants : beaucoup des bases dont vous avez besoin pour travailler font déjà partie de KDE, comme les bibliothèques principales ou les « Kparts », et sont tout à fait au point. Donc vérifiez toutes les ressources disponibles avant de tenter de résoudre votre propre problème.

Lorsque vous proposez une suggestion, changez *« nous devrions.. »* en « *je veux..* » : les plans grandioses sont inutiles, sauf si vous êtes disposé à vous mettre au travail pour les faire se produire. Vous trouverez de l'aide dès l'instant où vous commencerez !

Améliorez le code de manière itérative : ne laissez pas le parfait être l'ennemi du bien. Essayez une petite solution, faites-la fonctionner et améliorez-la par le biais de tests et de remaniements du code (en anglais, « refactoring ») pour produire un excellent correctif.

LA COMMUNAUTÉ KDE

La plate-forme de développement de KDE est créée et maintenue par une équipe internationale qui coopère au développement et à la distribution de logiciels « open source » libres, pour les ordinateurs de bureau et les portables. Notre communauté a développé une grande variété d'applications pour la communication, le travail, l'éducation et le divertissement. Nous mettons fortement l'accent sur la recherche de solutions aux problèmes anciens et nouveaux, en créant une atmosphère ouverte pour l'expérimentation.

En tant que communauté, nous apprécions toutes les contributions et pouvons utiliser les talents de tous les membres, dont la création artistique, le graphisme, la conception, la communication, les traductions, la documentation, les tests, les rapports de bogues et la « chasse » aux bogues, l'administration du système, sans oublier le codage.

QU'EST-CE QUI REND KDE SI ATTRAYANT ?

La meilleure chose à propos de KDE est notre formidable communauté ! Nous accueillons de nouveaux membres, offrant de l'aide et permettant à de nombreuses personnes d'expérimenter, apprendre et évoluer. Ce livre représente une partie de cette mission.

Nos produits sont utilisés par des millions de personnes travaillant à domicile ou dans un bureau et sont en cours de déploiement dans les écoles du monde entier. Le Brésil a lui seul compte plus de 50 millions d'enfants scolarisés utilisant des logiciels reposant sur KDE pour naviguer, apprendre et communiquer ! En complément du « Google Summer of Code », nous lançons notre propre « Saison de KDE », où les participants prennent la responsabilité de travailler sur un projet de façon intensive pendant l'été et obtiennent un tee-shirt sympathique à la fin, en remerciement. En 2011, nous avons eu 100 propositions. Quelle étonnante communauté !

EST-IL DIFFICILE DE S'IMPLIQUER ?

Pas du tout ! Chaque jour, de plus en plus de personnes rejoignent notre famille toujours croissante de contributeurs. KDE dispose d'une solide infrastructure de ressources web, forums, listes de diffusion, IRC et autres services de communication. Nous pouvons fournir des retours utilisateurs sur votre code et d'autres contributions dans le but de vous aider à apprendre. Naturellement, une attitude proactive et une personnalité conviviale aide sans conteste.

LE CODE DE CONDUITE DE KDE

Lorsque vous communiquez avec les canaux officiels de KDE, veuillez observer le Code de Conduite de KDE.

Notre Code de Conduite présente un résumé des valeurs partagées et des principes de *bon sens* dans notre communauté. Les ingrédients sociaux de base qui cimentent notre projet comprennent :

- Être bienveillant
- Être respectueux
- Être collaboratif
- Être pragmatique
- Soutenir les autres dans la communauté
- Obtenir un soutien des autres dans la communauté

Le Code de Conduite se trouve ici : *http://www.kde.org/code-of-conduct/*

LA VALEUR D'ÊTRE LIBRE (GRATUIT) ET « OPEN SOURCE »

Notre code peut être copié, modifié et redistribué par n'importe qui. Cela signifie que vous pouvez être sûr que le logiciel continuera à être disponible, même si les développeurs originels passent à autre chose. Cela veut dire également que le développement est transparent, avec les commentaires de quiconque sur la planète est intéressé. Enfin, le code « open source » facilite, pour un développeur, la recherche et la résolution des problèmes. Un des effets pratiques des logiciels libres est que de merveilleuses communautés tendent à se développer autour du code. Nous avons créé tout cela au sein de KDE.

La communauté KDE est impliquée dans le mouvement du Libre et de l'« Open Source » plus largement. Nous coopérons avec la communauté GNOME dans FreeDesktop.org et le « Desktop Summit » bi-annuel. Nous sommes un participant leader du « Google Summer of Code » et du « Googe Code-In » depuis leurs débuts. En outre, nous concourons régulièrement avec les membres d'autres groupes FOSS (Free Open Source Software) et participons aux efforts concernant les standards ouverts. Nous sommes soutenus dans ces tâches par la fondation KDE e.V., qui publie des rapports trimestriels d'activité sur *http://ev.kde.org/reports/*.

3. COMMENT OBTENIR DE L'AIDE

Naturellement, nous espérons que ce livre vous sera utile ! Mais inévitablement, vous serez amené à vous poser des questions ou aurez besoin d'aide pour un problème. Comme la communauté KDE est efficace et conviviale, choisissez la méthode la plus appropriée dans ce chapitre et demandez des informations. À moins que vous n'ayez un besoin spécifique de communiquer avec une personne particulière, veuillez vous adresser à l'ensemble de la liste ou du canal.

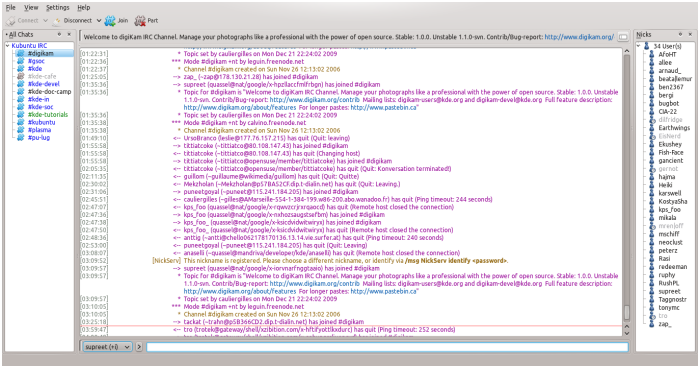
LISTES DE DIFFUSION DE KDE

Les listes de diffusion de KDE sont l'un des principaux canaux de communication de la communauté KDE. Tous les développeurs constateront que la liste KDE-devel est utile. De plus, ceux qui travaillent sur le cœur de la « KDE Software Compilation » (SC) seront amenés à s'abonner à KDE-Core-devel. Ceux qui travaillent sur des applications ou des projets KDE devront s'abonner à la liste de diffusion du projet. L'ensemble des listes de KDE se trouve sur https://mail.kde.org/mailman/listinfo.

De l'aide ainsi que des informations sont disponibles sur les listes. Cependant, si vous avez besoin d'aide rapidement, l'IRC peut s'avérer plus utile.

KDE SUR IRC

IRC ou « Internet Relay Chat » est un outil de communication texte seul et en temps réel. Il existe une variété de clients d'IRC disponibles dans KDE, tels que Konversation, Quassel, et Kvirc.



Presque tous les développeurs KDE font acte de présence ou sont inactifs sur les canaux IRC de KDE sur Freenode (irc.freenode.net). L'IRC est la meilleure façon d'obtenir de l'aide rapide par les développeurs KDE. En tant que développeur, vous serez amené à vous connecter à #kde, #kde-devel, ainsi qu'au canal ou aux canaux de votre projet. Vous constaterez que plus vous passez de temps sur IRC, plus vous arriverez à connaître vos collègues développeurs et nos utilisateurs KDE. Des amitiés durables ont vu le jour grâce aux canaux de KDE. De l'aide est disponible sur les services tels que « chanserv » et « nickserv » grâce aux commandes « /msg chanserv aide » et « /msg nickserv ». Des informations complémentaires sur l'utilisation de Freenode sont disponibles ici : http://freenode.net/using_the_network.shtml.

La « Userbase » conserve une liste des canaux actuels sur http://userbase.kde.org/IRC_Channels, De plus, vous pouvez également utiliser la commande IRC « command */msg alis list $searchterm* », où « *$searchterm* » est le sujet qui vous intéresse.

Lorsque vous posez des questions, gardez à l'esprit ces astuces :

- Si vous avez une question, il suffit de la poser. Il n'est nullement nécessaire de demander d'abord si vous pouvez ou non d'en poser une.
- Soyez préparé à attendre une réponse. Même si l'IRC est un mode plus en « temps réel » de communication que les listes de diffusion, il se peut que personne ne soit disponible pour répondre à votre question immédiatement. En général, si vous ne recevez pas de réponse sur IRC au bout d'environ une heure, il vaut mieux envoyer un courriel.
- Ne posez pas votre question plus d'une fois. Même si le canal est actif, la personne qu'il vous faut n'est peut-être pas disponible pour fournir une réponse. La seule exception à cette règle est, si on vous dit d'attendre qu'une certaine personne se connecte, d'effectuer une nouvelle demande quand vous voyez qu'elle est connectée. Encore une fois, si vous ne recevez pas de réponse à votre question dans l'intervalle d'une heure, il est préférable d'envoyer un courriel.
- Coller de grandes quantités de texte étant considéré contraire à la bonne étiquette, utilisez un « pastebin ». Le « pastebin » de KDE se trouve sur http://paste.kde.org

PROBLÈMES DE LA COMMUNAUTÉ KDE

Si vous rencontrez de mauvais comportements sur une liste ou un canal, veuillez contacter les propriétaires de la liste ou l'un des opérateurs. L'adresse du propriétaire de la liste est « *$listname-owner@kde.org* », où « *$listname* » est le nom de la liste. Dans un canal, les opérateurs peuvent être identifiés en exécutant cette commande : « */msg chanserv access #nom_du_canal liste* ». Il figure un signe « + » à côté des pseudonymes des opérateurs. Pour une aide générale sur la communauté KDE, veuillez écrire au Groupe de Travail communautaire à « *community-wg@kde.org* », ou arrêtez-vous sur notre canal IRC à « #kde-cwg ».

PREMIERS PAS

4. CHOISIR UN PROJET
5. CHOISIR UN PROJET
6. L'INFRASTRUCTURE QT
7. UTILISER GIT POUR LE DÉVELOPPEMENT KDE
8. CHOISIR UN EDI
9. LES API DE KDE
10. DOCUMENTATION

4. CHOISIR UN PROJET

Quand vous venez vers KDE en tant que développeur, il se peut que vous ayez déjà un projet favori et que vous sachiez comment vous souhaitez contribuer. Cependant, il vaut la peine d'examiner les différents projets énumérés dans le présent chapitre, pour découvrir toutes les façons dont vous pouvez être à même d'aider. Et même si vous n'êtes réellement intéressé que par un seul projet, il est utile de savoir ce que font d'autres personnes actives, car votre travail peut interagir avec eux.

INFRASTRUCTURES

Il s'agit des composants généraux à la base des applications et d'autres parties visibles de KDE. L'équipe travaille beaucoup pour rendre les bibliothèques modulaires, clarifier les dépendances, simplifier et accroître la qualité ainsi que la stabilité.

Bibliothèques du cœur de KDE (kdelibs)

Fonctions cruciales nécessaires dans l'ensemble de la plate-forme KDE

Composants graphiques & Classes

Composants graphiques et classes ne faisant pas partie de « kdelibs » mais extrêmement utiles

Serveur mandataire du service web de D-BUS

Un projet pour connecter des services web à l'infrastructure de notification D-BUS sous Linux

kdesu

Outils pour obtenir les privilèges de superutilisateur (*root*) sur différents moteurs

WebKit pour KDE

Un projet intégrant le moteur de navigateur QtWebKit dans la « KDE Software Compilation »

KioFuse

Ce composant insère des ressources KIO (modules d'entrée / sortie de KDE). Ce sont des fichiers distants, archivés ou comprimés dans la hiérarchie du système de fichiers racine.

Nepomuk

Opérations de base sur le bureau, pour l'annotation, l'indexation, la recherche et l'édition de liens

Gestion du réseau

Une applet et un outil de configuration pour la gestion de réseau Solid Networking et KNetworkManager

Oxygen

Création artistique pour la KDE SC

Solid

La bibliothèque matérielle pour la KDE SC

Telepathy

Une infrastructure de communications pour la messagerie instantanée, la VoIP et la collaboration

Akonadi

Un service de stockage extensible à l'échelle du bureau pour les données PIM et les méta-données

Projets connexes

Projets non centraux liés à la technologie KDE de différentes façons par divers moyens, tels que les dépendances ou les outils de construction

PROGRAMMES

Voici un petit découpage des applications créées et maintenues par les développeurs de KDE.

Amarok

Le slogan d'Amarok est *Redécouvrez votre musique*, et son développement s'articule autour cette idéologie. Les fonctionnalités principales d'Amarok telles que le *navigateur de contexte*, la recherche intégrée dans Wikipédia et le téléchargement des paroles aident les utilisateurs à trouver de nouvelles musiques et à en apprendre plus sur la musique dont ils disposent.

digiKam

Logiciel de gestion de photos

Gwenview

Afficheur d'images pour KDE

K3b

Un graveur de disques optiques

KDevelop4

Environnement de développement intégré (EDI) pour la KDE SC

KWin

Le gestionnaire de fenêtres de KDE

Marble

Un programme de globe terrestre visuellement esthétique

Okular

Un afficheur de documents unifié

Rekonq

Un navigateur web léger optimisé par WebKit et la KDE SC

Paramètres système

L'outil de configuration des paramètres système

SuperKaramba

Un outil pour créer des composants graphiques attractifs et d'autres éléments interactifs

SUITES

Ces projets regroupent de nombreuses applications connexes. Le concept clé est que les données créées dans une partie de la suite peuvent être facilement utilisées par une autre application dans la suite.

