

Les Outils de Développement

Gerald Baratoux

Les outils de développement

- ▶ Cours Amphi (4H)
- ▶ 1^{er} TP (4H) -> Les outils de versionnement
- ▶ 2^{ème} TP (4H) -> Les outils de gestion de projet
 - > Les outils de génération de documentation
- ▶ 3^{ème} TP (8H) -> Mise en application



Les outils de versionnement



Les outils de versionnement

- ▶ Un gestionnaire de version permet de donner des numéros de version successifs à des objets informatiques:
 - ▶ fichiers texte : code source, makefile, fichiers d'entrée/sortie
 - ▶ fichiers binaires (pas avec tous les gestionnaires de version)
 - ▶ arborescence d'un projet
- ▶ Seules les modifications successives sont stockées (diff), permettant ainsi de ne rien perdre sans pour autant conserver l'ensemble des fichiers.



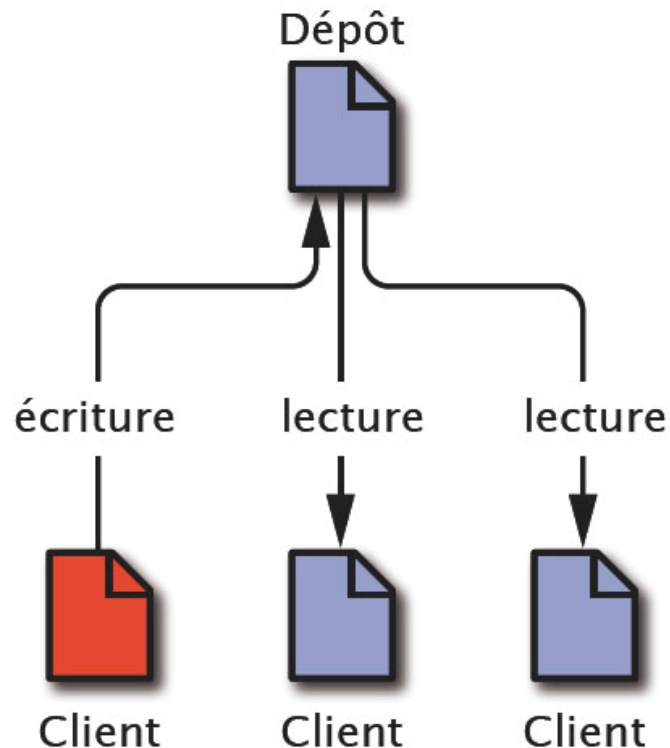
Les outils de versionnement

- ▶ Objectifs
 - ▶ faciliter la tâche de développement
 - ▶ assister/permètre le développement simultané en équipe
 - ▶ pérenniser les développements



Les outils de versionnement

► Système client Serveur



- Le dépôt contient l'ensemble du code source d'un projet
- Les développeurs peuvent récupérer le contenu du dépôt
- Les développeurs peuvent modifier les fichiers du projet
- Les modifications sont enregistrées dans le dépôt sans perte des versions précédentes (basé sur diff)

Les outils de versionnement

- ▶ Aucune perte dans la chronologie des développements
- ▶ Développement multi-utilisateurs rendu possible
- ▶ Recherche chronologique multi-critères (auteur, date, message, etc.)
- ▶ Consultation de l'évolution chronologique d'un fichier avec dates de modifications, type d'ajouts, auteur, etc.
- ▶ Branchements pour différentes versions de code avec tronc commun



Les outils de versionnement

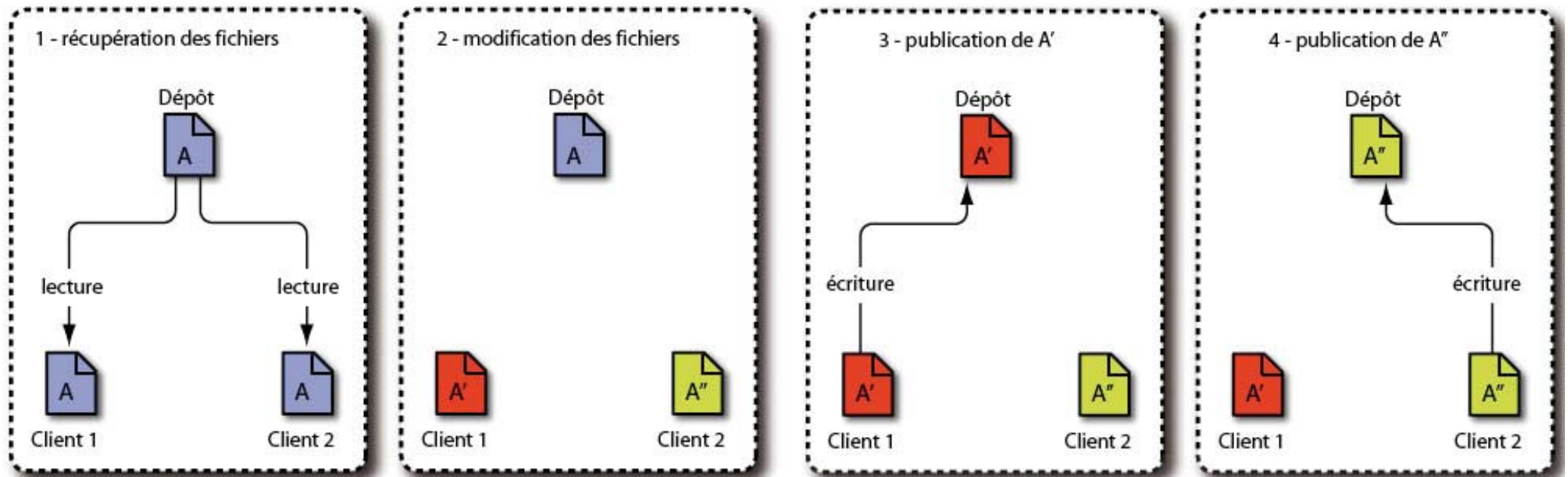
Conséquences :

- ▶ Développements durables et sûrs
- ▶ Sécurité pour les sauvegardes



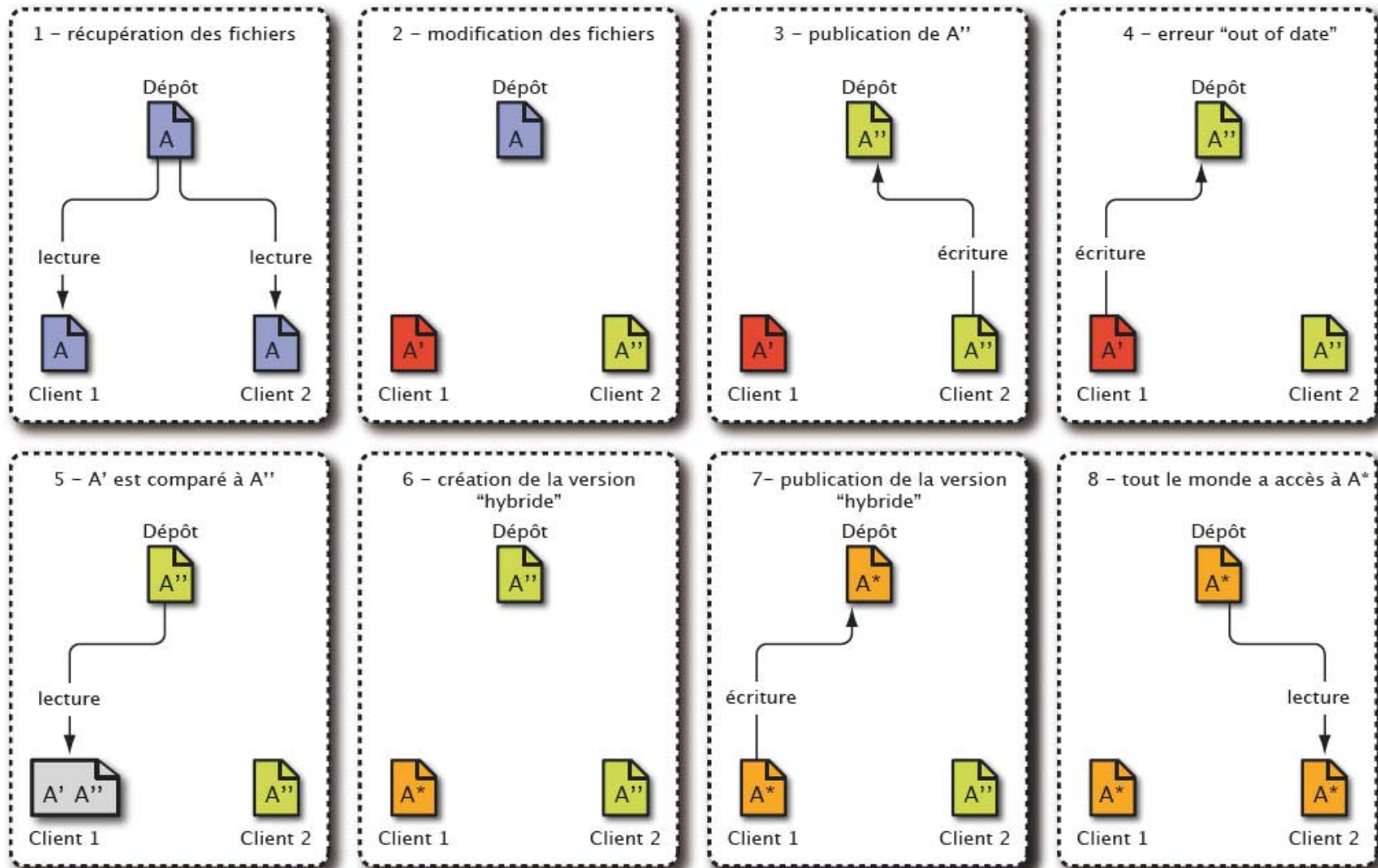
Les outils de versionnement

► Les problèmes à éviter



Les outils de versionnement

► Les gestionnaires modernes



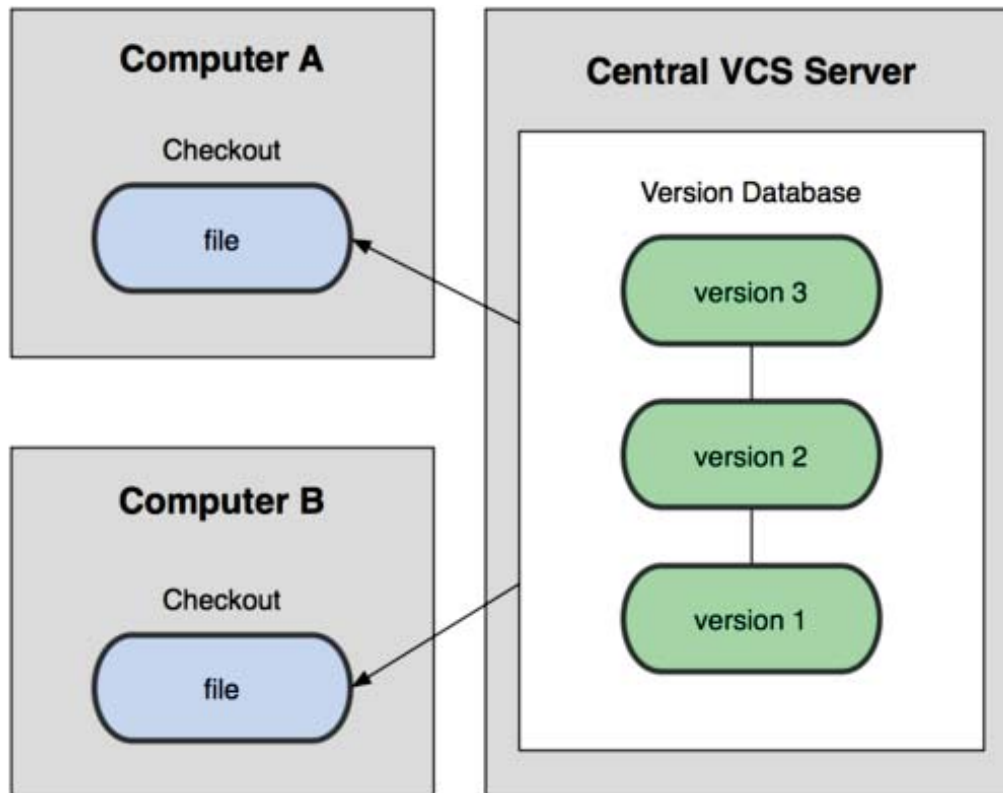
Les outils de versionnement

- ▶ Plusieurs outils de versionnement existent :
 - ▶ **Gestion de versions centralisée**
CVS, Subversion
 - ▶ **Gestion de versions décentralisée**
Git, Mercurial, ...



