

Gestion de sources Contrôle de versions

Sylvain GROSDÉMOUGE



Introduction

Intérêt

Historique de versions

Travail collaboratif

Sécurisation

Domaines d'application

Historique

Concepts de base

Concepts avancés

Outils graphiques

Historique de versions :

- Historique des modifications apportées à un projet
- Informations ciblées pour chaque modification (nature, date, auteur, ...)
- Navigation entre les différentes versions des fichiers
- Synchronisation des fichiers à une date antérieure
- Différences entre deux versions de fichiers

Introduction

Intérêt

Historique de versions

Travail collaboratif

Sécurisation

Domaines d'application

Historique

Concepts de base

Concepts avancés

Outils graphiques

Travail collaboratif :

- Permettre à des utilisateurs de travailler ensemble, en centralisant les modifications apportées à un projet
- Permettre de résoudre les conflits lorsque ces modifications portent sur les mêmes fichiers
- Permettre de suivre de manière précise les différentes tâches effectuées par une équipe sur un projet donné
- Permettre à un nouvel utilisateur de connaître l'évolution d'un code source

Introduction

Intérêt

Historique de versions

Travail collaboratif

Sécurisation

Domaines d'application

Historique

Concepts de base

Concepts avancés

Outils graphiques

Sécurisation :

- Récupération de données accidentellement perdues
- Stabilisation de versions
- Itérations rapides

Introduction

Intérêt

Domaines d'application

Historique

Concepts de base

Concepts avancés

Outils graphiques

Domaines d'application variés :

- Programmation (source codes, fichiers de configuration, ...)
- Sauvegardes des données (tout type de données)

Introduction

Historique

Première génération

Seconde génération

Troisième génération

... Et maintenant ?

Concepts de base

Concepts avancés

Outils graphiques

Première génération :

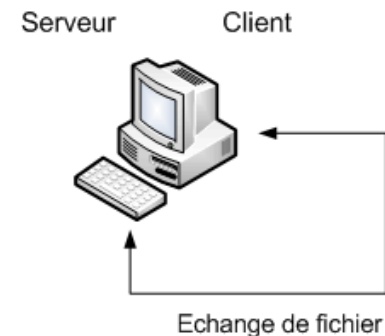
- Mono-utilisateurs
- Base de données locale
- Fichiers textes uniquement

1972 - SCCS (Source Code Control System)

Disponible seulement sur UNIX.

1982 - RCS (Revision Control System).

Premier système multi-plateformes.



Introduction

Historique

Première génération

Seconde génération

Troisième génération

... Et maintenant ?

Concepts de base

Concepts avancés

Outils graphiques

Seconde génération :

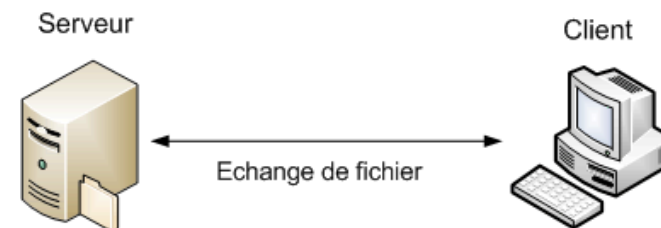
- Multi-Utilisateurs
- Base de données centralisée

1986 - CVS (Concurrent Version Control)

1995 - Perforce

2000 - Subversion

2004 - Microsoft Team Foundation Server



Introduction

Historique

Première génération

Seconde génération

Troisième génération

... Et maintenant ?