Projet KDE Éducation

Logiciels éducatifs pour KDE

KDE Finance

Applications financières

Jeux pour KDE

Jeux pour le bureau KDE

Kdetoys

Un ensemble de distractions amusantes

Utilitaires KDE

Une diversité d'outils à exécuter sur le bureau

KOffice et Calligra

Suites bureautiques reposant sur les bibliothèques KDE

PIM

Outils de gestion d'informations personnelles (Personal Information Management)

Plasma

Programmes pour la création rapide et facile de composants graphiques, y compris des lanceurs d'applications interactifs, des fenêtres et des gestionnaires de tâches

PLATES-FORMES

Ces projets s'assurent que KDE fonctionne sur divers systèmes d'exploitation.

KDE sous Mac OS X

Bibliothèques et applications KDE pour Mac OS X

KDE sous Windows

Bibliothèques et applications KDE pour Microsoft Windows

KDE sous FreeBSD

Bibliothèques et applications KDE pour FreeBSD et d'autres versions de BSD

Plasma Active

Un projet pour porter la technologie KDE sur les périphériques mobiles

TRAVAILLER AVEC L'ORGANISATION

Ces projets se consacrent aux personnes et aux processus qui rendent KDE possible.

Équipe de publication de KDE

Programme et coordonne les publications

Projet de documentation de KDE

Crée et maintient la documentation de KDE

kde.org

Fournit des informations autour des sites web *.kde.org

Promotion de KDE

S'occupe de la promotion de KDE et organise des conférences

Programme partenaire

Soutient les fabricants de logiciels indépendants (Independent Software Vendors, ISVs) partenaires de KDE

Projet d'ergonomie de KDE

Applique des principes d'ergonomie et les met en pratique dans l'environnement de bureau KDE

Projet d'accessibilité de KDE

Effectue la construction sur les fonctionnalités de Qt pour rendre les environnements interactifs plus accessibles pour les handicapés ou d'autres personnes ayant des besoins spéciaux

BugSquad de KDE

Conserve la trace des bogues entrants dans les logiciels KDE et suit les anciens bogues

Projets « Summer of Code »

Projets « Google Summer of Code » liés à KDE

« English Breakfast Network » (EBN)

Fournit des outils dédiés à la qualité du code KDE. dont la validation de la documentation de l'API de KDE, la validation de la documentation utilisateur, la vérification du code source, etc.

Recherche KDE

Soutient toute personne intéressée à contribuer aux projets de recherche (fondée) avec et au sein de la communauté KDE.

Si vous êtes encore dans le doute au sujet d'un projet sur lequel vous souhaitez travailler, alors essayez de vous mettre en relation avec les développeurs de la KDE SC sur IRC pour vous familiariser avec le projet.

5. CHOISIR UN PROJET

Lorsque vous venez vers KDE en tant que développeur, il se peut que vous ayez déjà un projet favori et que vous sachiez comment vous voulez contribuer. Cependant il vaut la peine d'examiner les différents projets répertoriés dans ce chapitre pour découvrir toutes les façons dont vous pouvez être en mesure d'aider. Et même si vous n'êtes réellement intéressé que par un seul projet, il est utile de savoir ce que font d'autres personnes actives, car votre travail peut interagir avec elles.

INFRASTRUCTURES

Il s'agit des composants généraux à la base des applications et d'autres parties visibles de KDE. L'équipe travaille dur pour rendre modulaires les bibliothèques, clarifier les dépendances, simplifier et accroître la qualité ainsi que la stabilité.

Bibliothèques principales de KDE (kdelibs)

Fonctions cruciales nécessaires dans l'ensemble de la plate-forme KDE

Composants graphiques & Classes

Composants graphiques et classes ne faisant pas partie de « kdelibs » mais extrêmement utiles

Serveur mandataire du service web de D-Bus

Un projet pour connecter des services web à l'infrastructure de notification de D-BUS sous Linux

kdesu

Outils pour obtenir les privilèges de superutilisateur (*root*) sur différents moteurs

WebKit pour KDE

Un projet intégrant le moteur de navigateur QtWebKit dans la KDE Software Compilation (KDE SC)

KioFuse

Ce composant insère des ressources KIO (modules d'E / S de KDE) tels que des fichiers distants, archivés ou comprimés dans la hiérarchie du système de fichiers racine.

Nepomuk

Opérations de base sur le bureau, concernant l'annotation, l'indexation, la recherche et l'édition de liens

Gestion du réseau

Une applet et un outil de configuration pour la gestion de réseau Solid Networking et KNetworkManager

Oxygen

Création graphique pour la KDE SC

Solid

La bibliothèque matérielle de la KDE SC

Telepathy

Une infrastructure de communications pour la messagerie instantanée, la VoIP et la collaboration

Akonadi

Un service de stockage extensible à l'ensemble du bureau pour les données et les méta-données de PIM

Projets connexes

Projets non centraux liés à la technologie KDE de différentes façons, telles que les dépendances ou les outils de construction

PROGRAMMES

Voici un petit découpage des applications créées et maintenues par les développeurs de KDE.

Amarok

Le slogan d'Amarok est *Redécouvrez votre musique*, et son développement s'articule autour de cette idéologie. Les fonctionnalités principales d'Amarok telles que le *navigateur de contexte* unique, la recherche intégrée sur Wikipédia et le téléchargement des paroles de morceaux de musique aident l'utilisateur à trouver de nouvelles musiques et à en apprendre davantage sur la musique dont ils disposent.

digiKam

Logiciel de gestion de photos

Gwenview

Afficheur d'images pour KDE

K3b

Un graveur pour les lecteurs optiques

KDevelop4

Environnement de développement intégré ou EDI (IDE, *Integrated Development Environment*) pour la KDE SC

KWin

Le gestionnaire de fenêtres de KDE

Marble

Un programme de globe terrestre visuellement esthétique

Okular

Un afficheur de documents unifié

rekonq

Un navigateur web léger optimisé par WebKit et la KDE SC

Paramètres système

L'outil de configuration des paramètres système

SuperKaramba

Un outil pour créer des composants graphiques attrayants et d'autres éléments interactifs

SUITES

Ces projets regroupent de nombreuses applications connexes. Le concept clé est que les données créées dans une partie de la suite peuvent facilement être utilisées par une autre application dans la suite.

Projet « KDE Education »

Logiciels éducatifs pour KDE

KDE Finance

Applications financières

Jeux pour KDE

Jeux pour le bureau KDE

Kdetoys

Un ensemble de distractions amusantes

Utilitaires KDE

Une diversité d'outils à utiliser sur le bureau

KOffice et Calligra

Suites bureautiques reposant sur les bibliothèques KDE

PIM

Outils de gestion d'informations personnelles (PIM, *Personal Information Management*)

Plasma

Programmes conçus pour la création rapide et facile de composants graphiques, dont des lanceurs d'applications interactifs, des fenêtres, ainsi que des gestionnaires de tâches et de fenêtres

PLATES-FORMES

Ces projets s'assurent que KDE fonctionne sur divers systèmes d'exploitation.

KDE sous Mac OS X

Bibliothèques et applications KDE pour Mac OS X

KDE sous Windows

Bibliothèques et applications KDE pour Microsoft Windows

KDE on FreeBSD

Bibliothèques et applications KDE sous FreeBSD et d'autres versions de BSD

Plasma Active

Un projet pour porter la technologie KDE vers les périphériques mobiles

TRAVAILLER AVEC L'ORGANISATION

Ces projets se consacrent aux personnes et aux processus qui rendent KDE possible.

Équipe de publication de KDE

Planifie et coordonne les publications

Projet de documentation de KDE

Crée et maintient la documentation de KDE

kde.org

Fournit des informations concernant les les sites web « *.kde.org »

KDE Promotion

Assure la promotion de KDE et organise des conférences

Programmes partenaire

Soutient les fabricants de logiciels indépendants partenaires de KDE (ISV, *Independent Software Vendors*)

Projet d'ergonomie de KDE

Applique les principes et les pratiques d'ergonomie à l'environnement KDE (*K Desktop Environment*)

Projet d'accessibilité de KDE

Effectue la construction sur les fonctionnalités de Qt pour rendre les environnements interactifs plus accessibles aux handicapés ou à d'autres personnes ayant des besoins spéciaux

KDE BugSquad

Conserve la trace des bogues entrants dans les logiciels KDE et explore les anciens bogues

Projets « Summer of Code »

Projets « Summer of Code » de Google liés à KDE

English Breakfast Network (EBN)

Fournit des outils dédiés à la qualité du code KDE, dont la validation de la documentation de l'API de KDE, la validation de la documentation de l'utilisateur, la vérification du code source, etc.

KDE Research

Soutient toute personne intéressée à à contribuer aux projets de recherche (financés) avec et au sein de la communauté KDE.

Si vous êtes encore dans le doute au sujet du projet sur lequel vous souhaitez travailler, alors essayez de vous mettre en relation avec les développeurs de la KDE SC sur IRC pour vous familiariser avec ce projet.

6. L'INFRASTRUCTURE QT

Pour commencer à développer sur la plate-forme de développement de KDE, vous devrez vous familiariser avec l'infrastructure Qt, une des clés de voûte du développement de KDE.

Qt (prononcé officiellement « kju:t ») est une infrastructure d'applications multi plate-fome fondée sur C++, largement utilisée pour développer des logiciels applicatifs avec une interface graphique utilisateur (GUI, Graphical User Interface). Ainsi, il constitue en grande partie une boîte à outils de composants graphiques, mais il sert également à développer des programmes sans interface graphique, tels que des outils en ligne de commande et des terminaux pour les serveurs.

Outre la plate-forme de développement de KDE, Qt est plus particulièrement employé dans Autodesk Maya, Adobe Photoshop Elements, OPIE, Skype, le lecteur multimédia VLC, VirtualBox et Mathematica, ainsi que par l'Agence Spatiale Européenne, DreamWorks, Google, HP, Lucasfilm, Panasonic, Philips, Samsung, Siemens, Volvo et les studios d'animation de Walt Disney.

AVANTAGES DE QT

Écrire du code une seule fois dans des plates-formes multiples

Qt permet d'écrire des applications avancées et des interfaces utilisateur que vous pouvez déployer au travers de différents postes de travail et systèmes d'exploitation embarqués, sans réécrire le code source, en économisant du temps et du coût de développement.

Créer des expériences utilisateur étonnantes

Que vous préfériez C++ ou JavaScript, Qt fournit les éléments de construction pour des interfaces modernes et interactives : un vaste ensemble de composants graphiques personnalisables, des canevas graphiques, des moteurs de style et bien plus. Vous pouvez incorporer des graphiques 3D, du multimédia (audio ou vidéo), des effets visuels et des animations pour distinguer votre application en dehors de la compétition.

En faire plus (et plus vite) avec moins

Qt est rapide à apprendre et à utiliser, en particulier lorsqu'on l'emploie en association avec le nouvel environnement multi plate-forme Qt Creator. Et la bibliothèque de classes modulaires de Qt fournit une grande partie partie de l'infrastructure nécessaire pour les applications interactives.

Mélanger du code web et du code natif dans une seule application

L'intégration de Qt avec le moteur de rendu web WebKit signifie que vous pouvez incorporer rapidement du contenu et des services provenant du Web dans votre application native et que vous pouvez utiliser l'environnement web pour distribuer vos services et votre fonctionnalité.

Pour apprendre comment utiliser Qt, nous recommandons les tutoriels suivants :

http://doc.qt.nokia.com/

7. UTILISER GIT POUR LE DÉVELOPPEMENT KDE

Git est un système de contrôle de versions gratuit et « open source » conçu pour tout manipuler, depuis de petits projets jusqu'à de très gros projets avec rapidité et efficacité. Il offre de nombreux outils pour savoir où vous en êtes au fur et à mesure lorsque vous modifiez des fichiers, ainsi que pour fusionner vos modifications avec celles effectuées par d'autres développeurs.

Vous pouvez en savoir plus sur git (et le télécharger si nécessaire) à l'adresse http://git-scm.com.

BASES DE GIT

Il existe plusieurs niveaux auxquels vos modifications peuvent résider dans git. Vous devrez passer par toutes les étapes avec soin afin d'enregistrer les changements correctement. Une séquence typique d'activités de « git » commence par le clonage d'un dépôt distant. À présent, vous disposez d'une copie complète de ce que possède le développeur originel. Puis vous devrez procéder comme suit :

1. Créer et modifier des fichiers.
2. Exécuter « git add » pour avertir « git » à propos de tous nouveaux fichiers que vous avez créés ou de fichiers que vous venez de modifier. Les fichiers comportant des changements se trouvent dans un état intermédiaire appelé « zone de transit » (*staging area*), mais pas encore dans un quelconque dépôt.
3. Exécuter « git commit » pour enregistrer vos changements dans votre propre dépôt local

Nous présenterons des exemples de tous ces aspects plus loin dans ce chapitre.