Les outils de versionnement

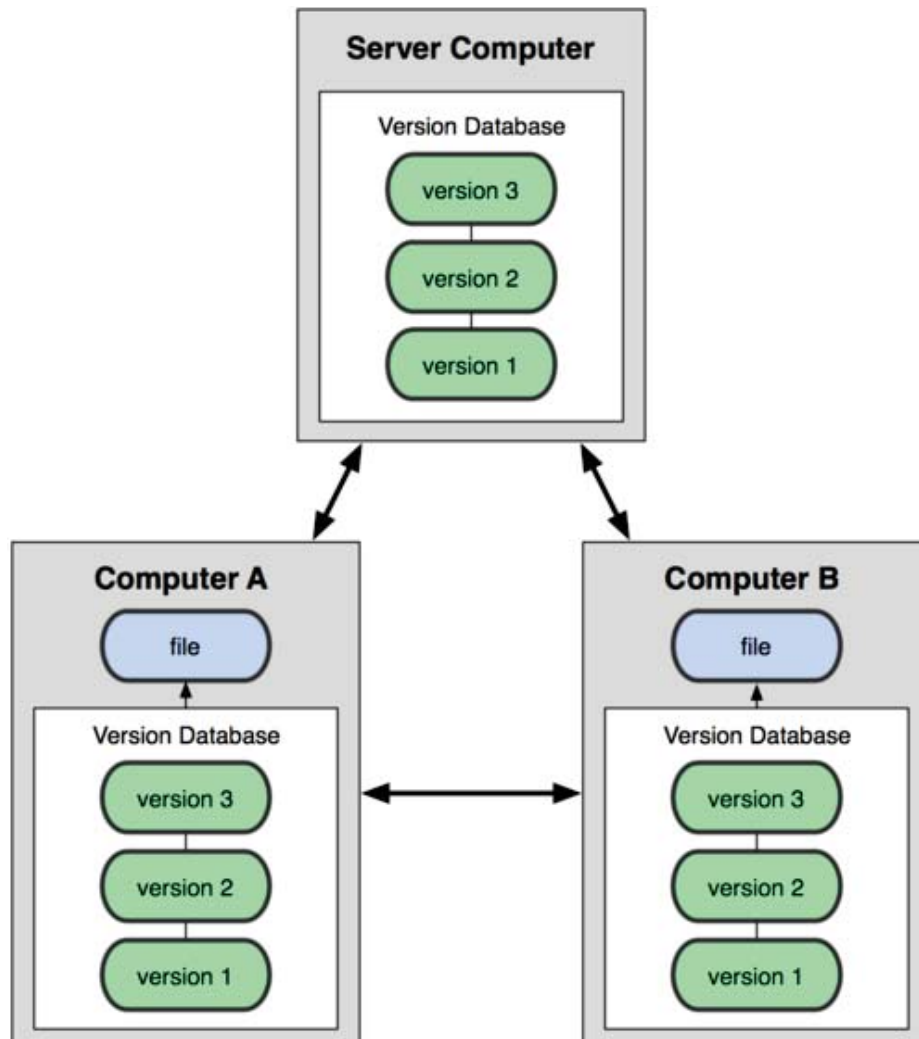
► Gestion de versions centralisée



Serveur central qui contient tous les fichiers sous gestion de version et des clients qui peuvent extraire les fichiers de ce dépôt central.

Les outils de versionnement

► Gestion de versions décentralisée



Les clients dupliquent complètement le dépôt. Chaque extraction devient une sauvegarde complète de toutes les données.

Git

Carte d'identité



Système de gestion de version distribué (DVCS).

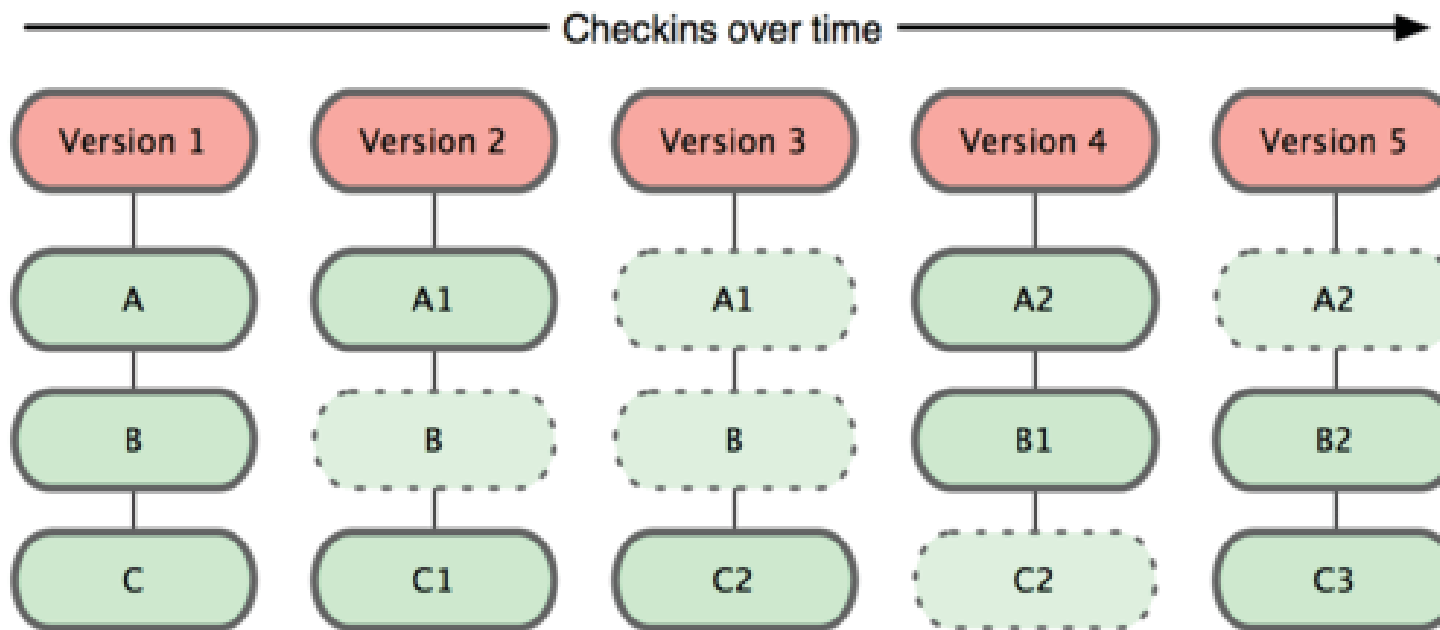
Bref historique (Wikipedia)

- ▶ De 1991 à 2002, le noyau Linux était développé sans utiliser de système de gestion de version.
- ▶ A partir de 2002, la communauté a commencé à utiliser BitKeeper, un DVCS propriétaire.
- ▶ En 2005, suite à un contentieux, BitKeeper retire la possibilité d'utiliser gratuitement son produit. Linus Torvalds lance le développement de Git et après seulement quelques mois de développement, Git héberge le développement du noyau Linux.

Git

Les principes de base

Un **dépôt** Git est une sorte de système de fichiers (base de données), enregistrant les versions de fichiers d'un projet à des moments précis au cours du temps sous forme d'instantanés.



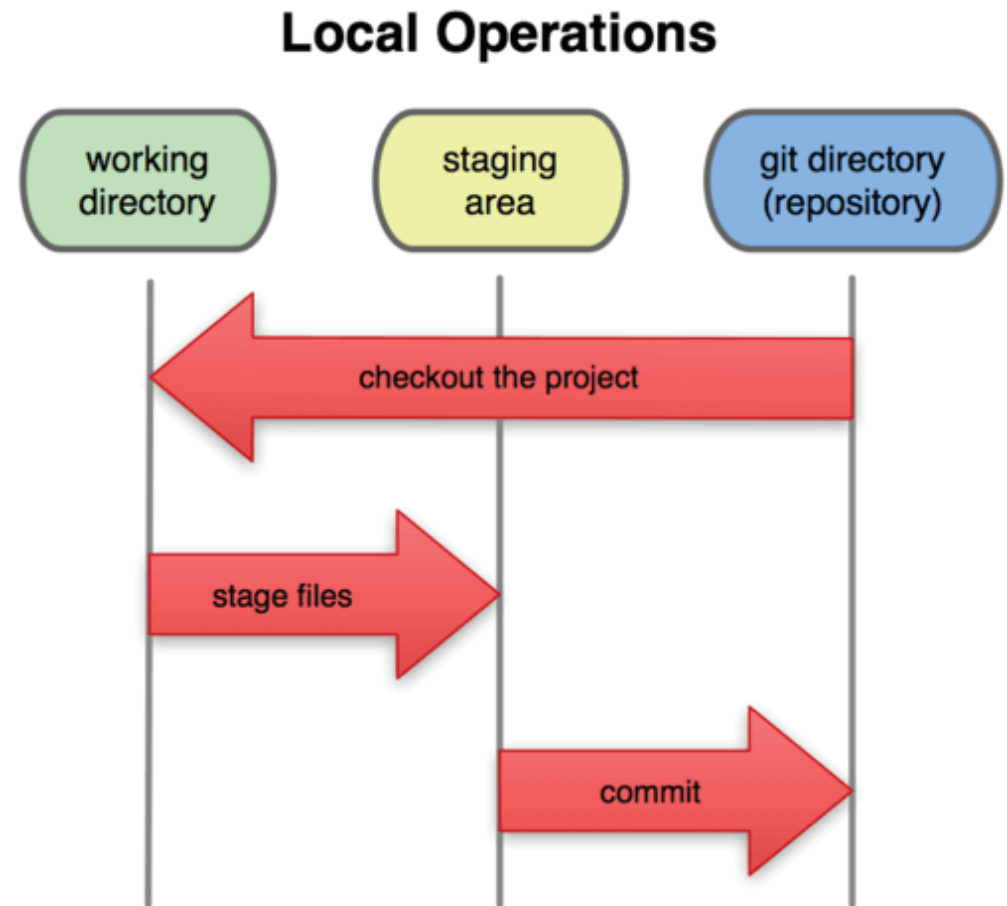
Comment fonctionne le processus de versionnement ?

Git

Les principes de base

3 sections d'un projet Git :

- ▶ **Le répertoire Git/dépôt** :
contient les méta-données et la base de données des objets du projet.
- ▶ **Le répertoire de travail** :
extraction unique d'une version du projet depuis la base de données du dépôt.
- ▶ **La zone de transit/d'index** :
simple fichier contenant des informations à propos de ce qui sera pris en compte lors de la prochaine soumission.



Git

Les commandes de base

Initialiser un dépôt

```
$ git init
```

Afficher l'état des fichiers du répertoire courant

```
$ git status
```

- ▶ *Untracked files* : fichiers non versionnés.
- ▶ *Changes to be committed* : modifications (ajout, suppression, changements) chargées en zone de transit (staging area), ou indexées.
- ▶ *Changes not staged for commit* : modifications n'ayant pas été chargées en zone de transit (ou indexées).