Concepts de base

Concepts avancés

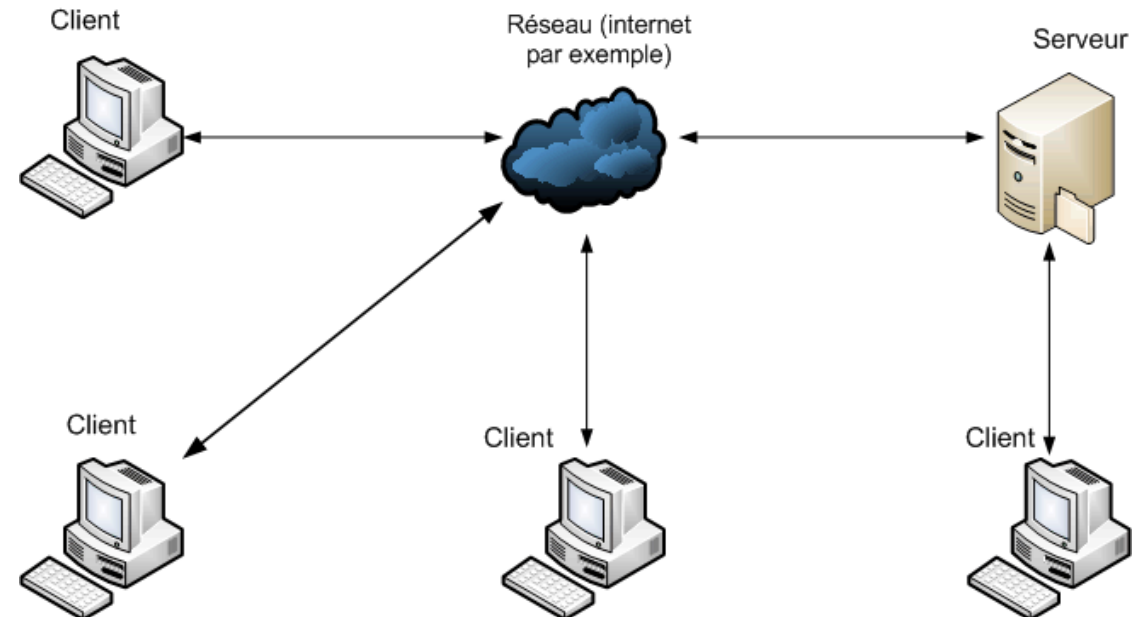
Outils graphiques

Troisième generation :

- Multi-Utilisateurs
- Base de données distribuée

2005 – GIT

2005 - Mercurial



Introduction

Historique

Première génération

Seconde génération

Troisième génération

... Et maintenant ?

Concepts de base

Concepts avancés

Outils graphiques

Et maintenant ?

Perforce, SVN, TFS pour les systèmes centralisés
(éditeurs de logiciels propriétaires)



Ubisoft, Nyse, Disney, nVidia, ...

Git, Mercurial pour les systèmes distribués
(logiciel libre)



git



Introduction

Historique

Concepts de base

Définitions

Dépôt

Projets

Révisions

Copie locale

Cas d'utilisation

Commandes de base

Concepts avancés

Outils graphiques

Notion de dépôt (ou repository)

- Emplacement où sont stockées les données du ou des projet géré
- Accessible à partir d'une URL (locale ou distante)
- Contient l'historique des changement apportés aux différents projets

Projets

- Un ou plusieurs projet par dépôt
- Généralement, un répertoire par projet, qui contient les fichiers du projet :

```
//depot
```

```
  /projet_1  
    main.c  
    ...
```

```
  /projet_2  
    main.c  
    ...
```

Introduction

Historique

Concepts de base

Définitions

Dépôt

Projets

Révisions

Copie locale

Cas d'utilisation

Commandes de base

Concepts avancés

Outils graphiques

Révisions

Contient 4 informations :

- Date
- Numéro de révision
- Utilisateur
- Description (Log)

Une révision pour chaque apport de modifications au dépôt de la part d'un utilisateur.