Au lieu d'avoir à vous rappeler et à saisir les différentes adresses de « git » complètes pour rapatrier des données depuis le serveur (*pulling*) et les envoyer vers le serveur (*pushing*), nous vous recommandons d'ajouter manuellement les lignes suivantes à votre configuration utilisateur de Git (*~/.gitconfig*) :

```
[url "git://anongit.kde.org/"]
    insteadOf = kde:
[url "git@git.kde.org:"]
    pushInsteadOf = kde:
```

CLONER DES DÉPÔTS

Après avoir défini votre « *~/.gitconfig* » comme affiché dans la section précédente, vous devez cloner les dépôts source à l'aide de la commande suivante :

```
git clone kde:nom du projet
```

où *nom_du_projet* est le nom du projet auquel vous souhaitez contribuer. Par exemple, pour soumettre un correctif pour « kdelibs », démarrez par la commande :

```
git clone kde:kdelibs
```

RAPATRIER DES CHANGEMENTS

Si vous avez déjà un dépôt « git » que vous devez mettre à jour avec les nouveaux changements depuis le source original, exécutez la commande suivante :

```
git pull
```

TRAVAILLER AVEC LES BRANCHES

Git est un système de contrôle de révisions très puissant qui prend en charge le concept des branches. Afin de développer une nouvelle fonctionnalité pour un projet de la plate-forme de développement de KDE, il est préférable d'utiliser une branche séparée pour le développement d'une fonctionnalité. Vous pouvez extraire une nouvelle branche grâce à la commande suivante :

```
git checkout -b nom de la fonctionnalité
```

par exemple :

```
git checkout -b maFonctionnalité
```

Plus tard, vous pourrez basculer entre des branches multiples à l'aide de la commande d'extraction de git :

```
git checkout maFonctionnalité
```

Chaque projet « git » comprend une branche appelée « maîtresse » (*master*), que les propriétaires du dépôt considèrent comme la branche principale. Habituellement, rien ne vient s'ajouter à cette branche jusqu'à ce qu'elle soit testée dans divers environnements et que les chefs de projets soient sûrs qu'elle est à la fois robuste et utile.

Suivre les branches distantes

Il existe deux types de branches dans « git », *locale*s et *distante*s. Les branches distantes sont celles qui se trouvent dans le dépôt distant. Vous pouvez également suivre les branches distantes via la commande suivante :

```
git checkout --track dépôt distant:branche distante
```

Par exemple, pour extraire la banche 4.7 de KDE, utilisez : git checkout --track origin:KDE/4.7

VALIDER VOTRE TRAVAIL

Avant d'envoyer quoi que ce soit dans la base de code de KDE, vous avez besoin d'une identité et d'un compte de développeur KDE. Veuillez vous connecter sur http://identity.kde.org/ pour enregistrer votre compte. Votre nom de compte sera presque toujours votre nom patronymique ; ne tentez pas de contourner cette règle. Les développeurs très expérimentés peuvent obtenir des comptes comportant des droits de validation (*commit*), mais ce point est en dehors de la portée de ce livre.

Afin de générer un « diff » (affichage des différences) de vos changements et de les réviser, lancez la commande suivante :

```
git diff
```

Vous pouvez rediriger cette sortie dans un correctif à l'aide de :

```
git diff > mon correctif.patch
```

Pour pouvoir valider votre travail, il vous faut d'abord ajouter les fichiers que vous avez modifiés dans la zone de transit grâce à la commande suivante :

```
git add nomdeFichier1 nomdeFichier2
```

Afin de valider vos changements dans votre dépôt local, utilisez la commande suivante :

```
git commit
```

Cette opération ouvre un éditeur de texte où vous pouvez saisir un message de validation décrivant vos modifications.

SOUMETTRE VOS CHANGEMENTS DANS L'ARBRE PRINCIPAL

Vous pouvez soumettre vos correctifs au « Comité de Révision de KDE » sur https://git.reviewboard.kde.org/. Connectez-vous à l'aide de votre compte d'identité KDE et proposez-y un correctif relatif à un projet.

Il est également possible d'envoyer directement des correctifs au comité de révision via une révision postérieure. La procédure est exposée sur :

> http://community.kde.org/Sysadmin/GitKdeOrgManual#Using_Review_Board_and_post-review

DÉPANNAGE

Pour tout problème lié à « git » et à la plate-forme de développement de KDE, vous pouvez chercher de l'aide sur l'un des canaux suivants sur irc.freenode.net.

- #kde-sysadmin
- #kde-git

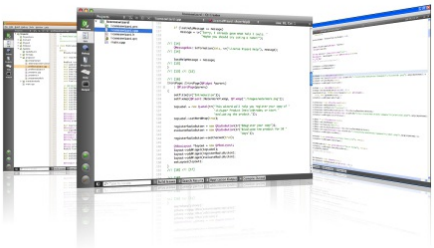
Vous pouvez en savoir plus sur la plate-forme de développement de KDE à l'aide de « git » sur http://techbase.kde.org/Development/Git.

8. CHOISIR UN EDI

Un environnement de développement intégré (EDI) permet de réaliser la gestion d'un projet, des tests et d'autres activités de manière pratique tout au long de votre codage. Nous vous recommandons d'installer un des EDI suivants et de faire en sorte que votre développement de KDE fonctionne avec.

Nous préconisons QtCreator pour sa facilité d'utilisation et ses fonctionnalités, en particulier son éditeur de texte intégré. Mais il est bon de savoir, si vous connaissez déjà Eclipse, que vous pouvez l'utiliser pour le développement de KDE aussi.

QTCREATOR

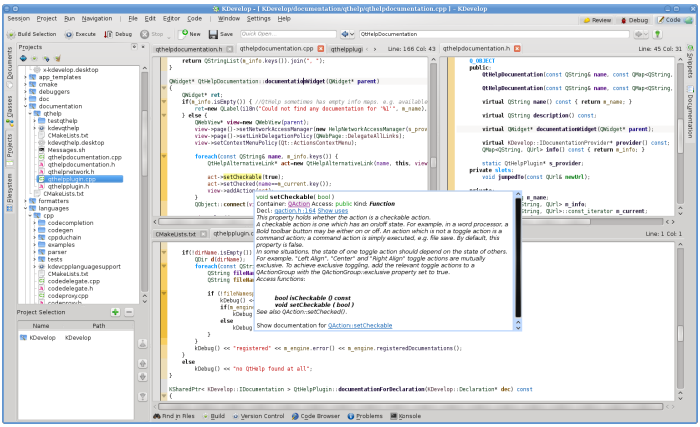


QtCreator est un EDI intégré, multi plate-forme pour C++ et JavaScript qui fait partie du « Qt SDK ». Il comprend un débogueur visuel et un outil de conception pour la disposition des interfaces graphiques et les formulaires. Les fonctionnalités de l'éditeur incluent la coloration syntaxique et le complètement automatique. QtCreator utilise le compilateur C++ GNU et les outils apparentés. Sous Windows, il peut employer MinGW ou MSVC avec l'installation par défaut et « cdb » lorsqu'il est compilé depuis le source.

Vous pourrez en savoir plus sur l'utilisation de QtCreator avec la plate-forme de développement de KDE sur :

http://techbase.kde.org/Development/Tutorials/Using_Qt_Creator

KDEVELOP



KDevelop est un EDI (en anglais, *Integrated Development Environment*) open source, gratuit, créé en tant que partie du projet de développement KDE pour prendre en charge le développement en C/C++ et d'autres langages sous Microsoft Windows, Mac OS X, Linux, Solaris et FreeBSD. Il est très complet et peut être étendu grâce à des modules externes. Il repose sur KDevPlatform et la plate-forme de développement KDE ainsi que les bibliothèques Qt et il est en développement depuis 1998.

Vous pouvez en savoir plus à propos de KDevelop sur :

http://userbase.kde.org/KDevelop4/Manual

ECLIPSE

Eclipse est un EDI « open source » très connu, utilisé principalement pour programmer en Java, mais il prend en charge un certain nombre d'autres langages, dont le C++, avec les modules complémentaires appropriés.

Vous pourrez en savoir plus sur l'utilisation de Eclipse avec la plate-forme de développement de KDE sur :

http://techbase.kde.org/Development/Tools/Eclipse

9. LES API DE KDE

Une Interface de programmation d'application (API, *Application Programming Interface*) est un ensemble de fonctions exposées par un programme ou une bibliothèque, qui peut être exécutée par d'autres programmeurs. Une API étend considérablement un programme en laissant des développeurs tiers ajouter de la fonctionnalité nouvelle. C'est un moyen idéal pour permettre d'ajouter de nouvelles fonctionnalités sans avoir à modifier le code de base existant.

La documentation d'une API explique comment le système fonctionne et quelles méthodes on peut appeler. Pour obtenir quelques tutoriels conviviaux, commencez par utiliser la myriade d'API de la plate-forme de développement KDE, consultez :

http://techbase.kde.org/Development/Tutorials

La documentation de l'APi de développement de KDE (apidox) se trouve sur :

http://api.kde.org

Cette documentation est livrée au format « qch », un format HTML étendu qu'on peut afficher soit dans l'Assistant de QtCreator soit par le biais de n'importe quel navigateur.

Les « apidox » sont traitées par l'outil de documentation Doxygen, lequel lit le code source d'une application (ou d'une bibliothèque) et produit à partir de là de la documentation élégamment mise en forme. Il existe un bon manuel de référence disponible, mais avec un peu de chance, vous n'aurez pas à le consulter simplement pour la construction de la documentation de base. Lorsque vous installez la plate-forme de développement de KDE à des fins de développement, vous devrez également installer Doxygen. Ensuite vous pourrez créer la documentation depuis n'importe quel projet à l'aide de la commande suivante :

```
make docs
```

Vous trouverez plus d'informations sur les documentations de l'API de KDE sur :

http://techbase.kde.org/Development/Tools/apidox

10. DOCUMENTATION

Si vous écrivez de nouvelles fonctions pour un projet KDE, nous espérons certainement que vous rédigerez de la documentation (apidox) pour celles-ci et essaierez d'expliquer aussi clairement que possible ce qu'on peut faire avec les fonctions. Mieux vous documenterez votre travail, plus les développeurs seront susceptibles de l'utiliser et moins ils risqueront de vous ennuyer en posant des questions de base sur son fonctionnement.

La documentation de l'API est parfois appelée « Manuel de référence des API ». Ce ne doit pas être un simple manuel de référence, cependant. Elle peut contenir quantité d'éléments additionnels, tels que tutoriels, des exemples et des informations historiques. Cette page fait référence à tout le matériau qui documente et explique l'API d'un morceau de code en tant que « apidox », le terme utilisé dans la documentation de la plate-forme de développement de KDE elle-même.

Écrire des « apidox » de base est amusant et simple : ajoutez des commentaires spécialement mis en forme dans votre code, expliquant ce que les choses sont censées être. Ces commentaires étant presque impossibles à distinguer de ce que vous écririez dans les en-têtes de votre code de toute façon, ce n'est pas difficile à faire. Voici un échantillon.

```
/**
 * @author praxagora
 *
 * Retourne un χ² comparant deux échantillons.
 * Les tableaux doivent être de la même taille.
 *
 * @param a Premier échantillon
 * @param b Second échantillon
 *
 * @return @size taille de chaque échantillon.
 */
double chi_square(double a[], double b[], int size) {
double total_a=0.0, total_b=0.0, total, totals[size],
square_a[size], square_b[size], square_totals[size],
    terms[size],
    terms_total=0.0, sum_total=0.0, square_sum_total=0.0;
  int i;
  /**
    * Cette boucle comprend la fonction complète et calcule
    * le χ² à partir des deux tableaux.
  */
    {
  for (i = 0; i < size ; i++)
```

Notez qu'un commentaire a été inclus avant la fonction, et avant l'accolade qui débute le niveau d'imbrication suivant, une boucle « for ». Tous les paramètres, ainsi que la valeur de retour, sont marqués.

Pour écrire une bonne « apidox », il faut d'abord des connaissances techniques : vous devez comprendre le code que vous documentez -- ou au moins savoir ce qu'il est censé faire et comment il est destiné à être utilisé. L'autre partie d'une bonne « apidox » relève de la discipline pure : les « apidox » sont les plus utiles lorsqu'elles sont exhaustives.

En fait, la documentation n'a habituellement pas à expliquer ce que fait le code à chaque étape. Le code devra être si clair, avec une disposition propre et des noms bien choisis pour les méthodes et les variables, qu'il s'auto-documentera. À la place, la documentation devra expliquer *à quoi* le code est destiné, quand et pourquoi il est appelé, les objectifs et les intervalles de ses arguments, voire peut-être l'algorithme utilisé sans oublier les compromis consentis en termes de temps et de mémoire.

Consultez la documentation de Qt pour vous faire une impression de ce à quoi doivent ressembler de bonnes « apidox ». Elles présentent une cohérence de style et sont imprégnées d'un souci d'exhaustivité. Vous pouvez en apprendre beaucoup sur Qt juste en lisant la documentation. Vous n'avez pas nécessairement besoin d'exécuter les programmes tutoriels ou de lire le code source de la bibliothèque pour découvrir ce que fait un paramètre ou un drapeau dans une certaine méthode de la bibliothèque. Elle vous est entièrement expliquée dans le détail.

ENVIRONNEMENT DE CONSTRUCTION DE DÉVELOPPEMENT DE KDE

11. PRÉ-REQUIS
12. AUTRES FAÇONS DE CONSTRUIRE LA KDE SC
13. CONSTRUCTIONS DE KDE SCRIPTÉES (KDESRC-BUILD)
14. DÉPANNER VOTRE CONSTRUCTION KDE

11. PRÉ-REQUIS

Avant de réellement créer l'environnement de développement de KDE depuis le source, réfléchissez -- avez-vous besoin de le faire ? Oui, si vous travaillez sur le cœur de la KDE SC. Vous n'avez pas besoin de tout l'environnement construit depuis le source si vous projetez de travailler sur une seule application. Nous recommandons de confirmer si vous devez ou non de construire l'environnement en postant une question sur la liste de diffusion ou en discutant sur l'IRC.