Git

Les commandes de base

Indexer l'ajout ou les changements d'un fichier

```
$ git add <fichier>
```

Annuler les modifications indexées d'un fichier

```
$ git reset <fichier>
```

Annuler les modifications non encore indexées d'un fichier

```
$ git checkout [--] <fichier>
```

Indexer la suppression d'un fichier

```
$ git rm <fichier>
```

Déversionner un fichier

```
$ git rm --cached <fichier>
```

Git

Les commandes de base

Afficher le détail des modifications non indexées

```
$ git diff
```

Afficher le détail des modifications indexées

```
$ git diff --staged
```

Soumettre les modifications indexées en zone de transit

```
$ git commit
```

Voir l'historique des soumissions

```
$ git log
```

Git

Travail avec des dépôts distants

Pour **collaborer**, il est nécessaire de communiquer et d'échanger avec un ou plusieurs dépôts distants hébergeant le même projet (typiquement des dépôts publics associés à une personne, une équipe ou tout le projet).

Les données des dépôts distants (objets “commit” et instantanés) sont entièrement copiées dans le dépôt local, et pour chaque branche <branche> d'un dépôt distant <dépôt> est maintenue une branche locale <dépôt>/<branche> non modifiable, permettant de suivre la position de <branche> sur <dépôt> localement.

Afficher la liste de tous les dépôts distants du projet

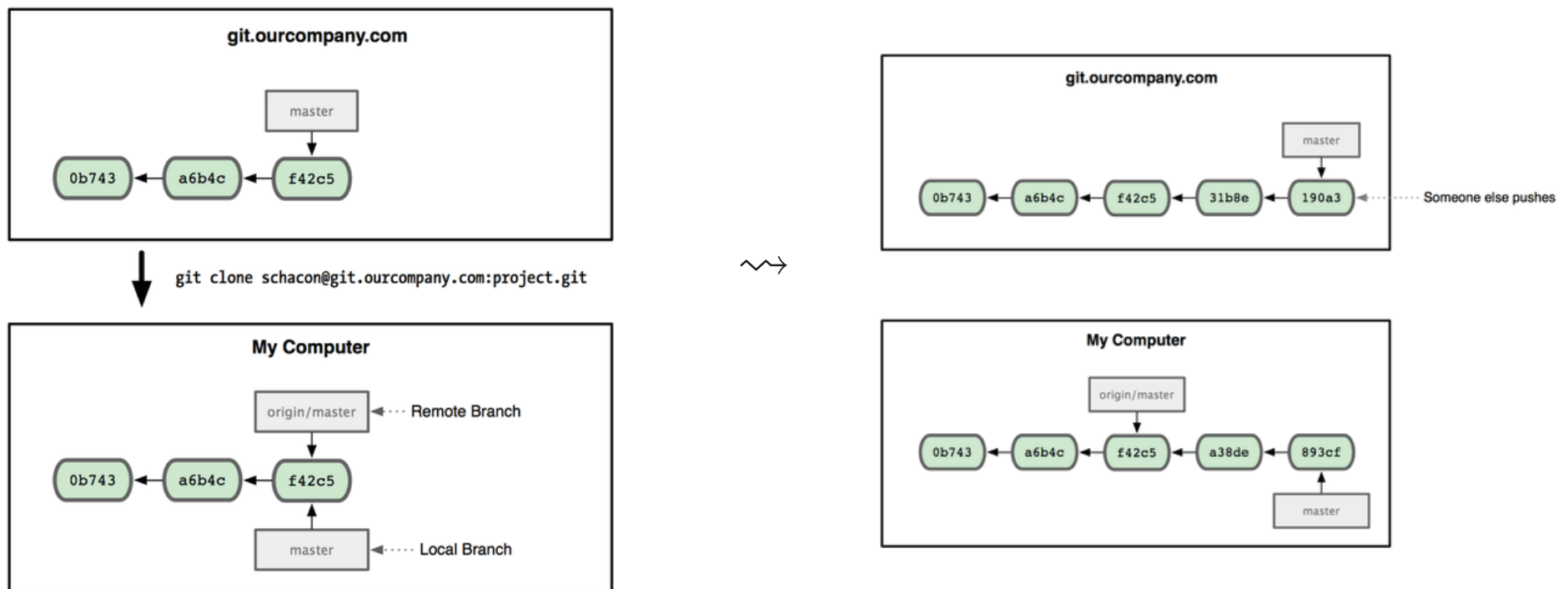
```
$ git remote
```

Git

Travail avec des dépôts distants

Cloner un dépôt distant (automatiquement nommé origin)

```
$ git clone <URL> [<répertoire>]
```

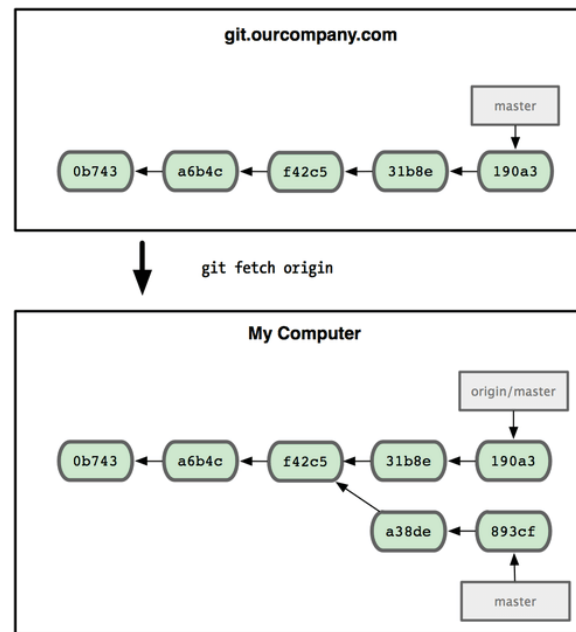


Git

Travail avec des dépôts distants

Récupérer les modifications d'un dépôt distant

```
$ git fetch <dépôt>
```

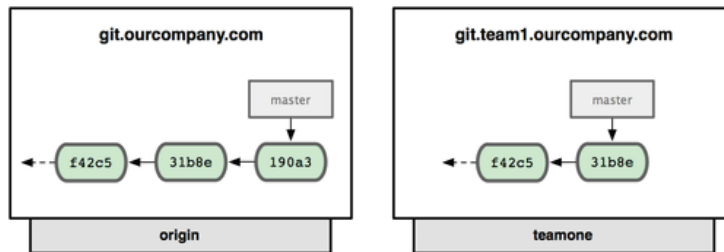


Git

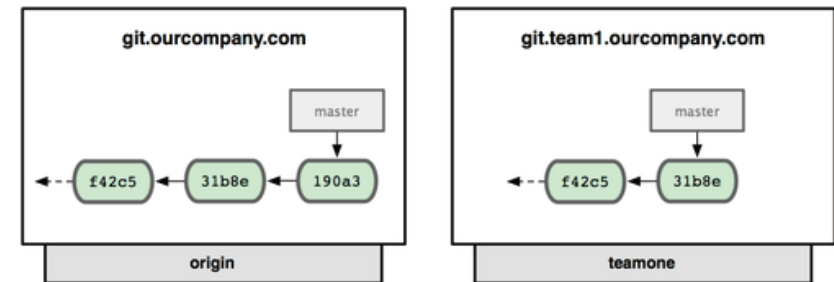
Travail avec des dépôts distants

Ajouter un dépôt distant

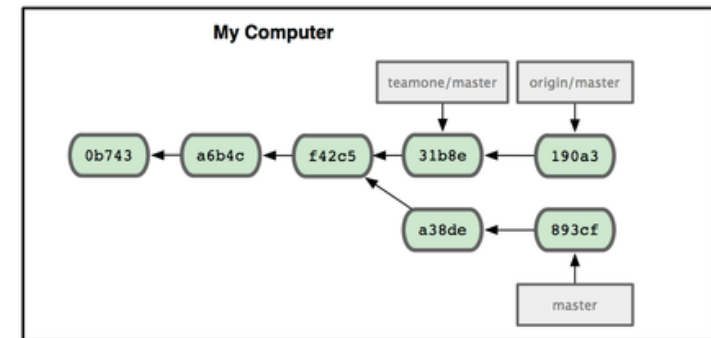
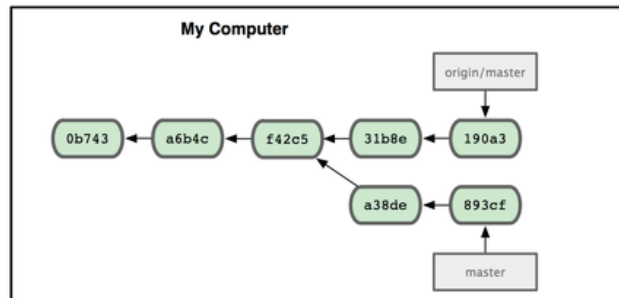
\$ git remote add <nom> <URL>



git remote add teamone git://git.team1.ourcompany.com



git fetch teamone



Git

Travail avec des dépôts distants

Mettre à jour un dépôt distant donné pour une certaine branche locale

```
$ git push <dépôt> <branche>
```

Attention : ceci fonctionne uniquement si le dernier objet “commit” de la branche concernée du dépôt distant a été récupéré et intégré dans la branche en question du dépôt local, en d’autres termes si la mise à jour de la branche du dépôt distant peut être faite par “fast-forward”.

Combiner git fetch et git merge

```
git pull <dépôt> <branche>
```

Combiner git fetch et git rebase

```
git pull --rebase <dépôt> <branche>
```


Git

Conseils, bonnes pratiques

Conseils

- ▶ User et abuser des branches.
- ▶ Eviter de systématiquement faire un `git pull`, mais réfléchir à ce qu'il est le plus pertinent de faire (fusionner ou rebaser).

Bonnes pratiques

- ▶ Ne jamais rebaser une branche déjà présente sur un dépôt public.
- ▶ Respecter les conventions de formatage du message de soumission : titre de 50 caractères au plus, suivi d'une ligne vide puis d'une description détaillée avec des lignes de 72 caractères au plus, le tout au présent de l'impératif (<http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html>).

Subversion

- ▶ Très répandu
 - ▶ Remplaçant de CVS
 - ▶ licence Apache et BSD
- ▶ Plusieurs interfaces graphiques
- ▶ Gestion facile des ajouts/suppressions de fichiers/répertoires
- ▶ Gestion des fichiers binaires
- ▶ Gestion des permissions des utilisateurs
- ▶ Incrémentation du numéro de version du projet