Introduction

Historique

Concepts de base

Définitions

Dépôt

Projets

Révisions

Copie locale

Cas d'utilisation

Commandes de base

Concepts avancés

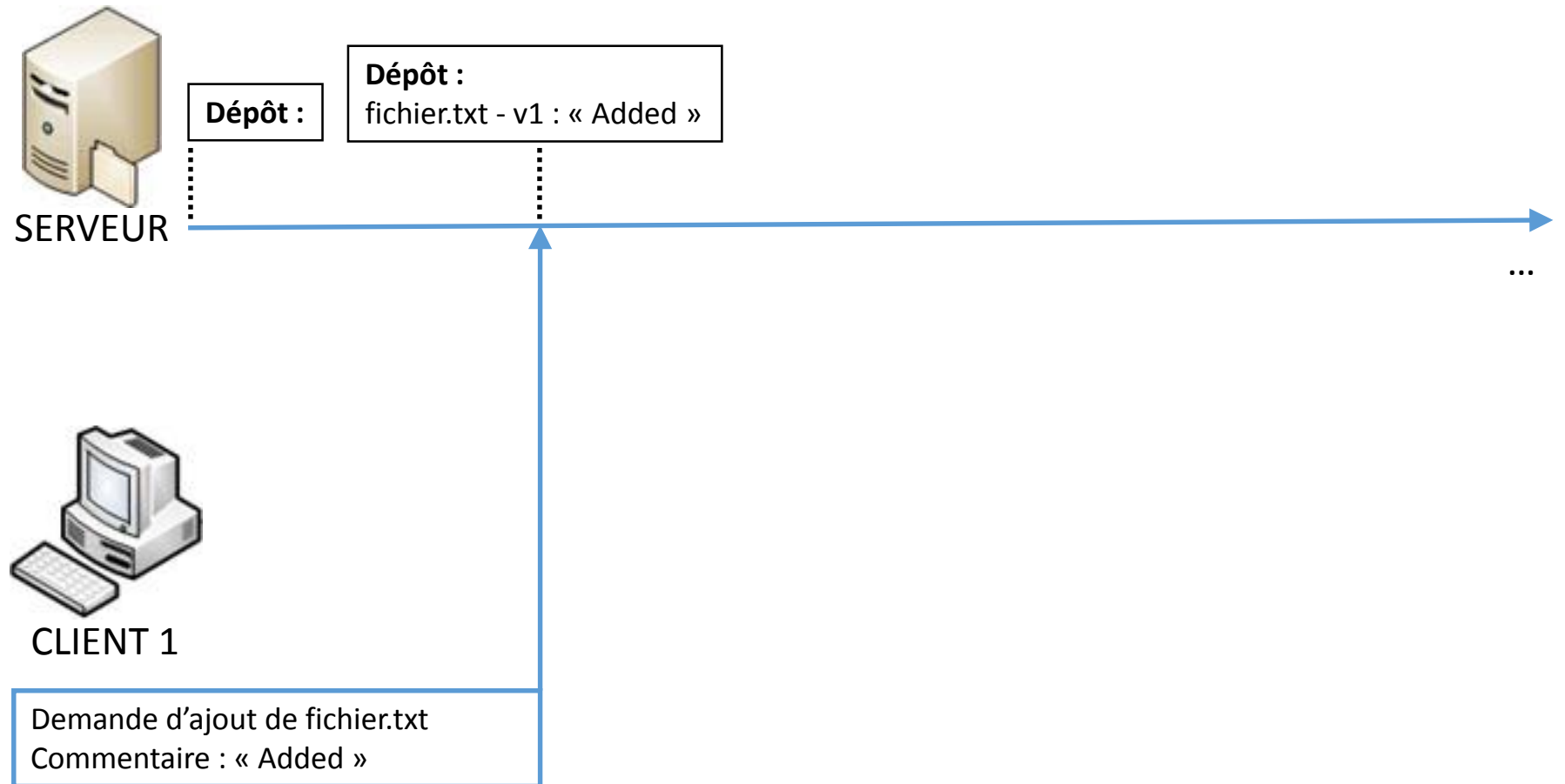
Outils graphiques

Copie locale (*ou* copie de travail, *ou* 'Working Copy')