En supposant que vous deviez construire KDE depuis le source, vous aurez à configurer votre environnement de compilation et installer les paquets contenant les fichiers d'en-tête utilisés pendant la compilation des sources de la plate-forme KDE.

CONFIGURER L'ENVIRONNEMENT

Afin de configurer l'environnement de construction de la plate-forme KDE, il faut créer un fichier définissant les variables d'environnement ainsi que d'autres paramètres en suivant les instructions figurant sur :

http://techbase.kde.org/Getting_Started/Build/Environment#Environment_Configuration

Enregistrez le fichier dans le dossier de votre environnement de construction sous le nom « .build-config ». Si vous avez besoin d'environnements de construction séparés -- comme un pour les versions stables et un pour les compilations réalisées la nuit -- il est recommandé de créer un script séparé pour chaque environnement de construction et de le placer dans le dossier racine de l'environnement.

EXIGENCES EN MATIÈRE DE CONSTRUCTION

Cette section détaille les exigences de logiciels que vous devez installer sur votre système avant de pouvoir commencer à construire la plate-forme KDE. Pour la plupart de ces exigences, il vaut mieux utiliser les paquets fournis par la distribution de votre système d'exploitation, mais dans certains cas, il vous faudra construire ces exigences vous-même.

Debian / Ubuntu

Voici quelles sont les dépendances de construction minimales sous Debian / Ubuntu pour KDE 4.6 et postérieur :

```
sudo apt-get install graphviz libxml2-utils libopenexr-dev libjasper-
dev libenchant-dev \
libavahi-common-dev libaspell-dev libasound2-dev libldap2-dev
libsasl2-dev \
libsmbclient-dev libxkbfile-dev libxcb1-dev libxklavier-dev
libxdamage-dev \
libxcomposite-dev libbluetooth-dev libusb-dev network-manager-dev \
libsensors4-dev libnm-util-dev libcfitsio3-dev libnova-dev libeigen2-
dev \
libopenbabel-dev libfacile-ocaml-dev libboost-python-dev libsvn-dev
libsvncpp-dev \
libqt4-dev libqca2-dev libstreamanalyzer-dev libstrigiqtdbusclient-dev
\
libcommoncpp2-dev libidn11 libidn11-dev libpci-dev libxss-dev libxft-
dev \
libpolkit-agent-1-dev libpolkit-backend-1-dev libpolkit-gobject-1-dev
git libpoppler-qt4-dev \
libspectre-dev
```

Les paquets supplémentaires et optionnels pour Debian / Ubuntu se trouvent sur :

> **http://techbase.kde.org/Getting_Started/Build/Distributions/Debian**

OpenSUSE

La construction minimale pour OpenSuse dépend de ce qui suit :

```
 yast -i alsa-devel automoc4 avahi-devel patch cups-devel kde4-
filesystem \ libbz2-devel avahi-compat-mDNSResponder-devel hal-devel
xorg-x11-devel \ libQtWebKit-devel libxml2-devel kdesdk4 clucene-core-
devel boost-devel \ libjpeg-devel liblrdf-devel libpng-devel libxslt-
devel libredland-devel \ Mesa-devel giflib-devel subversion gcc-c++
gmp-devel xine-devel \ libgpgme-devel pcre-devel dbus-1-devel libqt4-
devel cmake git \  doxygen polkit-devel docbook-xsl-stylesheets cyrus-
sasl-devel libical-devel
```

Les paquets supplémentaires et optionnels pour OpenSUSE se trouvent sur :

http://techbase.kde.org/Getting_Started/Build/Distributions/openSUSE

Autres distributions

Veuillez vous référer à
http://techbase.kde.org/Getting_Started/Build/Distributions

12. AUTRES FAÇONS DE CONSTRUIRE LA KDE SC

Vous n'avez pas toujours à construire la KDE SC depuis des sources minimales. De ombreuses constructions pré-conditionnées de la KDE SC sont disponibles. Ce chapitre va mettre l'accent sur l'installation de la KDE SC depuis le dépôt « git » maître sans la compiler depuis les sources.

PROJET NEON

La merveilleuse équipe du projet Neon fournit des constructions quotidiennes de la KDE SC pour les développeurs utilisant Kubuntu. Le projet Neon installe les binaires (exécutables) et les bibliothèques dans */opt/project-neon/,* ce qui vous donne une installation de type « bac à sable » de la KDE SC principale à côté de votre installation KDE SC stable.

Extrayez le projet Neon sur la page d'accueil Launchpad : https://launchpad.net/~neon. Vous pouvez installer les paquets à l'aide de la commande suivante :

```
sudo add-apt-repository ppa:neon/ppa && sudo apt-get update && sudo
apt-get install project-neon-base
```

Vous trouverez plus d'informations sur le projet Neon sur :

https://wiki.ubuntu.com/Kubuntu/ProjectNeon

http://techbase.kde.org/Getting_Started/Using_Project_Neon_to_contribute_to_KDE

OPENSUSE

Cette distribution GNU / Linux fournit de fréquents instantanés de la KDE SC principale. La finalité consiste à fournir des instantanés toutes les semaines, mais ce n'est pas toujours possible. Il se peut que ces paquets n'incluent pas toujours des correctifs spécifiques à OpenSUSE et ne soient pas disponibles pour les versions d'OpenSUSE plus anciennes. Vous devez ajouter le dépôt des paquets principaux pour utiliser les paquets « Extra » ou « Unstable:Playground ».

Version : 11.4

Paquets principaux :
http://download.opensuse.org/repositories/KDE:/Unstable:/SC/openSUSE_11.4/

Applications publiées :
http://download.opensuse.org/repositories/KDE:/UpdatedApps/openSUSE_11.4/

Extra

Applications supplémentaires :
http://download.opensuse.org/repositories/KDE:/Extra/openSUSE_11.4_KDE_Unstable_SC/

Unstable:Playground :
http://download.opensuse.org/repositories/KDE:/Unstable:/Playground/openSUSE_11.4_KDE_Unstable_SC/

Version: 11.3

Paquets principaux :
http://download.opensuse.org/repositories/KDE:/Unstable:/SC/openSUSE_11.3/

Applications publiées :
http://download.opensuse.org/repositories/KDE:/UpdatedApps/openSUSE_11.3/

Extra :
http://download.opensuse.org/repositories/KDE:/Extra/openSUSE_11.3_KDE_Unstable_SC/

Unstable:Playground :
http://download.opensuse.org/repositories/KDE:/Unstable:/Playground/openSUSE_11.3_KDE_Unstable_SC/

Version : Factory

Paquets principaux :
http://download.opensuse.org/repositories/KDE:/Unstable:/SC/openSUSE_Factory/

Applications publiées : Use the application packages from
openSUSE:Factory

Extra :
http://download.opensuse.org/repositories/KDE:/Extra/openSUSE_Factory_KDE_Unstable_SC/

Unstable:Playground :
http://download.opensuse.org/repositories/KDE:/Unstable:/Playground/openSUSE_Factory_KDE_Unstable_SC/

Vous trouverez plus d'informations sur les constructions principales
KDE d'OpenSUSE sur :

http://en.opensuse.org/KDE_repositories#Unstable:SC_aka._KUSC_.28KDE_trunk.29

13. CONSTRUCTIONS DE KDE SCRIPTÉES (*KDESRC-BUILD*)

La façon la plus simple de construire la KDE SC (*Software Compilation*) *ex nihilo* consiste à exécuter un script nommé « kdesrc-build » (anciennement « kdesvn-build »), écrit par Michael Pyne. Cette approche est fortement recommandée pour les novices en matière de construction de la KDE SC parce qu'il s'occupe de presque tout le processus à votre place. Les constructions restant compatibles avec les méthodes manuelles de construction de la KDE SC, vous pourrez changer plus tard si vous le souhaitez les modules que vous installez.

kdesrc-build automatise les tâches suivantes et bien plus :

- Effectuer l'extraction initiale
- Gérer les mises à jour des modules déjà extraits
- Configurer le système de construction du module
- Effectuer la construction et l'installation
- Spécifier vos options CMake ou configurer des drapeaux (donc vous n'aurez pas à vous en souvenir à chaque fois)
- Journaliser les erreurs de construction, ainsi vous pouvez les réviser plus facilement pour vous dépanner

Ce n'est pas de la panacée pour régler tous vos problèmes lors de la construction de la KDE SC. Un dépannage peut encore être requis. Nombre d'erreurs qui surviennent en utilisant d'autres méthodes se produiront ici aussi : vous devrez lire les fichiers journaux qui sont mémorisés à votre intention.

LIENS UTILES POUR CHERCHER DES MODULES

Lorsque vous utilisez *kdesrc-build*, il se peut que vous trouviez utile de chercher des modules dans le dépôt source. Les sources incluent ce qui suit :

- Pour parcourir n'importe lequel des divers projets de la KDE SC à l'aide de « git », vous pouvez utiliser https://projects.kde.org/ ou http://gitweb.kde.org/.
- Pour parcourir le dépôt « Subversion » de KDE, utilisez http://websvn.kde.org/trunk/KDE/

CONFIGURATION

Ces sous-sections couvrent tout ce dont vous avez besoin pour obtenir et installer *kdesrc-build* lui-même.

Pré-requis

kdesrc-build est vraiment facile à installer et à configurer, mais il faut que vous ayez les logiciels appropriés installés pour construire la KDE SC. Les pré-requis pour construire la KDE SC sont disponibles sur :

http://techbase.kde.org/Getting_Started/Build/Requirements

kdesrc-build requiert « Perl 5.8 » ou supérieur, installé par défaut avec la plupart des distributions et inclus dans le lien précédent. Vérifiez votre version de Perl avec :

```
perl -v
```

Vous avez également besoin de « libwww », une collection de modules liés à Internet, disponibles sur https://github.com/gisle/libwww-perl.

Télécharger et installer « kdesrc-build »

Une fois que votre système est configuré et en mesure de compiler la KDE SC, vous pouvez télécharger *kdesrc-build* sur son site web, http://kdesrc-build.kde.org. Le fichier que vous téléchargez contiendra (au moins) le script *kdesrc-build* et un fichier échantillon de configuration. Installer *kdesrc-build* est aussi simple qu'enregistrer le fichier et le rendre exécutable. Si vous le souhaitez, vous pouvez le déplacer dans un dossier dans votre « PATH ». Cependant pour cet exemple, nous le placerons dans le dossier source de la KDE SC que nous utilisons (*~/kdesrc*).

```
mkdir -p ~/kdesrc &&
cd ~/kdesrc &&
tar xjvf ~/path/to/kdesrc-build-1.12.tar.bz2 &&
cp kdesrc-build-1.12/kdesrc-build .
```

Sinon, il est possible de rapatrier directement le script *kdesrc-build* le plus récent (ainsi que le fichier échantillon de configuration) à l'aide de « git » :

```
git clone git://anongit.kde.org/kdesrc-build.git ~/kdesrc
```

MISE EN PLACE DE LA CONFIGURATION

Les options de configuration de *kdesrc-build* sont tirées d'un fichier nommé *~/.kdesrc-buildrc*. Les directives pour le modifier se trouvent sur :

http://techbase.kde.org/Getting_Started/Build/kdesrc-buildrc

Pour la majeure partie, les réglages par défaut dans le fichier « kdesrc-buildrc-sample » inclus devraient être suffisants. Vous pouvez le copier dans votre dossier personnel comme suit :

```
cp ~/kdesrc/kdesrc-build-1.12/kdesrc-buildrc-sample ~/.kdesrc-buildrc
# Maintenant modifiez le ~/.kdesrc-buildrc
```

Notez que le nom du fichier de configuration commence par un point placé en son début (.), ce qui en fait un fichier caché. Si vous utilisez Dolphin ou Konqueror depuis le bureau, vous risquez de devoir afficher les fichiers cachés afin de trouver le fichier de configuration et le modifier. Autrement, vous pouvez modifier le fichier échantillon avant de le copier dans *~/.kdesrc-buildrc*.

De plus, veillez à ce que les modules que vous voulez construire sont inclus. Vous aurez besoin au moins de ceux-ci :

- qt-copy
- kdesupport
- kdelibs
- kdepimlibs
- kdebase

Les modules étant construits dans l'ordre où ils apparaissent dans votre fichier *~/.kdesrc-buildrc*, le premier module devra être « qt-copy ». Le module « kdesupport » devra se trouver avant « kdelibs », lequel devrait se trouver avant n'importe quel autre module de la KDE SC, et ainsi de suite.

Le fichier échantillon de configuration n'inclut pas ces modules par défaut. Ainsi, vous n'aurez pas besoin d'effectuer beaucoup de changements, à moins que vous ne souhaitiez ajouter certains modules à la construction en les décommentant.

Si un module que vous souhaiteriez construire n'est pas déjà présent, ajoutez simplement les lignes suivantes à la fin du *~/.kdesrc-buildrc* :

```
module nom-module
end module
```

« *nom-module* » représente n'importe quel nom du module dans le dépôt de logiciels (par exemple, kdemultimedia).

Vous pouvez être amené à cocher l'option « make-install-prefix » si vous installez la KDE SC ou Qt dans un dossier ne se trouvant pas dans votre dossier personnel. Grâce à « make-install-prefix », comme vous pouvez exécuter *su* ou *sudo* pendant le processus « make install », vous pouvez installer des fichiers en tant que superutilisateur (*root*) ou définir certains programmes à exécuter avec des droits d'accès plus importants (requis pour certains programmes à lancer correctement).

```
module kdelibs
  make-install-prefix sudo -S # sudo sans stdin
end module

module kdebase
  make-install-prefix sudo -S
end module
```

Modules reposant sur « Git »

La plupart des modules de la KDE SC se trouve dans « git », bien que quelques-uns d'entre eux doivent encore être portés de « svn » vers « git ». Finalement, tous les modules de la KDE SC et les projets seront contenus dans « git ».