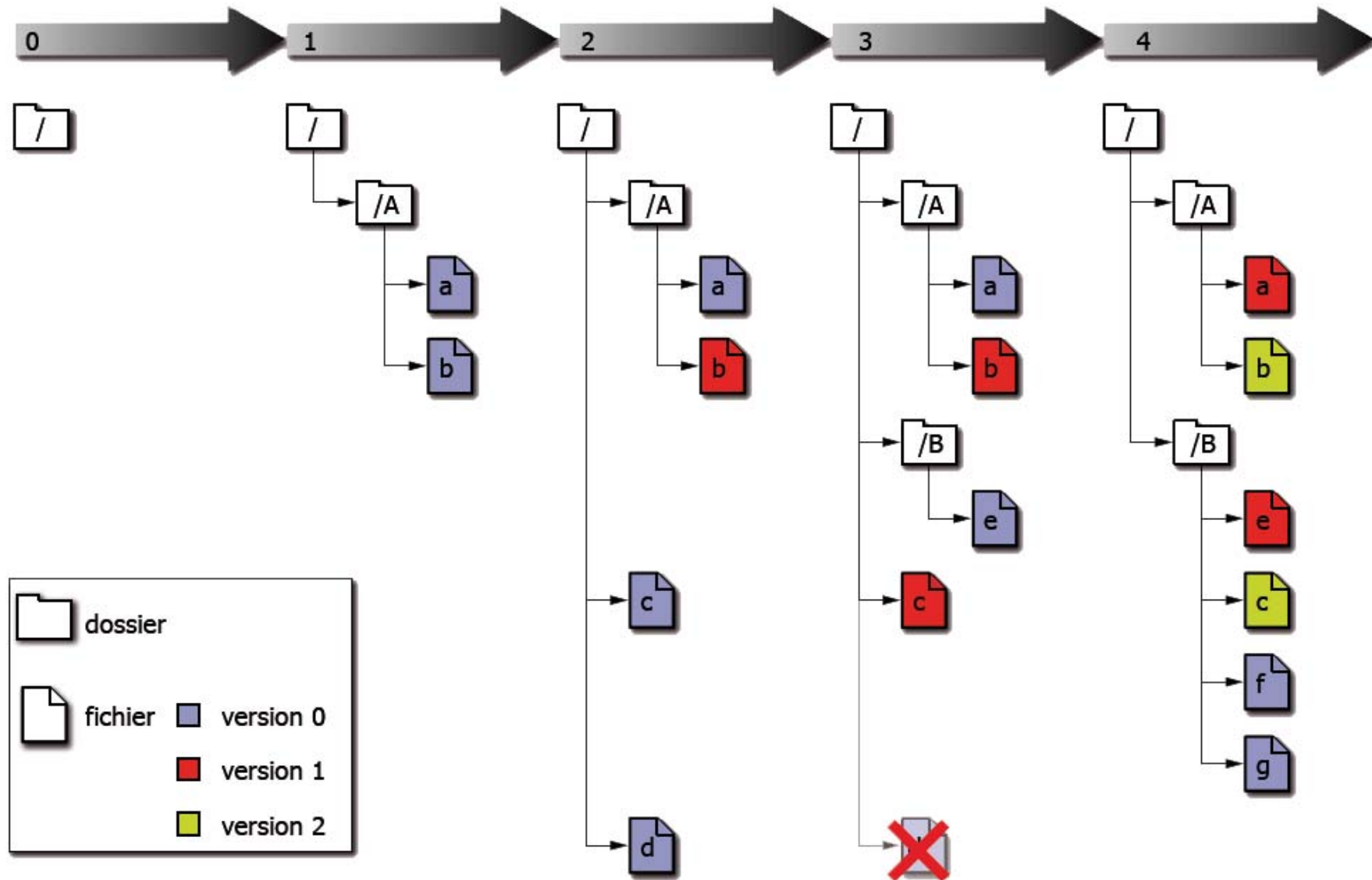


Chronologie d'un projet Subversion

- ▶ Création du dépôt et des droits utilisateurs
- ▶ Mise à jour de sa copie : **svn update**
- ▶ Changements dans le code : édition du source, **svn add**, **svn delete**, **svn copy**, **svn move**
- ▶ Examen des changements par rapport au dépôt : **svn status**, **svn diff**, **svn revert**
- ▶ Merge des modifications du dépôt en cas de conflit : **svn update**, **svn resolved**
- ▶ Soumission des modifications au dépôt : **svn commit**



Chronologie d'un projet Subversion



Accéder à Subversion

- ▶ Subversion peut être joint
 - ▶ Avec des clients SVN au travers une connexion HTTP ou HTTPS par extension du serveur Apache
 - ▶ En utilisant un protocole réseau propriétaire (grâce à la librairie de dev SVN)
 - ▶ En utilisant une interface Web

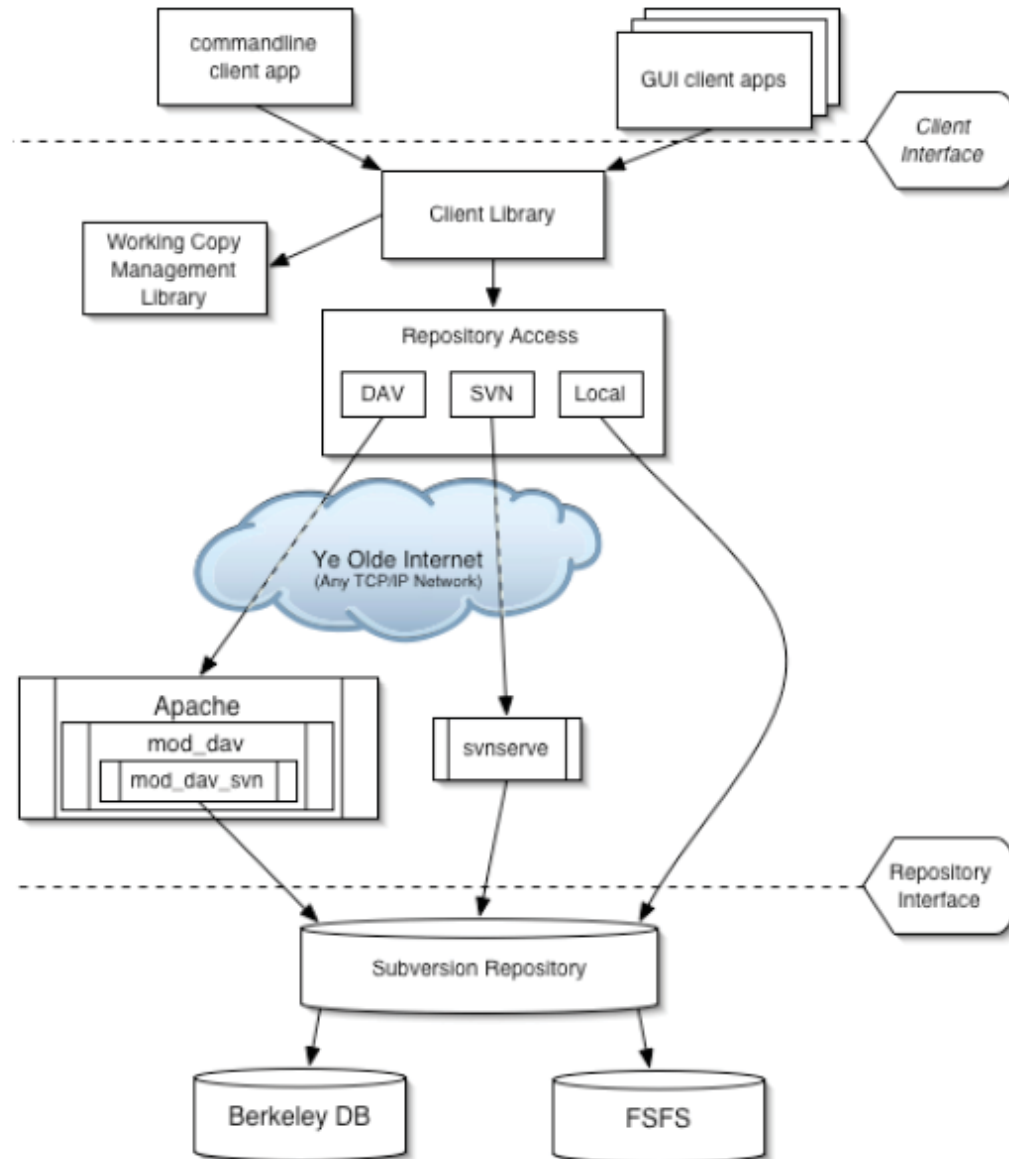


Les clients Subversion

- ▶ TortoiseSVN – qui permet d'accéder aux fichiers grâce à l'explorateur windows
- ▶ Subclipse – plugin SVN pour Eclipse
- ▶ AnkhSVN – plugin SVN pour Visual Studio
- ▶ De nombreux autres clients graphiques ...
- ▶ En ligne de commande



Connexions avec Subversion



Subversion en ligne de commande

► Projets

- Au sein d'un dépôt se trouvent un ou plusieurs projets. À chaque projet correspond en général un répertoire situé à la racine du dépôt et qui contient lui-même les fichiers et dossiers du projet proprement dit.
Exemple d'arborescence :

```
(dépôt)--+--/batchxsl--+--/trunk
|
|      |
|      +--/branches
|      |
|      +--/tags
|
+--/css-----+--/trunk
|
|      |
|      +--/branches
|      |
|      +--/tags
|
+--/test-----+--/rep1
|
|      +--/rep2
```


Subversion en ligne de commande

- ▶ **Projet déjà existant au sein d'un dépôt**
- ▶ Si le projet existe déjà au sein du dépôt, une seule commande suffit pour effectuer un *checkout* et récupérer la dernière version des fichiers : il s'agit de la commande *svn co*.

```
$ svn co https://cens-srv-dev1.ens-lsh.fr/svnrep/formationsvn .  
A credits.htm  
A index.htm  
Checked out revision 36.
```

La commande précédente a effectué un checkout du projet formationsvn (situé dans un répertoire racine du même nom) dans le répertoire courant.

Le résultat de la commande indique que deux fichiers ont été récupérés et que la dernière révision est la révision numéro 36.



Subversion en ligne de commande

- ▶ **Import d'un projet déjà existant en local**
- ▶ Si le projet n'existe pas dans le dépôt et qu'il faut le créer à partir de fichiers locaux, la commande à utiliser est `svn import`. Cette opération n'est en théorie effectuée que par la personne chargée de l'administration du dépôt.



Subversion en ligne de commande

- ▶ **Récupération de la dernière version du projet**
- ▶ Avant de travailler sur les fichiers du projet, il faut s'assurer que l'on est bien synchronisé avec le dépôt, c'est à dire que la copie de travail correspond bien à la dernière révision en cours. Pour cela, il faut effectuer un *update* à l'aide de la commande `svn update` :

```
$ svn update
U index.htm
Updated to revision 37.
```

La commande indique qu'un fichier du répertoire, vraisemblablement modifié par quelqu'un d'autre depuis notre dernier *update*, a été mis à jour dans notre copie de travail.



Subversion en ligne de commande

- ▶ **Mise à jour des modifications dans le dépôt**
- ▶ Une fois qu'on a modifié des fichiers, il faut basculer ces modifications au sein du dépôt pour qu'elles soient accessibles aux autres utilisateurs. Cette opération s'effectue à l'aide de la commande `svn commit` :

```
$ svn commit -m "Ajout d'une personne dans les credits"  
Sending credits.htm  
Transmitting file data .  
Committed revision 38.
```

Toute opération de commit s'effectue en indiquant un message décrivant les modifications effectuées (ici directement dans la ligne de commande). Il est possible d'effectuer cette opération sur un répertoire entier, ou sur seulement un ou plusieurs fichiers.

Si des modifications ont eu lieu par un autre utilisateur du dépôt depuis le dernier update, un message d'erreur le signale. Il faut alors effectuer un nouvel update et résoudre d'éventuels conflits avant de relancer le commit.



Subversion en ligne de commande

- ▶ **Récupération de la dernière version**
- ▶ Lorsqu'on travaille sur un fichier, il peut arriver que les modifications effectuées ne soient pas bonnes et qu'on souhaite retourner au fichier tel qu'il était lors du dernier *update*. La commande `svn revert` est faite pour ça :

```
$ svn revert credits.htm  
Reverted 'credits.htm'
```

Cette commande annule les modifications effectuées depuis le dernier *update*. À noter que tout est effectué en local, et qu'un accès au dépôt n'est pas nécessaire.



Subversion en ligne de commande

- ▶ **Récupération d'une version antérieure du dépôt**
- ▶ On peut aussi souhaiter revenir à une version antérieure d'un fichier situé dans le dépôt. Il faut alors utiliser `svn update` en précisant le numéro de la révision et le ou les fichiers :

```
$ svn update -r 36 credits.htm  
U credits.htm  
Updated to revision 36.
```



Subversion en ligne de commande

- ▶ **Ajout d'un fichier**
- ▶ Il faut utiliser `svn add`.

A noter que l'ajout n'est effectif qu'au prochain *commit* :

```
$ svn add liens.htm
A liens.htm
$ svn commit -m "Ajout du fichier de liens«
Adding liens.htm
Transmitting file data .
Committed revision 39.
```



Subversion en ligne de commande

- ▶ **Suppression d'un fichier**
- ▶ Il faut utiliser `svn delete` :

```
$ svn delete liens.htm  
D liens.htm  
$ svn commit -m "Suppression d'un fichier"  
Deleting liens.htm  
Committed revision 40.
```

Là aussi, la suppression n'est effective qu'au *commit* suivant.