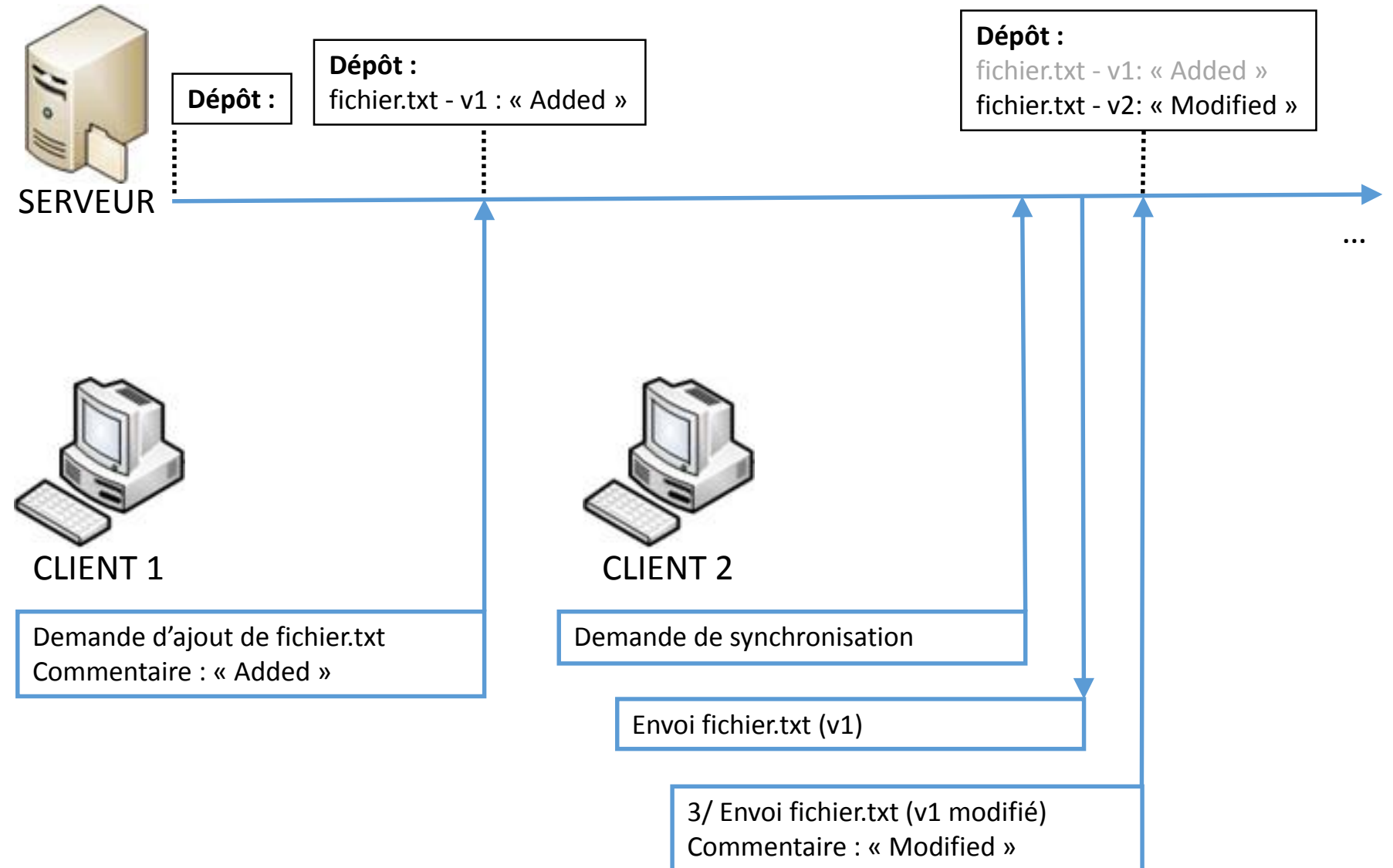
Répertoire situé en local, contenant :

- une copie des fichiers du dépôt à une révision donnée
- Les modifications apportées par l'utilisateur à ces fichiers