Pour construire ces modules dans *kdesrc-build*, il suffit d'ajouter quelques lignes à la configuration du module. Par exemple, konversation est développé dans le dépôt « Git » sur :

https://projects.kde.org/projects/extragear/network/konversation/repository

Ainsi, pour ajouter ce module, écrivez ce qui suit à la fin de *~/.kdesrc-buildrc* :

```
module konversation
    repository git://anongit.kde.org/konversation
    branch master
end module
```

Dans ce cas, j'ai sélectionné la branche maîtresse (« master ») car il s'agit de la branche « git » par défaut.

Maintenant, à chaque fois que vous construirez « konversation », *kdesrc-build* utilisera « git » au lieu de Subversion.

COMMANDES DE « KDESRC-BUILD » UTILES

kdesrc-build étant piloté depuis la ligne de commande, voici un guide pour certaines des options en ligne de commande les plus utiles :

Option	Effet
--pretend (-p)	Cette option est comme un coup d'essai. *kdesrc-build* traitera les options et sa configuration normalement et s'exécutera via la construction comme d'habitude, mais au lieu de télécharger ou d'exécuter la construction, elle se contentera d'afficher les messages sur ce qu'elle aurait fait. Vous devriez toujours exécuter avec « -p » avant de lancer le script en réel, pour être sûr qu'elle effectue ce que vous attendez.
--nosvn (--no-src)	Cette option ignore l'étape de mise à jour du code source. Elle est utile si vous exécutez *kdesrc-build* à nouveau peu de temps après une précédente mise à jour et ne voulez pas attendre juste pour découvrir qu'il n'existe pas de changements.
--refresh-build	Cette option amène *kdesrc-build* à supprimer les informations de construction actuelles pour les modules indiqués sur la ligne de commande et les construire à nouveau à partir de rien. Cette option ajoute beaucoup de temps mais offre la meilleure chance d'une construction réussie.

Tous arguments autres qu'une option sur la ligne de commande sont supposés être des modules à construire (et sont construits dans l'ordre prévu sur la ligne de commande). Si aucun module n'est spécifié, la totalité des modules listés dans le *~/.kdesrc-buildrc* sont construits dans l'ordre répertorié dans le fichier.

CONSTRUIRE LA KDE SC

Nous y sommes presque. Si vous êtes satisfait de vos paramètres, il est temps de tester *kdesrc-build*. En théorie, les choses sont aussi simples qu'exécuter *kdesrc-build* et y revenir ultérieurement.

```
cd ~/kdesrc
./kdesrc-build
```

Vous pouvez être amené à le tester en construisant simplement « qt-copy » d'abord, cependant.

```
cd ~/kdesrc
./kdesrc-build qt-copy
```

Si la construction a échoué (*kdesrc-build* sortira une erreur avec un message d'erreur surligné en rouge), il existe plusieurs éléments à vérifier :

1. Il se peut qu'il vous manque un logiciel essentiel requis (tel qu'une bibliothèque de développement)
2. Le code de la KDE SC en cours de compilation risque d'être cassé d'une certaine façon, donc il ne fonctionnera pas. Ce problème est couramment dû à du code récemment validé qui fonctionnait sur la machine du développeur ou occasionnellement, parce qu'on est lundi (lorsque des changements incompatibles sont permis dans « kdelibs »).
3. Il est possible que *~/.kdesrc-buildrc* ne soit pas configuré correctement, que vous tentiez d'installer dans un dossier dans lequel vous n'avez pas l'autorisation d'accéder, par exemple, ou vous avez peut-être spécifié un système « qtdir » qui n'existe pas.
4. Le module dépend peut-être d'une version plus récente de « qt-copy » ou « kdelibs » (ou d'un autre module). Dans ce cas, vous devrez exécuter *kdesrc-build* pour mettre à jour le module obsolète d'abord.

Comment découvrez-vous de quelle erreur il s'agissait ? La sortie de la commande défaillante sera dans le dossier des journaux. Par défaut, toute la sortie du journal se trouve dans le sous-dossier *log* du dossier source de la KDE SC journal. Le dossier des journaux se présente comme suit : *log/date-run/module/output-file.log*. Pour simplifier la recherche du fichier approprié, deux liens symboliques sont créés :

log/latest

> Contient la sortie de débogage depuis la dernière exécution de *kdesrc-build* (« --pretend » ne l'affecte pas).

log/latest/*module*/error.log

> Contient la sortie de débogage concernant la commande qui a causé l'échec de la construction du *module* spécifié.

Par exemple, si la construction de « qt-copy » a simplement échoué, vous pourriez lire la la sortie de la manière suivante :

```
cd ~/kdesrc
kwrite log/latest/qt-copy/error.log
```

Remplacez *kwrite* par votre éditeur préféré. Avec un peu de chance, la sortie peut vous guider dans la résolution du problème. Par exemple, si l'échec est la sortie de CMake indiquant qu'il vous manque une bibliothèque, installez cette bibliothèque et essayez à nouveau. Pour les erreurs de liens, vous pouvez essayer d'exécuter avec l'option « --refresh-build » sur le module (ou, si cela ne fonctionne pas, sur les bibliothèques requises comme « qt-copy » et « kdelibs »).

Si vous êtes déconcerté par l'erreur, il peut être bon d'attendre une journée, essayer de mettre à jour à nouveau et espérer que la raison de l'erreur a été corrigée. Vous pouvez également essayer d'envoyer un message à la liste de diffusion « kde-devel » pour savoir si d'autres personnes connaissent ou non le problème ou ont eu des difficultés similaires.

LANCER VOTRE NOUVELLE INSTALLATION KDE

En supposant que vous ayez obtenu assez de modules à construire et à installer pour disposer d'une installation KDE SC opérationnelle, il vous reste encore à configurer votre environnement correctement pour l'exécuter. *kdesrc-build* ne vous aidera pas ici (encore), donc vous devez suivre les instructions sur :

http://techbase.kde.org/Getting_Started/Using_an_IDE_with_KDE4

Veillez à utiliser les mêmes emplacements que ceux que vous avez définis dans *~/.kdesrc-buildrc*. Pour les variables KDEDIRS et KDEDIR, faites appel au paramètre de l'option « prefix » (dans la section globale). Pour la variable QTDIR, utilisez le réglage de l'option « qtdir ».

MAINTENIR VOTRE INSTALLATION KDE À JOUR

Maintenir votre installation KDE à jour est aussi simple que d'exécuter *kdesrc-build* à nouveau. Chaque construction comporte ces phases :

1. Mettre à jour le code source pour tous les modules en cours de construction.
2. Construire et ensuite installer tous les modules.

Comme les anciens dossiers de construction ne sont pas supprimés par défaut, la construction après une petite mise à jour ne prendra normalement pas autant de temps que la construction initiale d'un module. Cette opération est appelée « make incrémental ». Cependant, il peut s'avérer nécessaire de temps en temps de procéder à une reconstruction complète, en raison d'incohérences entre la configuration du dossier de construction et les changements apportés au dossier source. Vous pouvez utiliser l'option « --refresh-build » pour forcer une reconstruction complète.

Pour plus d'informations sur la manière de tirer profit de *kdesrc-build*, consultez la documentation sur : http:/kdesrc-build.kde.org/documentation/

Le site décrit l'intégralité des options des modules et des options en ligne de commande disponibles pour *kdesrc-build* et indique des astuces sur la façon d'effectuer diverses tâches utiles.

14. DÉPANNER VOTRE CONSTRUCTION KDE

Les erreurs de compilation et d'édition de liens sont des sources fréquentes de découragement. Prenez bonne note de la première occurrence d'une erreur dans votre processus de génération. Ce pourrait être aussi simple qu'une variable d'environnement incorrecte, une version inattendue d'une bibliothèque ou un pré-requis manquant. Veuillez lire les instructions attentivement, vérifiez s'il existe des fautes d'orthographe lors de l'utilisation des noms de modules ainsi que des commandes.

Veuillez réviser vos journaux et effectuer des recherches de correctifs. Si vous ne pouvez pas trouver une solution alors demandez de l'aide sur IRC ou une liste de diffusion.

SUJETS AVANCÉS

15. LIRE LES PILES D'APPELS (BACKTRACES)

15. LIRE LES PILES D'APPELS (*BACKTRACES*)

Une pile d'appels, également appelée *trace d'appels* (stack trace) ou *trace arrière de la pile* (stack traceback) est un rapport de la manière dont le programme a appelé différentes fonctions au cours de son exécution. Il est couramment utilisé pendant le débogage interactif et le débogage *post mortem*. Il peut également être affiché à l'intention de l'utilisateur d'un programme comme faisant partie d'un message d'erreur, que celui-ci peut signaler à un programmeur.

Chaque fonction place un *cadre de piles* (stack frame) sur la pile contenant ses arguments ainsi que d'autres informations dont elle a besoin pour s'exécuter. Les cadres de piles actifs reflètent un certain point dans le temps pendant l'exécution d'un programme. Une trace de piles permet de suivre la séquence des fonctions imbriquées appelées jusqu'au point où la trace d'appels est générée. Dans un scénario *post mortem*, la trace d'appels va jusqu'à la fonction où l'erreur s'est produite et l'inclut. Sachez, cependant, que la fonction où l'erreur est survenue peut ne pas être responsable de l'échec ; une erreur pourrait bien avoir été imbriquée dans une fonction plus importante (par exemple, en passant une valeur incorrecte à la fonction où le programme a échoué.

La figure suivante illustre un cadre de piles où « main() » appelée « hello() », laquelle appelé « hi() », laquelle a appelé « readinput() ». Une trace d'appels est susceptible de descendre depuis le dernier appel vers le premier, pour que « readinput() » puisse apparaître en premier lieu.

Les piles d'appels sont essentielles. Elles peuvent paraître sans signification à vos yeux, mais elles sont capables en fait de contenir une multitude d'informations utiles. Une pile d'appels décrit quelles ont été les fonctions appelées avant la panne, pour que les développeurs puissent repérer dans quelle fonction le dysfonctionnement a commencé. Les adresses mémoire exactes peuvent également aider à localiser les données posant problème, telles qu'un vidage mémoire ou « core dump » (un fichier abandonné lorsqu'un programme échoue, comportant le contenu de la mémoire vive au moment de l'échec). Mais produire des piles d'appels satisfaisantes présente un inconvénient : les bibliothèques et les exécutables occupent beaucoup plus d'espace disque que leurs composants de compteurs optimisés qui ne peuvent pas fournir les informations nécessaires pour produire une pile d'appels.

La boîte de dialogue de panne de KDE (Dr. Konqi) devrait apparaître immédiatement après une panne.

KDE Crash Dialog

Ouvrir l'onglet « Informations sur le développeur » affichera la pile d'appels pertinente. Ce processus pouvant prendre un certain temps et beaucoup de mémoire, l'opération peut soudainement paraitre stagner. Mais voici à quoi devrait ressembler le résultat :

```
Using host libthread_db library "/lib/libthread_db.so.1".
[Thread debugging using libthread_db enabled]
[New Thread -1232783168 (LWP 7604)]
[KCrash handler]
#6  0x0806be76 in TreeMapItem::parent (this=0x0)
    at /home/bram/KDE/kde3/kdeaddons/konq-plugins/fsview/treemap.h:285
#7  0x08065fea in TreeMapItemList::compareItems (this=0xbfec04a8,
item1=0x0,
    item2=0x0)
    at /home/bram/KDE/kde3/kdeaddons/konq-
plugins/fsview/treemap.cpp:720
#8  0xb7281619 in QGList::operator== () from /usr/qt/3/lib/libqt-
mt.so.3
#9  0x0806d498 in QPtrList<TreeMapItem>::operator== (this=0xbfec04a8,
    list=@0xbfec0468) at /usr/qt/3/include/qptrlist.h:74
#10 0x08062e18 in TreeMapWidget::mousePressEvent (this=0xbfec03ac,
    e=0xbfebff1c)
    at /home/bram/KDE/kde3/kdeaddons/konq-
plugins/fsview/treemap.cpp:1840
#11 0xb7004a63 in QWidget::event () from /usr/qt/3/lib/libqt-mt.so.3
#12 0xb6f6bca7 in QApplication::internalNotify ()
   from /usr/qt/3/lib/libqt-mt.so.3
#13 0xb6f6ca88 in QApplication::notify () from /usr/qt/3/lib/libqt-
mt.so.3
#14 0xb7725a84 in KApplication::notify (this=0xbfec055c,
receiver=0xbfec03ac,
    event=0xbfebff1c)
    at /home/bram/KDE/kde3/kdelibs/kdecore/kapplication.cpp:550
#15 0xb6f0bfd2 in QETWidget::translateMouseEvent ()
   from /usr/qt/3/lib/libqt-mt.so.3
#16 0xb6f0b8b0 in QApplication::x11ProcessEvent ()
   from /usr/qt/3/lib/libqt-mt.so.3
#17 0xb6f1b761 in QEventLoop::processEvents () from
/usr/qt/3/lib/libqt-mt.so.3
#18 0xb6f82831 in QEventLoop::enterLoop () from /usr/qt/3/lib/libqt-
mt.so.3
#19 0xb6f826b6 in QEventLoop::exec () from /usr/qt/3/lib/libqt-mt.so.3
#20 0xb6f6b72f in QApplication::exec () from /usr/qt/3/lib/libqt-
mt.so.3
#21 0x0805181e in main (argc=134673960, argv=0xffffffff)
    at /home/bram/KDE/kde3/kdeaddons/konq-plugins/fsview/main.cpp:55
```

Dans cette pile d'appels, le premier cadre de piles est affiché sur la ligne n°6. Du fait que la pile est déroulée depuis la fin jusqu'au début, on peut voir que l'appel qui a arrêté brutalement le programme était « parent() », qui a été appelé par « compareItems() » à la ligne n°7, lequel à son tour a été appelé par l'opérateur surchargé == sur la ligne n°8 et ainsi de suite.