Subversion en ligne de commande

- ▶ **Renommer un fichier**
- ▶ Il faut utiliser svn move :

```
$ svn move credits.htm merci.htm  
A merci.htm  
D credits.htm  
$ svn commit -m "Renommage d'un fichier"  
Deleting credits.htm  
Adding merci.htm  
Committed revision 41.
```



Subversion en ligne de commande

- ▶ **Résolution des conflits (1/5)**
- ▶ Les conflits peuvent intervenir au moment d'un *update*, lorsque des modifications ont été faites à la fois dans la copie de travail et dans le dépôt.
- ▶ Par exemple, si vous éditez en local un fichier pour lui rajouter une ligne, et qu'un autre utilisateur du dépôt a « *commité* » entre temps une modification différente sur le même fichier, votre *commit* va générer l'erreur suivante :

```
$ svn commit
Sending merci.htm
svn: Commit failed (details follow):
svn:Your file or directory 'merci.htm' is probably out-of-date
svn:
The version resource does not correspond to the resource within the transaction.
Either the requested version resource is out of date (needs to be updated), or the
requested version resource is newer than the transaction root (restart the commit).
```



Subversion en ligne de commande

- ▶ **Résolution des conflits (2/5)**
- ▶ Il vous faut alors effectuer un *update*, ce qui va mettre en concurrence les deux versions du ou des fichiers concernés. Deux cas de figure peuvent alors se présenter.

Dans le premier cas, le conflit peut être résolu automatiquement par Subversion car les modifications ne concernent pas les mêmes parties du fichier. Dans ce cas vous obtiendrez le message suivant :

```
$ svn update
G merci.htm
Updated to revision 42.
```



Subversion en ligne de commande

- ▶ **Résolution des conflits (3/5)**
- ▶ Il est quand même conseillé de vérifier manuellement le résultat de cette résolution « automatique ».

Dans le deuxième cas, les modifications ne peuvent être fusionnées automatiquement car elles concernent les mêmes parties d'un fichier. Dans ce cas un conflit est signalé lors de l'*update* :

```
$ svn update  
C merci.htm  
Updated to revision 43.
```



Subversion en ligne de commande

- ▶ **Résolution des conflits (4/5)**
- ▶ Dans ce cas, deux nouveaux fichiers font leur apparition dans votre copie de travail. Dans l'exemple précédent, on se retrouve avec :`merci.htm.mine` : copie du fichier tel qu'il se trouvait dans votre copie de travail, en local, avant de faire l'*update*. C'est la version que vous souhaitiez « commiter » avant de détecter un conflit ;
- ▶ `merci.htm.r42` : version du fichier pour la révision 42, c'est à dire lors de votre dernier *update*. C'est la version qui a servi de base pour les deux utilisateurs du dépôt qui ont travaillé en parallèle ;
- ▶ `merci.htm.r43` : version du fichier pour la revision 43, c'est à dire la version actuellement dans le dépôt. Il s'agit de la version modifiée par un autre utilisateur, « commitée » avant votre *update*, et dont le contenu est à l'origine du conflit.
- ▶ `merci.htm` : il s'agit d'une version qui, en quelque sorte « résume » les trois autres en faisant apparaître les différences entre versions au sein d'un seul fichier.



Subversion en ligne de commande

- ▶ **Résolution des conflits (5/5)**
- ▶ Dès lors, le travail consiste à éditer le fichier `merci.htm` jusqu'à ce que le conflit soit résolu⁴. Une fois ce travail terminé, on signale que le conflit est résolu à l'aide de la commande `svn resolved` :

```
$ svn resolved merci.htm  
Resolved conflicted state of 'merci.htm'
```

- ▶ On peut alors effectuer le *commit* final.



Subversion en ligne de commande

- ▶ **Protéger la copie de travail**
- ▶ Pour éviter les conflits, il s'agira de **verrouiller** un fichier pendant que vous l'utiliserez, pour être certain que personne ne sera en mesure de le modifier à son tour

```
$ svn lock merci.htm  
'merci.htm' locked by user 'jbarnier '.
```

- ▶ À partir de ce moment, n'importe qui demandant un *lock* sur ce fichier se verra gratifié d'un majestueux :

```
svn: warning:  
Path '/merci.htm' is already locked by user 'jbarnier' in filesystem '/home/jbarnier/svn/repository/db'
```

- ▶ Surtout, ne pas oublier de **déverrouiller** le fichier une fois qu'on a fini de travailler dessus, grâce au sous-programme **unlock** :



Subversion en ligne de commande

- ▶ **Informations sur la copie de travail**
- ▶ Pour obtenir des informations sur la copie de travail en cours, on peut utiliser `svn info` :

```
$ svn info
Path: .
URL: https://cens-srv-dev1.ens-lsh.fr/svnrep/formationsvn
Repository UUID: 090fa6ab-88f7-0310-b83e-cd111ae4905a
Revision: 40
Node Kind: directory
Schedule: normal Last Changed Author: jbarnier
Last Changed Rev: 40
Last Changed Date: 2005-05-30 14:58:11 +0200 (lun, 30 mai 2005)
```



Subversion en ligne de commande

- ▶ **Voir l'historique des modifications d'un fichier ou projet**
- ▶ La commande `svn log` permet d'afficher l'historique de toutes les modifications d'un fichier donné en paramètre ou d'un projet entier :

```
$ svn log index.htm
```

```
-----  
r37 | jbarrier | 2005-05-30 14:42:26 +0200 (lun, 30 mai 2005) | 2 lines  
Modification du titre de la page
```

```
-----  
r36 | jbarrier | 2005-05-30 14:34:05 +0200 (lun, 30 mai 2005) | 2 lines  
Ajout de deux fichiers de test.  
-----
```



Subversion en ligne de commande

- ▶ **Voir le statut de la copie de travail**
- ▶ La commande `svn status` permet d'avoir des informations sur l'état de la copie de travail depuis le dernier *update* :

```
$ svn status -v
?                                credits.htm
      40 40      jbarrier      .
      41 41      jbarrier      merci.htm
      40 37      jbarrier      index.htm
```



Subversion en ligne de commande

- ▶ **Parcourir le dépôt**
- ▶ La commande `svn list` permet d'afficher le contenu du dépôt à distance :

```
$ svn list https://cens-srv-dev.l.ens-lsh.fr/svnrep/formationsvn  
index.htm  
merci.htm
```



Subversion en ligne de commande

- ▶ **Parcourir le dépôt**
- ▶ La commande `svn list` permet d'afficher le contenu du dépôt à distance :

```
$ svn list https://cens-srv-dev.l.ens-lsh.fr/svnrep/formationsvn  
index.htm  
merci.htm
```



Subversion en ligne de commande

- ▶ **Différence entre deux fichiers (1/2)**
- ▶ La commande `svn diff` permet d'afficher les différences entre deux versions de fichiers

```
svn diff sandwiches.txt
```



Subversion en ligne de commande

- ▶ **Différence entre deux fichiers (2/2)**
- ▶ je vois qu'entre le fichier de la révision 6 et ma copie locale, il y a eu une ligne ajoutée (le signe + en début de ligne) disant *1 pot de mayonnaise*.
- ▶ Il va sans dire qu'un signe - indique un retrait de ligne.
- ▶ Je signale au passage que, puisque l'unité est la ligne, la correction d'un caractère provoquera la suppression de l'ancienne ligne et l'ajout de la ligne corrigée !

Index: sandwiches.txt

=====

--- sandwiches.txt	(revision 6)
+++ sandwiches.txt	(working copy)
7 tranches de jambon	
3 tomates	
9 cornichons	
+1 pot de mayonnaise	

Subversion en ligne de commande

- ▶ **Troncs, branches, tags**
- ▶ les arborescences des répertoires de projet contiennent souvent comme premier niveau de sous-répertoires les dossiers **trunk**, **branches** et **tags**.
- ▶ Le **trunk** est a version centrale de programme
- ▶ Une « **branche** » est en général créée lorsqu'un développement « secondaire » est mis en route
- ▶ La notion de **tags** correspond en partie à celle de *release*



Subversion en ligne de commande

▶ Les tags

- ▶ lorsqu'une release est faite on peut la taguer

```
$ svn copy https://svn.imerir.com/monProjet/trunk  
https://svn.imerir.com/monProjet/tags/0.18.3 -m "0.18.3 release"
```

- ▶ en cas d'erreur on peut supprimer ce tag

```
$ svn delete https://svn.imerir.com/monProjet/tags/0.18.3
```

- ▶ on peut ainsi récupérer une version fonctionnelle avec

```
$ svn co https:// svn.imerir.com/monProjet /tags/0.18.3
```

- ▶ ou (sans les .svn)

```
$ svn export https:// svn.imerir.com/monProjet /tags/0.18.3
```



Recapitulatif des commandes SVN

- | | |
|-------------------|--|
| ▶ svnadmin create | créer un repository |
| ▶ svn checkout | recupérer le contenu d'un repository |
| ▶ svn commit | soumettre des modifications au repository |
| ▶ svn import | importer un projet dans le repository |
| ▶ svn resolved | indiquer que les conflits ont été résolus |
| ▶ svn revert | annuler toute modification locale |
| ▶ svn update | mettre à jour le répertoire local |
| ▶ svn cat | lire le contenu d'un fichier du repository |
| ▶ svn diff | regarder les différences entre des versions du repository |
| ▶ svn info | obtenir des infos sur le réperoire local |
| ▶ svn list | lister le contenu du repository |
| ▶ svn log | voir les messages accompagnant chaque révision du repository |
| ▶ svn status | afficher l'état des fichiers/dossiers du répertoire local |
| ▶ svn add | ajouter un fichier/dossier dans l'arborescence de la base |
| ▶ svn copy | copier des fichiers/dossiers dans l'arborescence du repository |
| ▶ svn delete | supprimer des fichiers/dossiers de l'arborescence du repository |
| ▶ svn mkdir | créer un dossier dans l'arborescence du repository |
| ▶ svn move | déplacer des fichiers/dossiers dans l'arborescence du repository |



Subversion

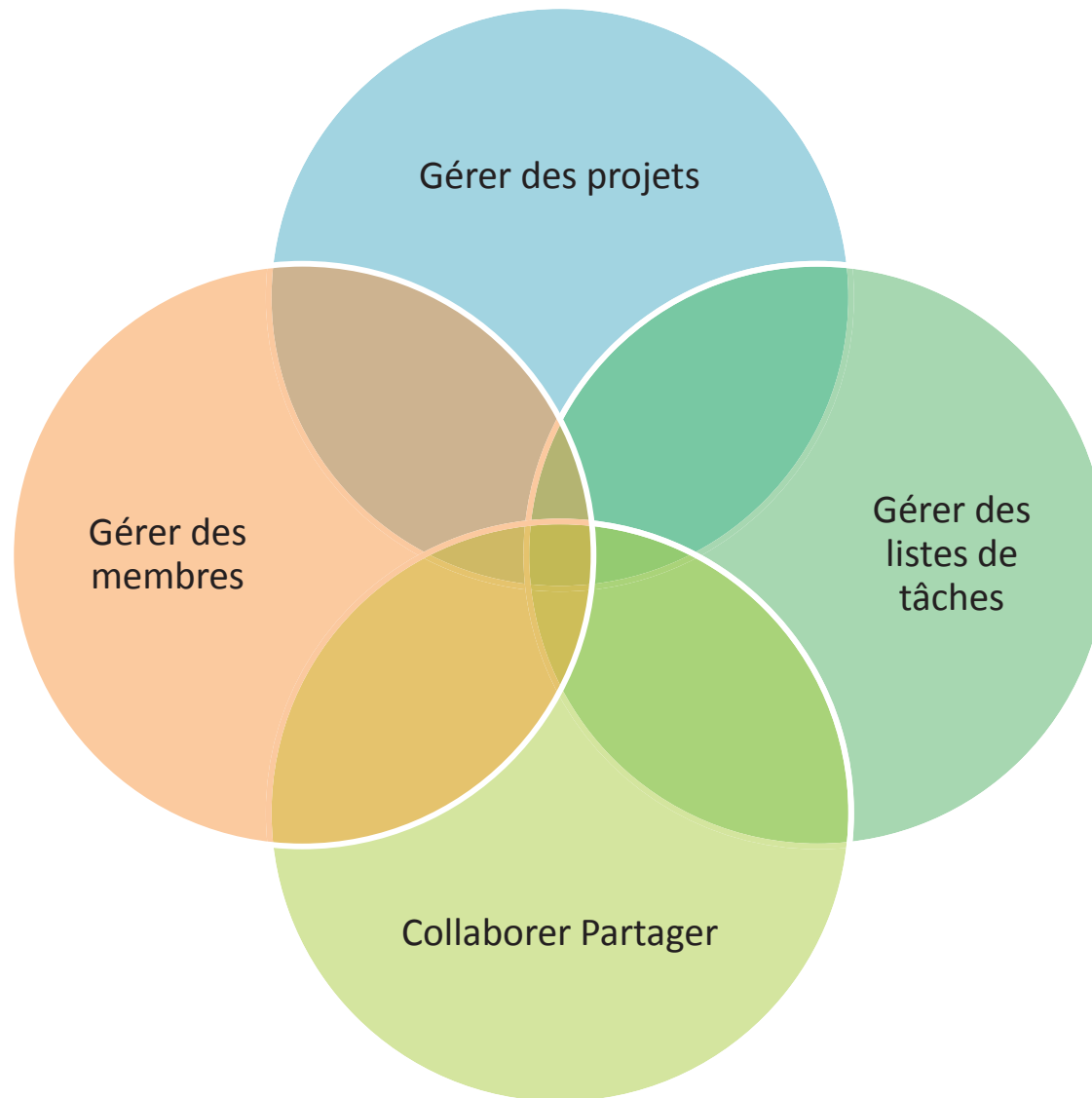
- ▶ Site subversion : <http://subversion.tigris.org/>
- ▶ Visual Suversion windows : <http://www.visualsvn.com/>
- ▶ Tortoise subversion : <http://tortoisesvn.tigris.org/>



Redmine



A quoi sert Redmine ?