Après le numéro de ligne, le nombre hexadécimal commençant chaque ligne est l'adresse en mémoire où le cadre d'appels démarre pour chaque fonction. À moins que vous n'ayez un vidage mémoire (*core dump*), celui-ci ne vous est d'aucune utilité. Plus intéressantes sont les listes d'arguments et les adresses de leurs données entre parenthèses. Ainsi, la ligne n°6 montre que « parent() » a été appelée avec un argument unique, « this », dont la valeur est 0 (0x0 en hexadécimal). Naturellement, le nom « this » est affectée à l'objet sur lequel la méthode a été exécutée. Ainsi la méthode « parent() » a été effectivement appelée sans arguments. Les méthodes, dans les langages orientés objet, sont passées par le pointeur vers l'objet sur lequel elles ont été exécutées en tant que leur premier argument. Donc, « compareItems() » sur la ligne n°7 a été appelée avec deux arguments, mais parce que « this » a été passée comme premier argument, il en existe trois affichées entre parenthèses.

Sur la ligne n°6, la chaîne « (this=0x0) » indique que la fonction « parent() » est appelée avec un pointeur NULL. Bien sûr, n'importe quel programme « plantera » s'il essaie d'y récupérer des données ou de placer des données dans une adresse à laquelle il n'a pas accès. Comme l'adresse 0x0 sur pratiquement tous les systèmes informatiques est réservée et indisponible pour le programme, vous pouvez indiquer que lire à partir de, ou écrire vers, un pointeur NULL provoquera un « plantage ». On peut également voir, dans la documentation de la fonction Qt « parent() », qu'elle est appelée sans arguments (donc le problème n'était pas causé par un argument incorrect) et retourne un pointeur vers le parent de l'objet sur lequel il est appelé. Par conséquent, le développeur devra essayer de découvrir quel objet « parent() » a été appelé et pourquoi il n'a pas été possible de retourner le parent.

ANNEXES

16. OUTILS UTILES
17. GUIDE DU DÉVELOPPEUR KDE – FOIRE AUX QUESTIONS (FAQ)
18. GLOSSAIRE GÉNÉRAL
19. LIENS UTILES
20. À PROPOS DE CE LIVRE

16. OUTILS UTILES

Ce chapitre liste certains de nos outils favoris pour le développement, le débogage, la construction de logiciels et autres activités. Certains sont évoqués dans d'autres parties du livre également.

OUTILS PRINCIPAUX

Ceux-ci sont absolument requis pour construire et développer des logiciels KDE. Sous Linux, ils sont généralement fournis par votre distribution. Sur d'autres plates-formes, les paquets devraient normalement être disponibles, souvent par téléchargement directement à partir de la page d'accueil de l'outil. Naturellement, les outils de développement standard, tels qu'un compilateur C / C++ et un certain type d'éditeur de texte sont également nécessaires.

CMake - CMake est un système de construction de choix pour la « KDE SC ». Une fois que vous l'avez obtenu, vous pouvez vous en servir pour configurer un projet de logiciel à construire et ce processus vous informera à propos de tout autre pré-requis manquant.

Git - Comme la plupart des projets de la « KDE SC » est développée dans Git, vous en aurez besoin pour obtenir le code source dans le premier emplacement. Vous pouvez trouver les URL de Git pertinentes dans le dossiers des projets de la « KDE SC » pour suivre les instructions de configuration de « git ».

Subversion - La « KDE SC » utilise toujours Subversion pour certains aspects, notamment les traductions.

OUTILS DE DÉBOGAGE ET D'ANALYSE

Ces outils sont utilisés pour analyser un programme et effectuer des tâches telles que le profilage.

Valgrind - Valgrind aide à trouver des fuites de mémoire et des blocs de mémoire non initialisés. Il comprend également un outil de profilage, et bien plus. Valgrind est l'un des outils de développement les plus importants.

Le débogueur du Projet GNU (GDB, *GNU Project Debugger*) - GDB aide à trouver des problèmes dans le code source en permettant de définir des points d'arrêt, parcourir le code, examiner les traces d'appels, etc. Il continuer à évoluer, mais nous recommandons actuellement la version 6.x. Des interfaces graphiques sont disponibles. Consultez le tutoriel de débogage « Déboguer avec GDB » sur http://techbase.kde.org/Development/Tutorials/Debugging/Debugging_with_GDB.

KDbg et DDD - KDbg et DDD sont des interfaces graphiques à GDB.

Outils MS Windows (Process Explorer, Console, WinDbg, DebugView, etc.) - Vous trouverez plus d'informations à leur sujet sur la page « KDE sous Windows » sur Techbase :

http://techbase.kde.org/Projects/KDE_on_Windows

OUTILS DE DÉVELOPPEMENT

Il s'agit d'outils plus simples qui vous aideront au cours du développement. Nous avons abordé certains d'entre eux précédemment.

Qt Creator

Il s'agit de l'EDI recommandé pour le développement de la « KDE SC », facilitant l'écriture de nouveaux programmes et pour intégrer ceux qui existe dans un EDI. Des détails complémentaires se trouvent dans ce tutoriel :

http://www.developer.nokia.com/Community/Wiki/How_to_use_Qt_Creator_IDE

KDevelop

KDevelop est l'EDI créé comme composant de la « KDE SC » pour développer des applications Qt et C++. Il comprend un débogueur intégré, un puissant éditeur doté de la coloration syntaxique, un assistant de projet pour créer des applications depuis des modèles, la suite automake / autoconf, et même la documentation des classes. D'autres détails se trouvent également sur le wiki de KDevelop.

http://kdevelop.org/

Eclipse

Eclipse a été développé pour la programmation Java, mais conçu avec une intention d'extensibilité qui l'a rendu célèbre pour de nombreux langages de programmation. Vous en savoir plus sur http://www.eclipse.org.

EDI MS Visual Studio® Express (Windows uniquement)

Visual C++® Express est la version libre (gratuite) du compilateur Microsoft Visual Studio, et est officiellement prise en charge par Nokia. C'est un choix pour compiler des applications Qt et KDE sous Windows. D'autres informations se trouvent sur :

http://techbase.kde.org/Projects/KDE_on_Windows

OUTILS D'INTERNATIONALISATION (I18N)

Ces outils vous aident à créer des applications susceptibles d'être visualisées et utilisées dans différents langages et avec les conventions locales des différents pays en matière de monnaie, heure, etc.

Lokalize

Lokalize est un système de traduction assistée par ordinateur (CAT, *Computer-Aided Translation*) axé sur la productivité et l'assurance de qualité. Offrant les composants habituels en matière d'outils de TAO, comme une mémoire de traduction, un glossaire, etc, Il comporte également une fonction de fusion (synchronisation) unique des traductions. Lokalize est destiné à la traduction de logiciels et intègre également des outils externes de conversion pour la traduction de documents de bureau en indépendant.

Dr. Klash

Cet outil signale les raccourcis en conflit dans les menus. Il est utile tant pour les traducteurs que pour les développeurs.

Le langage x-test

Ce langage vous aide à trouver des chaînes non traduite dans des applications. Si vous démarrez votre application avec la locale « x-test », toutes les chaînes traduites apparaîtront avec des « xx » au début et à la fin.

OUTILS D'AIDE

Ces outils fournissent une variété de fonctions utiles.

kde4-config

Cet outil aide l'utilisateur à en savoir plus à propos de l'installation de la « KDE SC ».

kfmclient

Cet outil permet de contrôler le navigateur Konqueror par le biais de scripts.

kioclient

Outil en ligne de commande pour les opérations réseau transparentes.

kconf_update

Cet outil met à jour les fichiers de configuration.

Outils apidox

Ces outils aident à développer et générer la documentation de l'API pour votre code.

Automoc4

Il s'agit d'un outil d'aide qui construit la « KDE SC4 ». Il génère automatiquement des fichiers « moc ».

svnmerge.py

Cet outil vous aide à conserver une trace des fusions entre différentes branches SVN.

ASSURANCE DE QUALITÉ

Révision du code

KDE utilise le Comité de Révision pour effectuer les révisions de code ; consultez la page sur le Review Board sur la « KDE Techbase » pour plus d'informations :

http://techbase.kde.org/Development/Review_Board

Construction continue - Tableaux de bord

Le tableau de bord de Dirk comporte tous les modules KDE, alors que le tableau de bord win32 offre une sélection disponible sous Windows.

« English Breakfast Network » - Analyse statique

Le « English Breakfast Network » est une collection de machines qui effectuent un contrôle de qualité automatisé des artéfacts des sources KDE. Pour l'essentiel, cela signifie qu'ils disposent d'un clone de « git » de la base de code complète de KDE (y compris la documentation, etc.) et ils exécutent les outils de vérification dessus.

EBN utilise :

- Krazy - Analyse du code
- APIDOX - Documentation de l'API
- Sanitizer - Vérificateur DocBook
- Contrôles d'ergonomie

17. GUIDE DU DÉVELOPPEUR KDE – FOIRE AUX QUESTIONS (FAQ)

Q. Je voudrais aider / développer la « KDE SC ». Comment puis-je débuter ?

R. Un excellent moyen de se familiariser avec la base de code consiste à corriger des bogues. Vous pourrez trouver un certain nombre de « tâches junior » sur le Bugzilla de KDE. Cherchez le lien sur les « tâches junior » sur le Bugzilla de KDE (http://bugs.kde.org) sur la gauche.

Q. Je ne comprends pas ce que [insérez un terme étrange ici] signifie.

R. Vérifiez le terme dans le *Glossaire général et le Jargon de KDE*.

Q. À quel projet devrais-je contribuer ?

R. Cherchez un projet qui vous intéresse et commencez à coder ! La plupart des personnes commence par « choisir celui qui leur plaît » d'abord, ce qui signifie qu'ils essaient de corriger les bogues qui les irritent le plus. Consultez également les projets de la « KDE SC » dans la section « *Choisir un projet* ».

Q. Comment puis-je configurer l'Environnement de Développement ?

R. Ce point est documenté dans la section « *Environnement de construction du développement de KDE » de ce livre*.

Q. Quels langages de programmation dois-je connaître ?

R. La majeure partie de la « KDE SC » est écrite en C++, mais la « KDE SC » comporte des liaisons pour Python, Ruby, etc. La boîte à outils Qt vous sera très utile dans la plus grande partie de la base de code de la « KDE SC ».

Q. Quels concepts de programmation ai-je besoin de connaître ?

R. Une compréhension de la programmation orientée objet (OOP, *Object-Oriented Programming*) est précieuse si voulez coder sur la plate-forme de développement de KDE, mais n'est pas nécessaire pour développer des applications. Il faut aussi des connaissances sur le contrôle de versions logicielles et les concepts de communications de Qt des signaux et des slots (méthodes de classes).

Q. Comment puis-je lire les piles d'appels ?

R. Ce sujet est abordé dans la section *Lire les piles d'appels* de ce livre.

Q. Comment puis-je utiliser Valgrind ?

R. Veuillez vous référer à http://techbase.kde.org/Development/Tools/Valgrind pour obtenir une brève vue d'ensemble de Valgrind.

Q. Quelle est la meilleure distribution pour le développement de la KDE SC ?

R. Toute distribution fournissant les tout derniers et les meilleurs paquets de la KDE SC convient parfaitement pour la développer.

Q. Comment puis-je signaler un bogue ?

R. Utilisez Bugzilla, notre logiciel de suivi de bogues. Un bon rapport de bogue est exhaustif (contenant tous les détails pouvant être pertinents, tels que le système d'exploitation que vous employez, les versions de tous les logiciels connexes et les actions précises que vous effectuiez lorsque le bogue s'est produit). Il devra également être factuel, poli et écrit clairement. Pour avoir une rapide présentation de Bugzilla, veuillez vous référer à :

http://techbase.kde.org/Contribute/Bugsquad/Quick_Introduction_to_Bugzilla

Pour obtenir plus de Questions fréquemment posées, visitez le site http://techbase.kde.org/Category:FAQs

18. GLOSSAIRE GÉNÉRAL

A

Accessibilité (a11y) : la capacité de toutes les personnes, quels que soient le handicap ou la gravité de l'infirmité à utiliser les fonctionnalités d'un programme

Akonadi : service de stockage extensible à l'ensemble du bureau, destiné aux données et méta-données de gestion des informations personnelles (PIM, *Personal Information Management*), fournissant un accès simultané en lecture, écriture et recherche. Fournit une identification et une extraction des objets à l'échelle du bureau.

Akregator : agrégateur de flux « open source », prenant en charge à la fois RSS et Atom. Les flux peuvent être triés par catégories, et il existe une fonctionnalité de recherche incrémentale pour les titres de toutes les entrées dans la base de données.

Algorithme : procédure pas à pas pour les calculs, le traitement de données et le raisonnement automatisé

Amarok : *Redécouvrez votre musique* grâce à l'application Amarok de KDE. Les principales fonctionnalités d'Amarok telles que le *navigateur de contexte* unique, la recherche intégrée dans Wikipédia et le téléchargement des paroles aident les utilisateurs à trouver de nouvelles musiques et à en apprendre plus sur la musique dont ils disposent.

(API, *Application Programming Interface*) : un ensemble particulier de règles de codage et de spécifications que les logiciels peuvent suivre pour communiquer entre eux. Il sert d'interface entre différents logiciels et facilite leur interaction, de la même façon que l'interface utilisateur facilite l'interaction entre les humains et les ordinateurs.

apidox : documentation de l'API

Applet : programme écrit en Java pour être imbriqué dans un autre environnement, tel qu'une page web.

B

Bogue : une erreur, un défaut, une imperfection, un échec ou une faute dans un programme ou un système informatique qui produit un résultat incorrect ou inattendu, ou encore l'amène à se comporter de manière non intentionnelle.

Boîte à outils (*Toolkit*) : ensemble d'unités de construction pour les interfaces graphiques utilisateur. La KDE SC utilise la boîte à outils Qt.

Bot : applications exécutant automatiquement des tâches sur l'Internet ou sur les canaux IRC.

Bug Squad : l'équipe qui conserve la trace des bogues entrants dans les logiciels KDE.

Bugzilla : un traqueur de bogues web polyvalent et un outil de test utilisé par la communauté KDE.

Bureau : en informatique graphique, un environnement de bureau (DE, *Desktop Environment*) désigne un style d'interface graphique utilisateur (GUI, *Graphical User Interface*) dérivé d'une métaphore du bureau qu'on voit sur la plupart des ordinateurs personnels modernes. Les bureaux Linux les plus appréciés sont les espaces de travail KDE et GNOME.

C

Canal, canal IRC : l'endroit élémentaire pour poser des questions et obtenir de l'aide sur IRC. Il est impoli de diriger des questions sur une seule personne plutôt que de faire une demande sur le canal en général.

Chaîne de compilation *(Toolchain)* : ensemble d'outils de programmation utilisés pour créer un produit (généralement un autre programme informatique ou un système de programmes). Les outils peuvent être employés dans une chaîne, pour que la sortie de chaque outil devienne l'entrée pour le suivant, mais le terme est largement utilisé pour désigner n'importe quel ensemble d'*outils de développement liés*.

CMake : un système de construction « open source » permettant aux développeurs d'automatiser la compilation, les tests et le conditionnement des logiciels reposant sur des spécifications écrites dans des fichiers texte.

Code : texte écrit dans un langage de programmation informatique.

Comité de révision (*Reviewboard*) : outil de révision web de code collaboratif, disponible en tant que logiciel libre sous licence MIT. Une alternative à Rietveld et Gerrit, le Comité de Révision s'intègre avec Bazaar, ClearCase, CVS, Git, Mercurial, Perforce et Subversion.

Compilateur : pour utiliser un compilateur afin de traiter du code source en code exécutable. Également une destination pour les messages journalisés par les programmes ou le système d'exploitation, où les administrateurs ou les développeurs peuvent visualiser les messages.

Composant graphique (*widget*) : élément d'une interface graphique utilisateur (GUI, *Graphical User Interface*) affichant un arrangement des informations modifiable par l'utilisateur, telles qu'une fenêtre ou une zone de texte. La caractéristique de définition d'un composant graphique est de fournir un seul point d'interaction pour la manipulation directe d'un type précis de données. En d'autres termes, les composants graphiques sont des blocs de construction visuels de base qui, combinés dans une application, maintiennent toutes les données traitées par l'application et les interactions disponibles sur ces données.

Console : une interface en ligne de commande (CLI, *Command Line Interface*).

Construction : raccourci pour « construction de logiciels », faisant référence soit au processus de conversion des fichiers de code source en artéfacts logiciels autonomes qu'on peut exécuter sur un ordinateur, soit au résultat de cette opération. Une des étapes les plus importantes de la construction d'un logiciel est le processus de compilation, qui convertit les fichiers de code source en code exécutable.

Contrôle de versions : le contrôle de révision, également dénommé « contrôle de versions » et « contrôle de sources » (et un aspect de la gestion de configuration des logiciels – ou SCM, *Software Configuration Management*) est la gestion des changements effectués sur des documents, des programmes et d'autres informations stockées sous la forme de fichiers informatiques. Il est couramment utilisé dans le développement de logiciels, où une équipe peut modifier les mêmes fichiers. Les changements sont habituellement identifiés par un code sous forme de nombre ou de lettre, appelé « numéro de révision », « niveau de révision » ou simplement « révision ».

Correctif (*patch*) : logiciel conçu pour corriger des problèmes ou mettre à jour un programme informatique ou prendre en charge ses données. Celui-ci inclut les vulnérabilités de sécurité de correction et d'autre bogues, ainsi que l'amélioration de la convivialité ou des performances.

C++ : le langage de codage avec lequel les logiciels de KDE sont principalement construits.

D

Débogage *post mortem* : débogage après qu'un rapport de bogue a été déposé.

Débogueur : un programme informatique utilisé pour tester et trouver des bogues dans d'autres programmes (les programmes *cibles*).

Dépendance : un paquet qu'il faut installer afin que votre application compile et s'exécute.

Dépôt distribué : une approche poste à poste (*peer-to-peer*) au partage et à la maintenance du code ou d'autre tâches collaboratives, contrairement à l'approche client-serveur des systèmes centralisés. Plutôt qu'un dépôt central unique sur lequel les clients se synchronisent, la copie de travail de chaque homologue de la base de code est un dépôt séparé lié par un réseau de confiance.

Diff : un utilitaire de comparaison de fichiers affichant en sortie les différences entre deux fichiers. Il désigne également la sortie de ce type de programme, qu'on peut appeler un correctif (*patch)* du fait que la sortie peut être appliquée avec le correctif du programme Unix.

digiKam : un organiseur et éditeur d'images utilisant la plate-forme KDE. Il fonctionne sur la plupart des environnements de bureau et des gestionnaires de fenêtres connus, prend en charge tous les principaux formats de fichiers image et peut organiser des collections de photographies dans des albums regroupés dans des dossiers ou des albums dynamiques, par date, chronologie ou étiquettes. Les utilisateurs peuvent également ajouter des légendes et des notes à leurs images, les parcourir et enregistrer des recherches pour une utilisation ultérieure. Avec des modules externes (*plugins*), on peut aussi exporter des albums vers 23hq, Facebook, Flickr, Gallery2, les fichiers KML de Google Earth, SmugMug, Piwigo et Simpleviewer, ou encore les graver sur un CD, voire créer une galerie web.

Distribution (Distro) : une sélection de paquets composant un système de logiciels opérationnel et regroupé pour l'utilisateur. Souvent appliquée au système GNU / Linux en particulier.

Docs : l*a documentation,* une partie essentielle du processus de développement.

E

EBN : « *English Breakfast Network* », un site dédié à la contemplation du thé, la validation de la documentation de l'API de KDE, la validation de la documentation utilisateur, la vérification du code, la réflexion profonde et l'observation des étoiles.

Eclipse : un environnement de développement logiciel multi-langage comprenant un environnement de développement intégré (IDE, *Integrated Development Environment*) et un système de modules externes (*plugins*) extensible. Il est écrit en grande partie en Java et peut être employé pour développer des applications en Java et, au moyen de divers modules externes, d'autres langages de programmation dont Ada, C, C++, COBOL, Perl, PHP, Python, R, Ruby (y compris l'infrastructure Ruby on Rails), Scala, Clojure, Groovy et Scheme.

Éditeur de texte : programme utilisé pour modifier des fichiers en texte simple

Environnement : un ensemble de scripts de configuration et de commandes *bash* fournis comme configuration recommandée lorsqu'on construit manuellement des logiciels KDE.

Environnement de développement intégré (IDE, *Integrated Development Environment*) : une application logicielle fournissant des commodités exhaustives aux programmeurs informatiques pour le développement de logiciels. Un EDI se compose normalement d'un éditeur de code source, d'un compilateur et / ou d'un interpréteur, d'un outil d'automatisation de construction et d'un débogueur (également connu sous le nom d'*environnement de conception intégré*, *environnement de débogage intégré ou environnement de développement interactif)*.

Ergonomie : facilité d'utilisation et d'apprentissage d'un objet créé par un utilisateur, dans ce cas, notre logiciel.

e.V. : signifie « *Eingetragener Verein* » (association déclarée), une association de volontaires enregistrée en Allemagne. La KDE e.V. est une organisation à but non lucratif officielle représentant la communauté KDE sur les aspects légaux et financiers.

F

Forum : un site de discussion en ligne où des personnes peuvent tenir des conversations en postant des messages. Elles diffèrent des salons de discussion en ce sens que les messages sont archivés. Un forum de discussion est une structure hiérarchique ou de type arborescent : un forum peut contenir un certain nombre de sous-forums, dont chacun peut comporter plusieurs sujets. Dans le sujet d'un forum, chaque nouvelle discussion lancée est appelée « fil » (*thread*), et toute personne qui le souhaite peut y répondre.

FOSS : *Free and Open Source Software* (logiciel libre et open source) ou *Free / Libre Open Source Software*, logiciel « open source » / libre / gratuit (F/ OSS, FOSS) est un logiciel libre pour lequel une licence est généreusement accordée aux utilisateurs qui se voient le droit d'utiliser, étudier, partager, modifier et améliorer sa conception grâce à l'accès à son code source.

Frameworks : en commençant par Frameworks 5.0, KDE comporte une feuille de route pour la prochaine version majeure des bibliothèques de KDE et les exigences de temps d'exécution, l'accent étant mis sur la modularité, la clarté des dépendances, la simplification et l'accroissement de la qualité.

Freenode : un réseau IRC utilisé pour discuter de projets dirigés entre homologues.

G

GDB : débogueur de GNU, habituellement appelé juste GDB et nommé « gdb » en tant que fichier exécutable. Il s'agit du débogueur standard pour le système de logiciels de GNU.

Git : un système de contrôle de révisions distribué où l'accent est mis sur la vitesse et la prise en charge de multiples membres, partagé parmi de nombreux développeurs. Chaque dossier de travail Git est un dépôt exhaustif complet présentant un historique complet et des fonctions de suivi des révisions complet, ne dépendant pas de l'accès réseau ou d'un serveur central. Logiciel libre distribué sous GPL v.2.

Gluon : une façon de créer et de jouer à des jeux, et un moyen pour les joueurs ainsi que les créateurs de jeux de se réunir et discuter à propos de leur intérêt partagé. Vous pouvez utiliser le puissant Gluon Creator pour construire des jeux, interagir avec d'autres créateurs et joueurs de jeux sur le site du réseau GamingFreedom.org, voire jouer avec eux sur n'importe laquelle des nombreuses plates-formes prises en charge avec une des applications Gluon Player.

GPL : la *Licence Publique Générale de GNU* est une licence libre sous *copyleft* pour les logiciels et d'autres sortes de travaux.

H

I

Infrastructure : une abstraction dans laquelle les logiciels fournissant de la fonctionnalité générique peuvent être changés sélectivement par du code utilisateur, offrant ainsi les logiciels spécifiques à une application. Il s'agit d'une collection de bibliothèques logicielles prévoyant une interface de programmation d'application définie (API, *Application Programming Interface*).

Internationalisation (i18n) : l'insertion de structures (*constructs*) qui facilitent le changement de l'interface et du langage d'un programme pour différentes cultures et pays (voir également **Localisation**).

Internet Relay Chat (IRC) : un outil texte de communication en temps réel reposant sur du texte. Les canaux KDE se trouvent sur *irc://irc.freenode.net*.

J

K

Kate : un éditeur de texte inclus dans la KDE SC. Le nom « Kate » est un acronyme pour « KDE Advanced Text Editor ».

KDE PIM : gestion de données personnelles de KDE *(KDE Personal Information Management)*, telles que Kontact, KMail, KOrganizer, etc.De plus, un groupe de travail au sein du projet élargi KDE SC qui développe les applications Kontact individuelles de façon coordonnée.

KDE SC : compilation des logiciels KDE (*KDE Software Compilation*), les sources pour la distribution KDE.

KMail : client de courrier électronique de KDE prenant en charge les dossiers, le filtrage, l'affichage du courrier en HTML et les ensembles de caractères internationaux. Il peut manipuler IMAP, IMAP IDLE, dIMAP, POP3, sans oublier les boîtes aux lettres locales pour le courrier entrant. Il peut envoyer du courrier via SMTP ou sendmail.

Konqueror : navigateur web de KDE et gestionnaire de fichiers. Fournit la fonctionnalité d'afficheur de fichiers pour une grande diversité d'aspects : fichiers locaux, fichiers sur un serveur ftp distant et fichiers dans une image disque.

Konsole : un émulateur de terminal gratuit qui fait partie de la KDE SC. Les applications KDE comme Konqueror, Krusader, Kate, Konversation, Dolphin et KDevelop utilisent Konsole pour fournir la fonctionnalité de terminal intégré.

Kontact : gestionnaire d'informations personnelles de KDE et suite de logiciels collaboratifs. Prend en charge les calendriers, les contacts, les notes, les listes de tâches, les nouvelles et le courrier électronique. Utilise KParts pour imbriquer les diverses applications (KMail, KAddressBook, Akregator, etc.) dans l'application du conteneur.

Kopete :multi protocole de KDE, client de messagerie instantanée faisant partie du logiciel libre.

Kpackage Kit : interface de KDE pour PackageKit. PackageKit est une suite « open source » d'applications logicielles conçues pour fournir une interface cohérente et de haut niveau pour un certain nombre de différents systèmes de gestion de paquets.

Kparts : infrastructure de composants pour la KDE SC. Par exemple, Konsole est disponible en tant que KPart et est utilisée dans des applications comme Konqueror et Kate.