Les fonctionnalités de Redmine

- ▶ gestion multi-projets
- ▶ gestion fine des droits utilisateurs définis par des rôles
- ▶ gestion de groupes d'utilisateurs
- ▶ rapports de bogues (bugs), demandes d'évolutions
- ▶ Wiki multi-projets, forums multi-projets
- ▶ news accessibles par RSS / ATOM
- ▶ notifications par courriel (mail)
- ▶ gestion de feuilles de route, GANTT, calendrier, historique
- ▶ intégration avec divers suivis de versions : SVN, CVS, Mercurial
Git, Bazaar & Darcs
- ▶ multilingue (25 langues disponibles pour la 0.7.0)
- ▶ support de plusieurs bases de données : MySQL, PostgreSQL ou SQLite.



Redmine

- ▶ **Créé en 2006 par Jean-Philippe Lang**
- ▶ **OpenSource GPLv2**
- ▶ **90 000 à 300 000** visites mensuelles
- ▶ **15 millions** de pages sur Google
- ▶ **10 000 à 50 000** téléchargement mensuel
- ▶ Gestion de projet « classique » ou « agile »
- ▶ Plus de 200 plugins existants (5 nouveaux par mois)



Redmine

- ▶ **Pourquoi utiliser Redmine ?**
- ▶ **Besoin de partage d'information**
 - ▶ Besoin de partage d'information en général
 - ▶ Similaire wiki
 - ▶ News, forum
 - ▶ Equipes distribuées géographiquement
 - ▶ Gestion de tâches, features, etc.
- ▶ **Contraintes**
 - ▶ Multi-technos (Java, .NET, C++, PL/SQL?)
 - ▶ Intégration avec d'autres outils



Redmine

- ▶ Open source (Ruby on Rails)
- ▶ Similaire à Trac (moins grande base d'utilisateurs Trac pour l'instant)
 - ▶ utilisé par exemple pour Typo3 (l'un des principaux CMS Open source)
- ▶ Initialement reprise des fonctionnalités de Trac
 - ▶ gestion de ticket, wiki, browser repository de code, roadmap, activité, recherche sur l'ensemble des modules, flux RSS
- ▶ Améliorations par rapport à Trac
 - ▶ Multi-projet à la base et notions de projets / sous projet
 - ▶ Administration entièrement web : configuration (champs custom, workflow, droits), création de projets, d'utilisateurs, etc.
 - ▶ News, forum, vision « calendrier »
 - ▶ Meilleure architecture back : intégration native LDAP, meilleure intégration SVN, CVS et autres repo.
- ▶ Communauté très réactive
 - ▶ Plusieurs gestionnaire de source ajoutés rapidement
 - ▶ plugin usine de dev (Continuous Integration) déjà disponible



Tour d'horizon des outils existants

- ▶ Jira (License)
 - ▶ Gestion de tickets (bugs, features, etc.)
 - ▶ Utilisé sur de nombreux projets open source
 - ▶ Customisation avancée en java
- ▶ Mantis
 - ▶ PHP, simple à installer
 - ▶ Champs et workflow customisables
- ▶ Trac
 - ▶ Outil collaboratif avec Wiki, roadmap, activité, tickets, browser code source (subversion et autres)
 - ▶ Champs de formulaire et workflow customisables (V0.11)
 - ▶ Customisation avancée en python
- ▶ Redmine
 - ▶ Fonctionnellement très similaire à trac
meilleure architecture, multi-projet, configuration web
 - ▶ News, releases, forum, changelog, vue calendrier.
 - ▶ Simple à installer et à administrer
 - ▶ Intégration Subversion, CVS



Overview

FLOW3 is an enterprise application framework. It is the foundation of TYPO3 5.x but can also be used independently from the TYPO3 CMS.

- Subproject of: [FLOW3 Packages](#)

Issue tracking

- [Bug](#): 1 open on 35
- [Feature](#): 30 open on 52
- [Major Feature](#): 12 open on 12
- [Task](#): 0 open on 6

[View all issues](#)

Recent activities

Monday 7

[Revision 701](#): * Added remove method to F3_FLOW3_Cache_VariableCache and F3_FLOW3_Cache_AbstractCa...
sebastian

Sunday 6

[Feature #454](#): processor_crap should provide at least all TYPO3v4 options
Jochen Rau

Members

Lead Developer



[Robert Lemke](#) (robert)
@,***

Developer



[Karsten Dambekalns](#) (k-fish)
@,***



[Andreas Förthner](#) (andi)
@,***



[Christian Jul Jensen](#) (julle)
@,***



[Sebastian Kurfuerst](#) (sebastian)
@,***

Planning

[Calendar](#) | [Gantt](#)

- ▶ Multi-projet
- ▶ Roadmap,
- ▶ News

Redmine : Demandes

Home My page Projects Administration Help

Logged in as guillaume.tardif My account Sign out

TestProject

Search: Jump to a project...

Overview Activity Roadmap **Issues** New issue News Documents Wiki Files Repository Settings

Issues

Filters

☒ Status

Add filter:

✓ #	Tracker	Status	Priority	Subject	Assigned to	Updated
<input checked="" type="checkbox"/> 7	Bug	New	Normal	Another new Feature		05/04/2008 05:44 PM
<input type="checkbox"/> 6	Featu	<div><div>Edit</div><div>Status</div><div>Priority</div><div>Assigned to</div><div>% Done</div><div>Copy</div><div>Move</div><div>Delete</div></div>	Normal	New Feature		05/04/2008 05:44 PM
<input type="checkbox"/> 3	Featu				Guillaume Tardif	04/10/2008 11:29 AM
<input type="checkbox"/> 2	Featu				Guillaume Tardif	05/04/2008 05:41 PM

(1-4/4) | Per page:

Also available in: [Atom](#) | [CSV](#) | [PDF](#)

Issues

[View all issues](#)
[Summary](#)
[Change log](#)

Custom queries

[Assigned to me](#)
[V1.0](#)

- ▶ Filtre modifiable simplement
- ▶ Édition des tickets en direct / multi-édition (utile par exemple pour prioriser ou assigner les tickets aux versions)
- ▶ Possibilité de sauvegarder des filtres (perso ou publique)
- ▶ Possibilité de déplacer les tickets vers un autre tracker / projet
- ▶ Flux RSS / exports

Redmine : Demandes

[Overview](#) [Download](#) [Activity](#) [Roadmap](#) [Issues](#) [New issue](#) [News](#) [Wiki](#) [Forums](#) [Repository](#)

Defect #663 [Update](#) [Watch](#) [Copy](#)

homepage link in project settings is too short
Added by Stuart George [128 days ago](#). Updated [22 days ago](#).

Status :	Closed	Start :	2008-02-14
Priority :	Low	Due date :	
Assigned to :	-	% Done :	<div></div> 0%
Category :	Projects		
Target version :	0.8		
Affected version :	devel	Resolution :	Fixed

Description [Comment](#)
When setting a url for a project, it seems to be very short
I had a url that was 69 characters and was too long. I would suggest bumping it to 128 or something.

Related issues
duplicated by [Defect #1095](#) Homepage length limit for Projects Closed 2008-04-23

History
2008-05-25 15:35 - Jean-Philippe Lang [#1](#)

- **Status** changed from *New* to *Closed*
- **Target version** set to *0.7.2*
- **Resolution** set to *Fixed*

Fixed in [r1457](#) (migration required). Limit is now 255 chars.

Associated revisions
[Revision 1457](#)
Added by jplang [26 days ago](#)
Change projects homepage limit to 255 chars ([#663](#), [#1095](#)).

2008-05-29 19:59 - Jean-Philippe Lang [#2](#)

- **Target version** changed from *0.7.2* to *0.8*

Also available in: [Atom](#) | [PDF](#)

- ▶ Description wiki des tickets
- ▶ Historique de toutes les modifications
- ▶ Tickets liés, dupliqués
- ▶ Commits liés au ticket, fermeture automatique des tickets (mots clefs dans le commentaire de commit)
- ▶ Notification mail (« watch ») ou flux RSS

Redmine : Forum, Annonces

TestProject Search: Jump to a project... ▼

[Overview](#) [Activity](#) [Roadmap](#) [Issues](#) [New issue](#) [News](#) [Documents](#) [Wiki](#) **Forums** [Files](#) [Repository](#) [Settings](#)

Forums »

test forum 📩 New message ★ [Watch](#)

Subject	Author	Created	Replies	Last message ▼
test message	Guillaume Tardif	06/21/2008 02:01 PM	0	

(1-1/1) | Per page: 25, 50, 100

- ▶ Spécifique aux projets (possibilité de voir les news de tous les projets, comme l'activité générale transverse)
- ▶ Notifications par flux RSS ou mail
 - ▶ News : notification mail générale ou non
 - ▶ Forum : possibilité de s'abonner à un forum (watch)



Redmine : Ma Page

Redmine

Recherche: Aller à un projet... ▼

Ma page



Personnaliser cette page

Calendrier

	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
	7	8	9	10	11	12	13
15	<div>♦ Sfer - Assistance #3: Install ANT</div> <div>→ Sfer - Assistance #2: Install redmine as a service</div>		<div>♦ Sfer - Assistance #2: Install redmine as a service</div> <div>Début: 07/04/2008</div> <div>Date d'échéance: 09/04/2008</div> <div>Assigné à: Guillaume Tardif</div> <div>Priorité: Normal</div>				

Temps passé (les 7 derniers jours)

Total: 0.50

Activité	Projet	Commentaire	Heures
Aujourd'Hui			
Conception	Sfer - Assistance #3		0.50  

Demandes soumises

#	Tracker	Sujet
3	Sfer - Assistance Fermé - 07/04/2008 16:25	Install ANT
2	Sfer - Assistance Nouveau - 07/04/2008 10:29	Install redmine as a service

Voir toutes les demandes

Demandes surveillées

Aucune donnée à afficher

Demandes qui me sont assignées

#	Tracker	Sujet
2	Sfer - Assistance Nouveau - 07/04/2008 10:29	Install redmine as a service

Voir toutes les demandes

Dernières annonces

- ▶ Page d'accueil et vision « calendrier »
- ▶ Multi-projet
- ▶ personnalisable

Redmine : Roadmap

[Home](#) [My page](#) [Projects](#) [Administration](#) [Help](#)

Logged in as guillaume.tardif [My account](#) [Sign out](#)

TestProject

Search: [Jump to a project...](#)

[Overview](#) [Activity](#) [Roadmap](#) [Issues](#) [New issue](#) [News](#) [Documents](#) [Wiki](#) [Files](#) [Repository](#) [Settings](#)

Roadmap



V1.0

71 days late (04/11/2008)

initial version



3 closed (50%) 3 open (50%)

Related issues

- [Bug #4](#): feature 4
- [Bug #5](#): test
- [Bug #7](#): Another new Feature
- [Feature #1](#): Test feature 1
- [Feature #2](#): Test feature 2
- [Feature #6](#): New Feature



V2.0

52 days late (04/30/2008)

sample version



0 closed (0%) 1 open (100%)

Related issues

- [Feature #3](#): Test feature 3

Roadmap

- ☒ Bug
- ☒ Feature
- ☒ Support

☐ Show completed versions

[Apply](#)

Versions

V1.0
V2.0

Administration









- ▶ A installer
 - ▶ Ruby on Rails (ou instantRails qui inclue MySQL)
 - ▶ My SQL
 - ▶ Redmine
- ▶ Installation Redmine très simple
 - ▶ Création d'un schema redmine dans la BDD
 - ▶ 2 lignes de commandes
 - ▶ préparation de la BDD
 - ▶ Lancement serveur



Redmine : Administration

Redmine

Administration

-  Projects | New
-  Users | New
-  Roles and permissions
-  Trackers | Issue statuses | Workflow
-  Custom fields
-  Enumerations
-  Settings
-  Information

- ▶ Transverse projets
- ▶ Utilisateurs : possibilité de création automatique et / ou lien LDAP
- ▶ Roles, trackers, workflow, champs customs tickets : définis de façon transverse (utilisable ensuite par projet)
- ▶ customisation des notifications par mail



Redmine : Administration

Redmine

Workflow

Select a role and a tracker to edit the workflow:

Role: Tracker:

Current status	New statuses allowed					
	Nouveau	Assigné	Résolu	Commentaire	Fermé	Rejeté
Nouveau	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Assigné	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Résolu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Commentaire	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fermé	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Rejeté	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[Check all](#) | [Uncheck all](#)

Redmine

New custom field (Issues)

Name *	<input type="text"/>
Format	<input type="text" value="Text"/>
Min - Max length	<input type="text" value="0"/> - <input type="text" value="0"/> (0 means no restriction)
Regular expression	<input type="text"/> (eg. ^[A-Z0-9]+\$)
Default value	<input type="text"/>

Trackers

☐ Anomalie ☐ Evolution ☐ Assistance

Required ☐

For all projects ☐

Used as a filter ☐

Searchable ☐

Redmine : Création d'un projet

New project

Name *

30 characters maximum.

Subproject of

Description

B **I** **U** **S** **C** **H1** **H2** **H3**   **pre**  

Identifier *

*Length between 3 and 20 characters. Lower case letters (a-z), numbers and dashes allowed.
Once saved, the identifier can not be changed.*

Homepage

Public ☒

Trackers

☒ Bug ☒ Feature ☒ Support

Modules

☒ Issue tracking ☒ Time tracking ☒ News ☒ Documents ☒ Files ☒ Wiki
☒ Repository ☒ Boards



Redmine : Administration d'un projet

TestProject

Overview Activity Roadmap Issues New issue News Documents Wiki Files Repository **Settings**

Settings

Information Modules Members Versions Issue categories Wiki **Repository** Forums

User	Role
Guillaume Tardif	Manager <input type="button" value="Change"/>
Toto Titi	Developer <input type="button" value="Change"/>

New member

Role:

- ▶ Modification des informations de base et modules actifs
- ▶ Ajout d'utilisateurs au projet et rôles des users
- ▶ Ajout de version (jalons) / catégories de tickets
- ▶ Page d'accueil wiki
- ▶ Lien avec un dépôt (SVN, CVS, Mercurial, Git)
- ▶ Création de forums

DOXYGEN



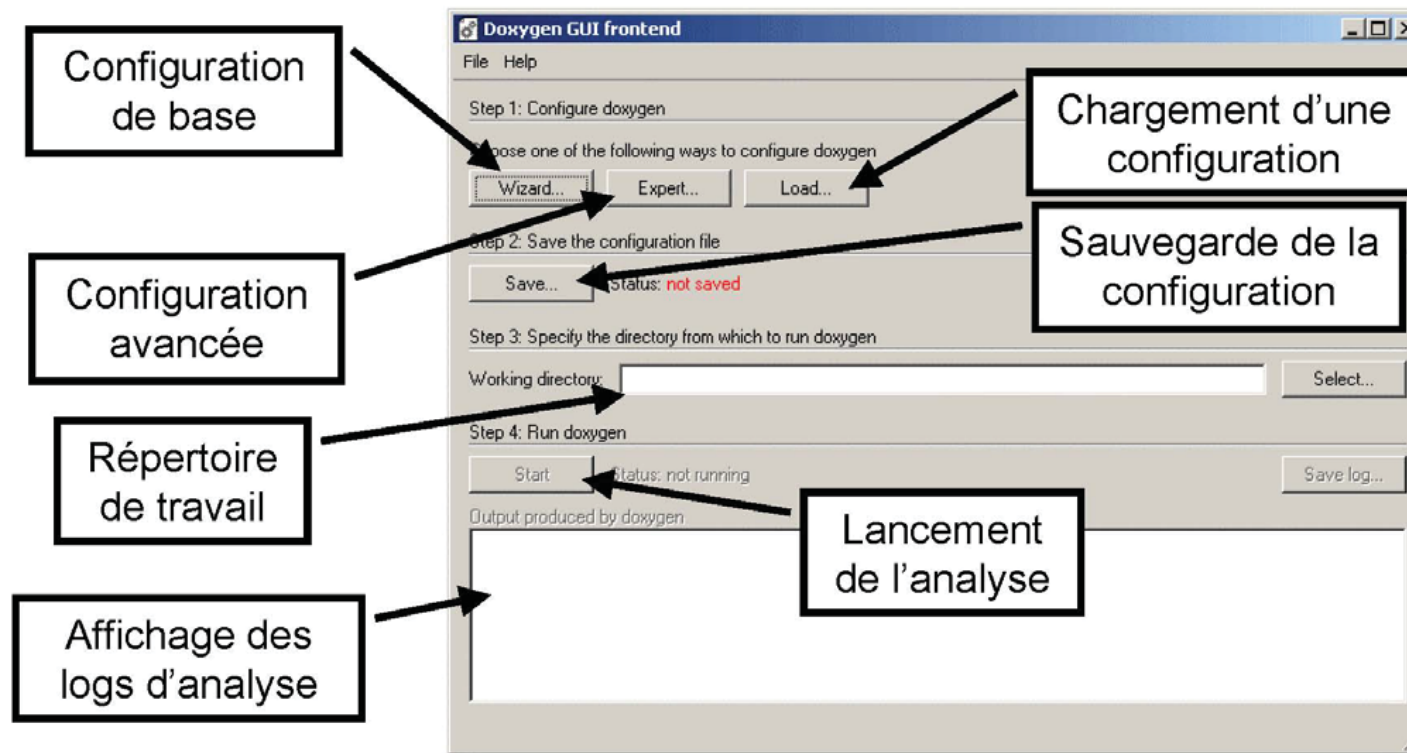
DOXYGEN

- ▶ Cet outil est Open Source et sous licence GNU GPL
- ▶ Il est disponibles sous les plateformes :
 - ▶ - Linux (Debian, RedHat, Suse...)
 - ▶ - Mac OS
 - ▶ - Windows



Prise en main

- Une interface graphique permet de configurer aisément Doxygen et de lancer l'analyse de vos sources pour en produire une documentation.



Configuration

Avant de lancer l'analyse de vos sources, il faut configurer Doxygen. Il existe deux modes de configuration :

- Wizard : configuration de base
- Expert : configuration avancée pour les experts

Vous pouvez laisser les paramètres par défaut.

Vous pouvez enregistrer votre configuration pour la recharger lors d'une analyse ultérieure.



Configuration Wizard

La configuration de base est constituée de 4 onglets :

- Project : informations générales sur votre projet
- Mode : le mode d'analyse
- Output : le type de documentation à générer
- Diagrams : les diagrammes à générer



Configuration Wizard - Project

The screenshot shows the 'doxywizard' window with the 'Project' tab selected. The window title is 'doxywizard'. The tabs are 'Project', 'Mode', 'Output', and 'Diagrams'. The main text says 'Provide some information about the project you are documenting'. There are two text input fields: 'Project name:' with the value 'GICOM' and 'Project version or id:' with the value '1.0'. Below these, the text says 'Specify the directory to scan for source code'. There is a text input field for 'Source code directory:' with the value 'D:/GICOM' and a 'Select...' button. A checkbox labeled 'Scan recursively' is checked. Below this, the text says 'Specify the directory where doxygen should put the generated documentation'. There is a text input field for 'Destination directory:' with the value 'D:/GICOM/doc/' and a 'Select...' button. At the bottom are 'OK' and 'Cancel' buttons. On the left, five French annotations in boxes point to specific fields: 'Nom de votre projet logiciel' points to the Project name field; 'Numéro de version de votre logiciel' points to the Project version or id field; 'Localisation de vos sources' points to the Source code directory field; 'Analyse récursive des sous-répertoires' points to the Scan recursively checkbox; and 'Localisation de la documentation à générer' points to the Destination directory field.

Nom de votre projet logiciel

Numéro de version de votre logiciel

Localisation de vos sources

Analyse récursive des sous-répertoires

Localisation de la documentation à générer

doxywizard

Project Mode Output Diagrams

Provide some information about the project you are documenting

Project name: GICOM

Project version or id: 1.0

Specify the directory to scan for source code

Source code directory: D:/GICOM Select...

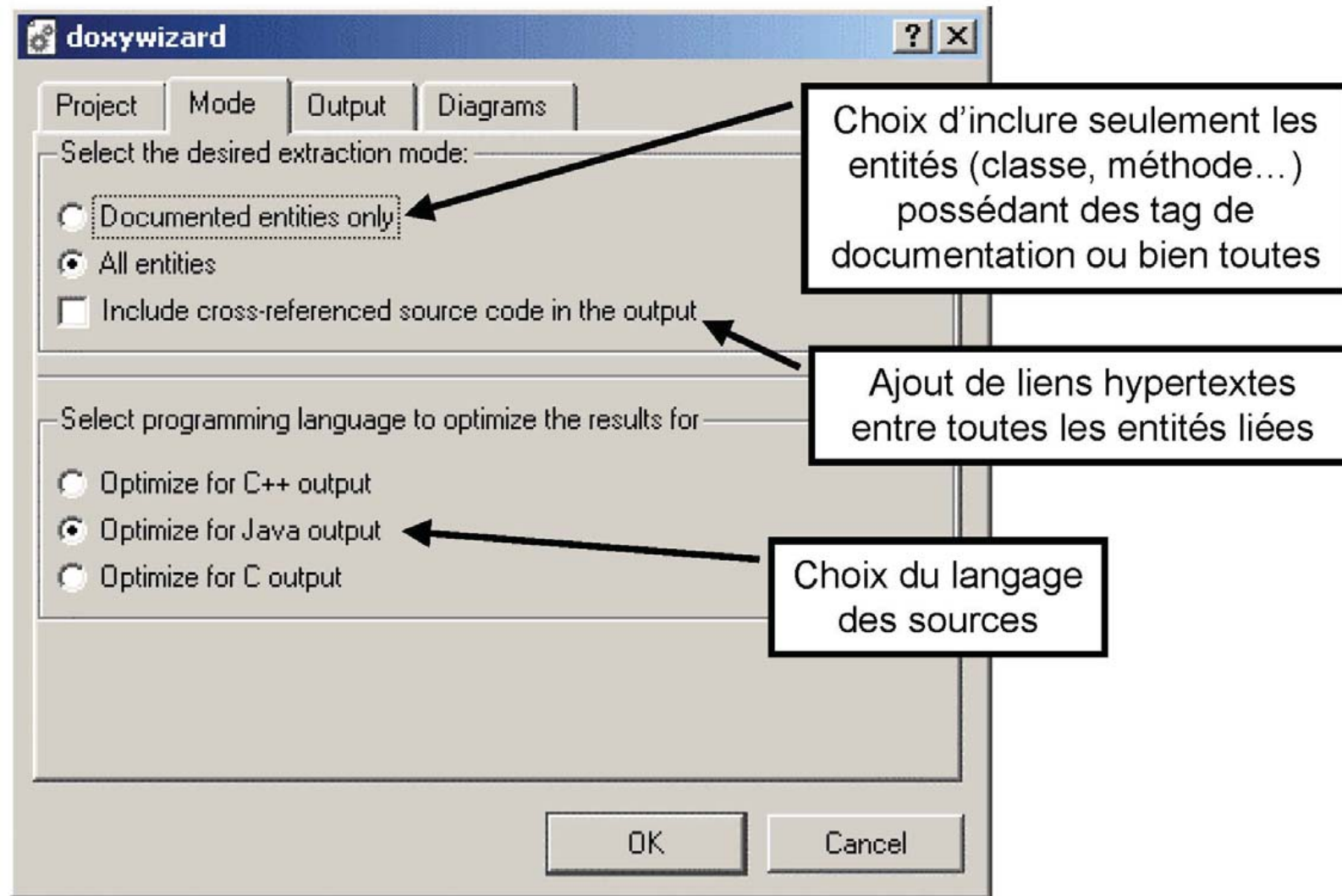
☒ Scan recursively

Specify the directory where doxygen should put the generated documentation

Destination directory: D:/GICOM/doc/ Select...