Konversation : IRC (*Internet Relay Chat*) convivial construit sur la plate-forme KDE.

KWin : le gestionnaire de fenêtres faisant partie intégrante de la KDE SC. On peut aussi l'utiliser tel quel ou avec d'autres environnements de bureau.

L

LAMP : acronyme pour une intégration de logiciels ou plates-formes composées de Linux, Apache, MySQL et Perl / PHP / Python.

Liste de diffusion : une collection de noms et d'adresses utilisée par un individu ou une organisation pour envoyer des éléments à des destinataires multiples. Souvent étendu pour inclure les personnes abonnées à ce type de liste, ainsi le groupe d'abonnés est désigné sous le nom de *la liste de diffusion* ou simplement *la liste.*

Localisation (l10n) : fait s'afficher les changements requis pour afficher l'interface d'un programme à l'aide de la langue et les conventions d'un pays particulier (voir aussi **Internationalisation**).

M

N

Nepomuk : projet d'espaces de travail pour KDE prenant en charge les annotations, l'indexation, la recherche et l'édition de liens.

Nom d'écran : le nom d'écran *(nick)* ou pseudonyme d'un utilisateur.

Nuit (de) : une construction neutre qui reflète l'état actuel du code source contrôlé dans le système de contrôle de versions par les développeurs, tel qu'il est construit dans un environnement neutre (c'est-à-dire dans un environnement non utilisé pour le développement). Une construction de nuit est une construction neutre qui se déroule automatiquement, généralement chaque nuit. Le Projet Neon est un projet de ce type pour KDE.

O

Octroi de licence : instruments légaux (habituellement par voie de droit des contrats) gouvernant l'utilisation ou la redistribution des logiciels. Tous les logiciels sont protégés par des droits d'auteurs (*copyright*), sauf les éléments tombés dans le domaine public.

Okular : afficheur de documents universel de KDE reposant sur KPDF.

OOP (*Object-oriented programming*) : un paradigme de programmation utilisant des *objets* – structures de données composés de champs de données et de méthodes conjointement avec leurs interactions – pour concevoir des applications et des programmes informatiques.

Opérateurs (*IRCops*) : opérateurs de canal ayant des pouvoirs sur le canal IRC, comprenant la modération ou l'expulsion d'utilisateurs perturbateurs. Comme les *IRCops* ou *sysops* contrôlent le serveur IRC, ils contrôlent les canaux tout en ayant le contrôle sur qui peut participer. Sur la plupart des systèmes, les opérateurs (*ops*) sont identifiés par un symbole situé près de leurs pseudonymes, mais Freenode déconseille aux opérateurs d'apparaître comme tels à moins qu'ils n'aient du travail à effectuer sur le canal.

P

Paquet, empaquetage : il existe deux types de paquets susceptibles d'être téléchargés depuis le site FTP de KDE : les paquets binaires ou exécutables (rpm, deb et similaires) et les paquets source. Les paquets binaires sont des versions compilées (« exécutables ») de la KDE SC construits pour s'exécuter sur un système d'exploitation ou une distribution spécifique. Les paquets source représentent le code brut qui compose la KDE SC et il faut les compiler avoir de pouvoir les utiliser. Les paquets de logiciels KDE disponibles depuis les distributions peuvent être légèrement différents des paquets source KDE purs.

Pastebin : une application web permettant aux utilisateurs d'envoyer sur un serveur des fragments (*snippets*) de texte, habituellement des échantillons de code source, pour un affichage public. L'utilisation en est encouragée dans sur canaux IRC, où coller de grandes quantités de texte est considéré comme contraire à l'étiquette. Le programme « pastebin » de KDE est hébergé sur http://paste.kde.org.

Phonon : API multimédia fournie par Qt ; l'abstraction standard pour gérer les flux multimédia au sein de la KDE SC.

Pile d'appels (*backtrace*) : également appelée *trace arrière de la pile* (stack backtrace) ou trace d'appels (stack trace). Il s'agit d'un rapport des cadres de piles actifs à un certain moment dans le temps pendant l'exécution d'un programme. Lorsqu'on utilise des logiciels KDE, on obtient la meilleure pile d'appels depuis Dr. Konqui, si elle apparaît immédiatement après une panne. Sinon, on peut faire appel à GDB pour obtenir une pile d'appels.

Plasma Active : la toute dernière initiative de l'équipe de développement de Plasma, apportant la fonctionnalité KDE aux périphériques mobiles.

Plasma : infrastructure de la KDE SC pour faciliter la création de composants graphiques (*widgets*). Ceux-ci regroupent les lanceurs d'applications interactifs, les gestionnaires de fenêtres et de tâches, et plus.

Plasmoïde (composant graphique) : composant graphique dans l'environnement Plasma Desktop.

Q

QtCreator : un EDI intégré, multi plate-forme pour C++ et JavaScript faisant partie du SDK Qt.

Qt : infrastructure d'applications multi plate-forme largement utilisée pour des logiciels d'application de développement avec une interface graphique utilisateur (GUI, *Graphical User Interface*).

Quassel : client d'IRC distribué, multi plate-forme, ce qui signifie qu'un (ou plusieurs) clients peuvent se connecter et se déconnecter d'un noyau central -- tout comme la combinaison bien connue de clients IRC en mode écran et en mode texte tels que WeeChat *(Wee Enhanced Environment for Chat)*, mais graphique.

R

rekonq : navigateur web pour KDE reposant sur WebKit.

Révision: examen systématique (souvent en tant que révision par des pairs) de code source informatique. Il est destiné à trouver et corriger des erreurs oubliées dans la phase de développement initiale, améliorant à la fois l'ensemble de la qualité des logiciels et les compétences des développeurs.

S

Script : petit programme écrit pour un interpréteur de commandes ou un autre langage de scriptage.

Serveur : programme informatique s'exécutant pour satisfaire les requêtes d'autres programmes, les *clients*.

Solid : interface d'intégration de périphériques pour la KDE SC. Elle fonctionne selon des principes similaires à Phonon, le pilier multimédia de KDE ; plutôt que gérer le matériel en propre, elle rend au contraire accessibles des solutions existantes grâce à une seule API.

Source : instructions lisibles par l'utilisateur dans un langage de programmation, en vue d'être transformées en instructions machine par un compilateur, un interpréteur, un assembleur ou un autre système de ce type.

Sprint : rencontre en face à face de membres d'équipes qui habituellement travaillent ensemble à distance.

Suite : collection de programmes informatique, le plus souvent des logiciels d'applications et de programmation de fonctionnalité apparentée, partageant souvent une interface utilisateur plus ou moins commune et une certaines capacité à échanger sans problème des données entre eux.

Summit (Sommet) : dans KDE et FOSS, une rencontre important de membres qui habituellement travaillent distance. Les *sprints* d'équipe peuvent avoir lieu avant, pendant et après un grand sommet.

SVN (Subversion) : un système de contrôle de versions et de révisions de logiciels distribués sous licence libre, faisant partie de la Fondation Apache.

Système de construction : outils logiciels qui scriptent ou automatisent une grande variété de tâches qu'effectuent les développeurs dans leurs activités quotidiennes, telles que compiler du code source informatique en code binaire (exécutable), conditionner du code binaire, exécuter des tests, déployer du code pour des systèmes de production et créer de la documentation, sans oublier les notes de
version.

T

Techbase : wiki de documentation pour les développeurs de KDE.

Telepathy : une infrastructure de communication en temps réel prenant en charge la messagerie instantanée, la VoIP et la collaboration.

Terminal : interface pour la saisie en série et l'affichage de données texte. Voir aussi **Console**.

Tests : investigation menée pour fournir aux participants des informations sur la qualité du produit ou du service en cours de test.

Tests unitaires (*Unit tests*) : méthode grâce à laquelle des unités individuelles de code source sont testées pour déterminer s'ils sont adaptés à l'utilisation. Une unité est la plus petite partie testable d'une application. En programmation orientée objet, une unité est habituellement un une interface, telle qu'une classe.

Traitement de texte : application informatique utilisée pour la production (y compris la composition, l'édition, la mise en forme (formatage) et éventuellement l'impression) de n'importe quel type d'élément imprimable.

Trunk (Tronc) : la branche (version) sans nom d'un arbre de fichiers sous contrôle de révision. Le tronc (*master*) est d'ordinaire à la base d'un projet sur lequel le développement progresse.

U

Userbase : wiki de documentation de l'utilisateur de KDE.

V

Valgrind : outil de programmation soumis à la licence GPL conçu pour le débogage de la mémoire, la détection des fuites de mémoire et le profilage. Le nom « Valgrind » vient du nom désignant l'entrée principale du Valhalla dans la mythologie scandinave.

Valider (*Commit*) : rendre permanent un ensemble de modifications provisoires.

Variable : nom symbolique donné à une quantité connue ou inconnue d'informations, dans le but de permettre d'utiliser le nom indépendamment des informations qu'elles représentent. Un nom de variable dans du code source informatique est habituellement associé à un emplacement de stockage de données et donc aussi à son contenu. Ceux-ci peuvent changer au cours de l'exécution du programme.

W

Wiki : site web permettant la création et l'édition d'un nombre quelconque de pages web interconnectée via un navigateur web utilisant un langage de balisage simplifié ou un éditeur de texte « WYSIWYG ». Les wikis sont généralement optimisés par un logiciel de wiki et sont souvent employés de façon collaborative par des utilisateurs multiples.

X

X, système X window : système de logiciels informatiques et protocole réseau fournissant une base pour les interfaces graphiques utilisateur (GUI, *Graphical User Interface*) et une capacité de périphérique d'entrée complète pour les ordinateurs connectés en réseau. Il crée une couche d'abstraction matérielle où le logiciel est écrit pour utiliser un ensemble généralisé de commandes, permettant l'indépendance des périphériques et la réutilisation de programmes sur n'importe quel ordinateur qui implémente X. **X.Org** sert d'implémentation canonique de X, et constitue ce qu'utilise la KDE SC.

Y

Yakuake : émulateur de terminal déroulant reposant sur la technologie Konsole de KDE.

Z

Z-Machine : machine virtuelle utilisée par Infocom pour ses jeux d'aventure texte. Kwest est un interpréteur Z-machine pour KDE.

19. LIENS UTILES

KDE : http://kde.org/

La page d'accueil du Projet KDE

KDE Techbase : http://techbase.kde.org/

Le wiki consacré au développement de KDE

KDE Userbase : http://userbase.kde.org/

Le wiki dédié aux utilisateurs de KDE

KDE Identity : https://identity.kde.org/

Un système de pointage unique utilisé à travers plusieurs sites web KDE

Page des projets KDE : https://projects.kde.org/

Une vue d'ensemble de tous les projets hébergés dans git.kde.org reposant sur la technologie KDE

Tutoriels Qt : http://doc.qt.nokia.com/

Une référence exhaustive de la documentation de Qt

Git en 30 minutes : http://blip.tv/scott-chacon/git-in-30-minutes-4064151

Une vidéo utile présentant les concepts concernant « git »

The Dot : http://dot.kde.org/

La publication officielle de la communauté KDE

Behind KDE : http://behindkde.org/

Des personnes dans les coulisses de KDE interviewent ceux qui travaillent sur KDE

Planet KDE : http://planetkde.org/

Agrégation des blogues des membres de la communauté KDE

The KDE Release Schedule : http://techbase.kde.org/Schedules

20. À PROPOS DE CE LIVRE

Remerciements

Quelques personnes nous ayant réellement aidé(e)s à faire de ce livre ce qu'il est, il est incomplet sans un vote de remerciements et un témoignage d'appréciations à leur égard. Donc, dans un ordre spécifique :

- Gunner, de la société Aspiration (http://www.aspirationtech.org/), nous a inspiré et rendu tous plus intelligents, comme l'ont fait d'autres équipes lors du concours, notamment OpenMRS, OpenStreetMap et Eden.
- Nóirín Plunkett & Belinda Lopez ont fourni quantités de ressources d'écriture et d'édition pour nous tous.
- Adam, de Floss Manuals nous a apporté une aide continuelle et l'équipe de Booki a travaillé avec nous sur la fonctionnalité nécessaire et la correction de bogues au fur et à mesure que la semaine et le travail avançaient.

Nous remercions également la communauté KDE, qui nous a offert une atmosphère dans laquelle nous pouvions à la fois progresser et évoluer en tant que contributeurs et êtres humains, mais aussi pour nous avoir fourni une merveilleuse documentation dans laquelle nous avons pu puiser en toute liberté.

En particulier, Ingo Malchow, Lydia Pintscher et l'équipe de promotion de KDE nous a apporté une grande aide dans notre temps limité.

L'équipe

Ce livre a été initialement créé au Concours du Livre de trois jours en octobre 2011, au Googleplex à Mountain View, Californie. Carol Smith, du renommé « Google Summer of Code » nous a obtenu le financement nécessaire pour faire venir en avion Karan, Seb et Rohan depuis l'Inde, pour s'occuper de la maison et nous nourrir pendant que nous travaillions. Rohan est arrivé 21e au concours et le groupe a célébré l'évènement avec une partie de X-box !

Rohan Garg - rohan16garg@gmail.com
Supreet Pal Singh - supreetpal@gmail.com
Karan Pratap Singh - wizard.karan@gmail.com
Valorie Zimmerman - valorie.zimmerman@gmail.com
Andy Oram - andyo@oreilly.com

L'équipe KDE au « Doc Sprint », octobre 2011

La traduction

Remercions le travail de qualité de Joëlle Cornavin, membre de l'Équipe française de traduction de KDE, qui a réalisé entièrement seule la traduction de ce livre.