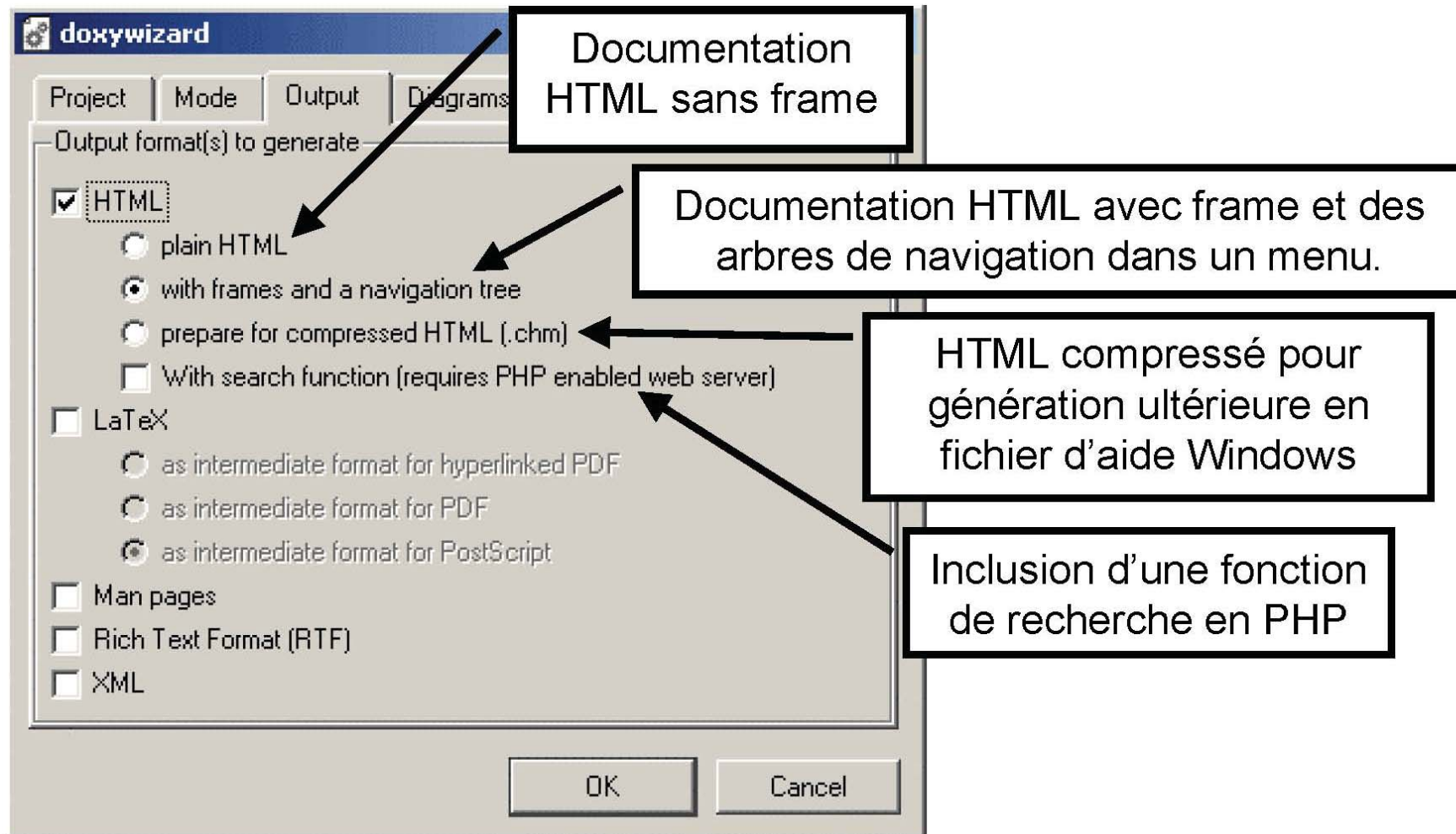
OK Cancel

Configuration Wizard - Mode

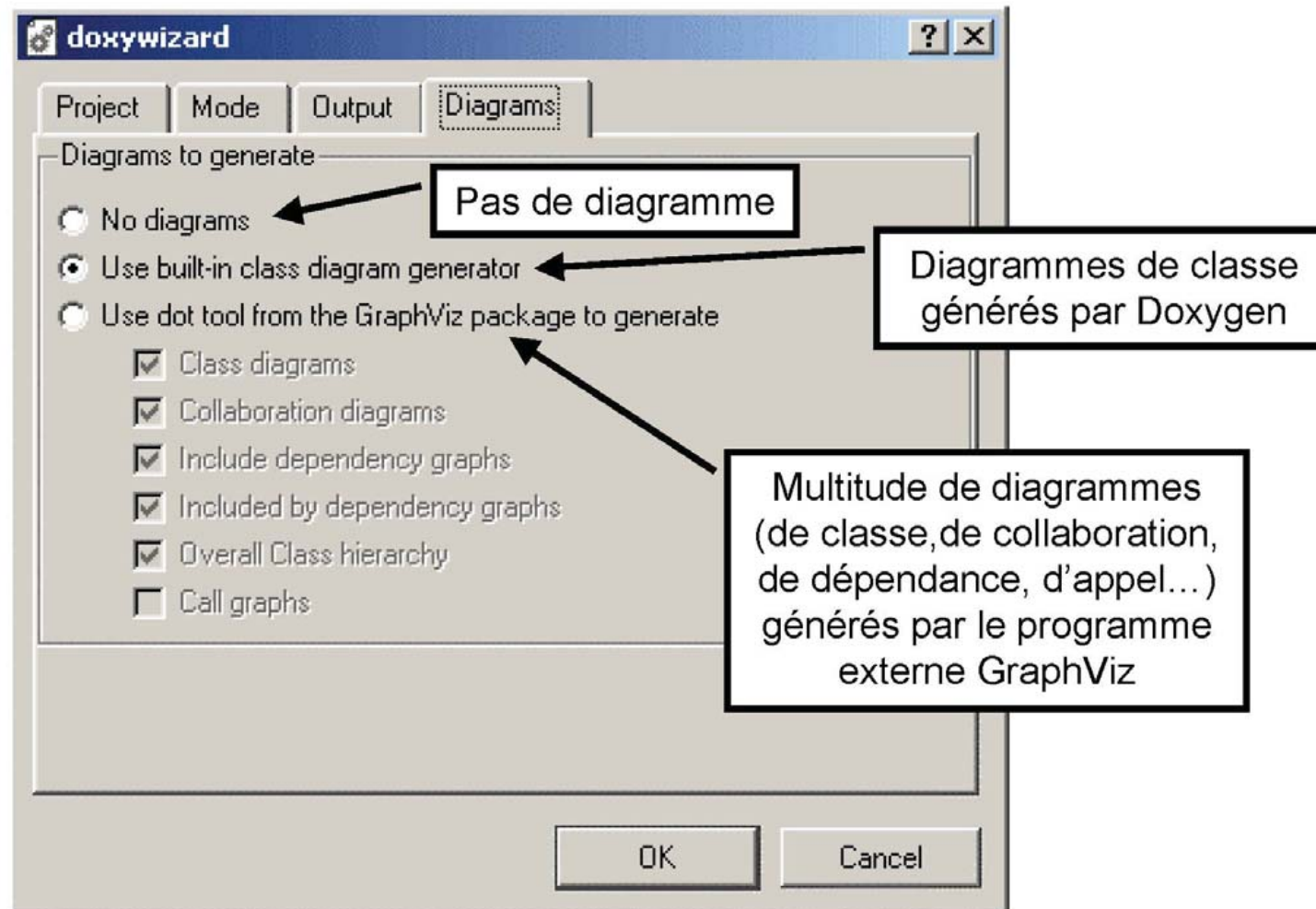


Configuration Wizard -Output

- La documentation générée par Doxygen peut être de type HTML, LaTeX, Man, RTF ou XML



Configuration Wizard - Diagrams



Analyse

- ▶ Le lancement de l'analyse de vos codes sources se fait en cliquant sur « Start ».
- ▶ Le détail des opérations effectuées par Doxygen apparaît dans la zone de texte. Puis s'affiche : « * * * Doxygen has finished » lorsque la génération de la documentation a été réalisée.



Rendu HTML

- ▶ La documentation HTML produite par Doxygen se trouve dans le sous-répertoire html/ du répertoire de sortie spécifié dans la configuration.

The screenshot displays the Doxygen HTML output for the `ConfigOption` class. On the left, a 'Menu de navigation' (Navigation Menu) lists various documentation pages. The main content area, titled 'ConfigOption Class Reference', includes an inheritance diagram showing `ConfigOption` as the base class for `ConfigBool`, `ConfigEnum`, `ConfigInfo`, `ConfigInt`, `ConfigList`, `ConfigObsolete`, and `ConfigString`. Below the diagram, the 'Public Types' section shows an enumeration `OptionType` with a value `MAX_OPTION_LENGTH = 23`. The 'Public Member Functions' section lists a function `OptionType kind() const`. The 'Detailed Description' section provides an abstract base class description and its definition location in `config.h`.

Menu de navigation

Diagramme

Exemple de documentation HTML produite

Commentaires du code source

- ▶ Les commentaires du code source font la documentation générée.
Les commentaires extraits sont les suivants :

```
/**
```

```
* commentaires
```

```
*/
```

```
/*!
```

```
* commentaires
```

```
*/
```

```
/**
```

```
commentaires
```

```
*/
```

```
/// /// commentaires ///
```

```
//! //! commentaires //!
```

```
/////////////////////////////////// /// commentaires //////////////////////////////////
```



Commentaires du code source

- ▶ Les courtes descriptions (à distinguer des descriptions longues normales) affichées sur les pages Packages, Class Hierarchy, Class List et File List sont introduites par le mot clé `\brief`.

```
/*! \brief Voici une courte description  
sur plusieurs lignes. *  
Et ici la description longue normale. */
```

Si le paramètre de configuration `JAVADOC_AUTOBRIEF` est mis à YES, alors le mot clé `\brief` est inutile :

```
/** Voici une courte description  
terminée par un point. *  
Et ici la description longue normale. */
```



Commentaires du code source

Les commentaires Javadoc traditionnels sont interprétés par Doxygen.

Rappel des tags Javadoc les plus courants :

`@author`, `@deprecated`, `@exception`, `@param`, `@return`,
`@see`, `@serial`, `@serialData`, `@serialField`, `@since`,
`@throws`, `@version`