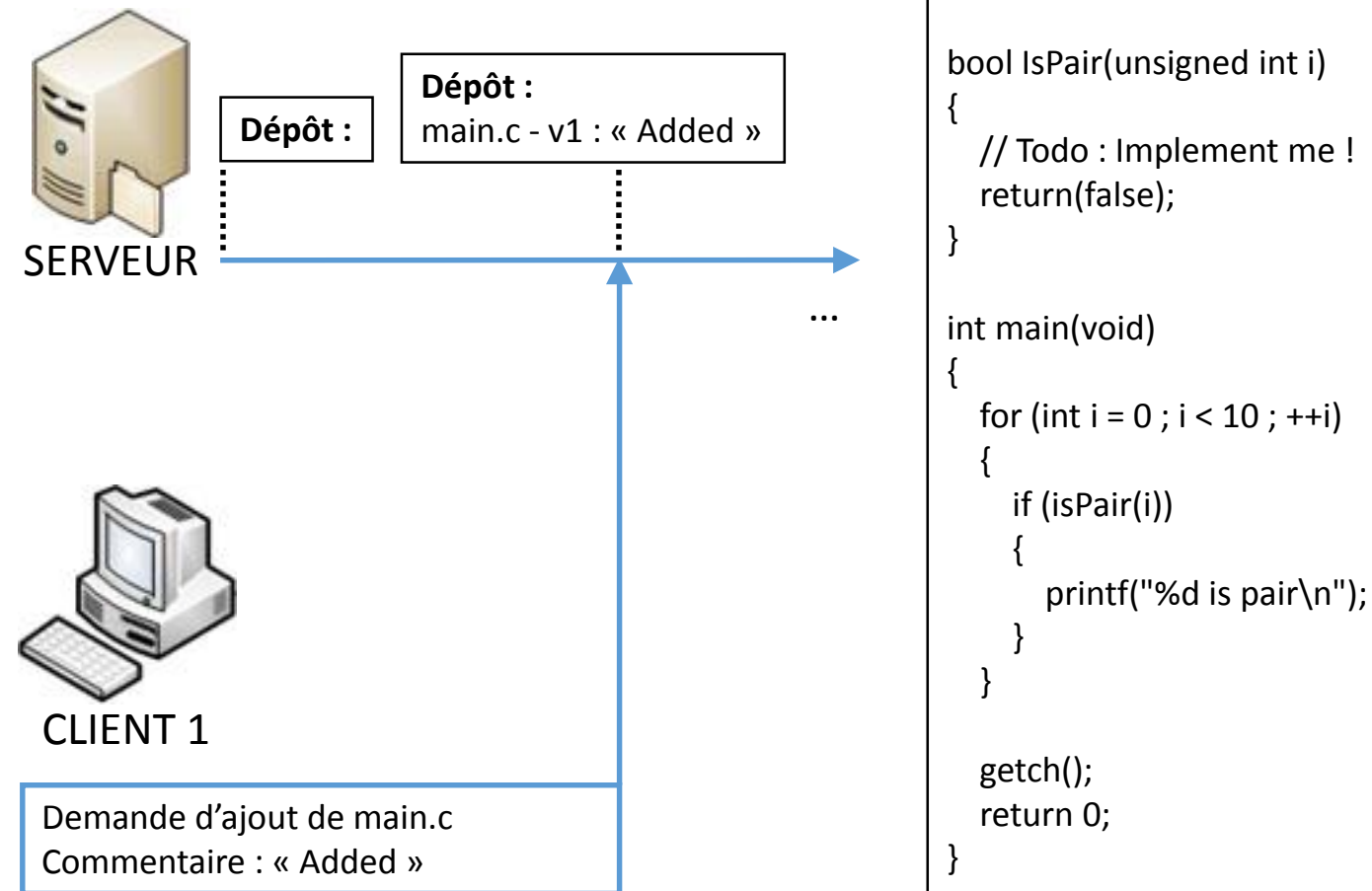
Ajout



Modification

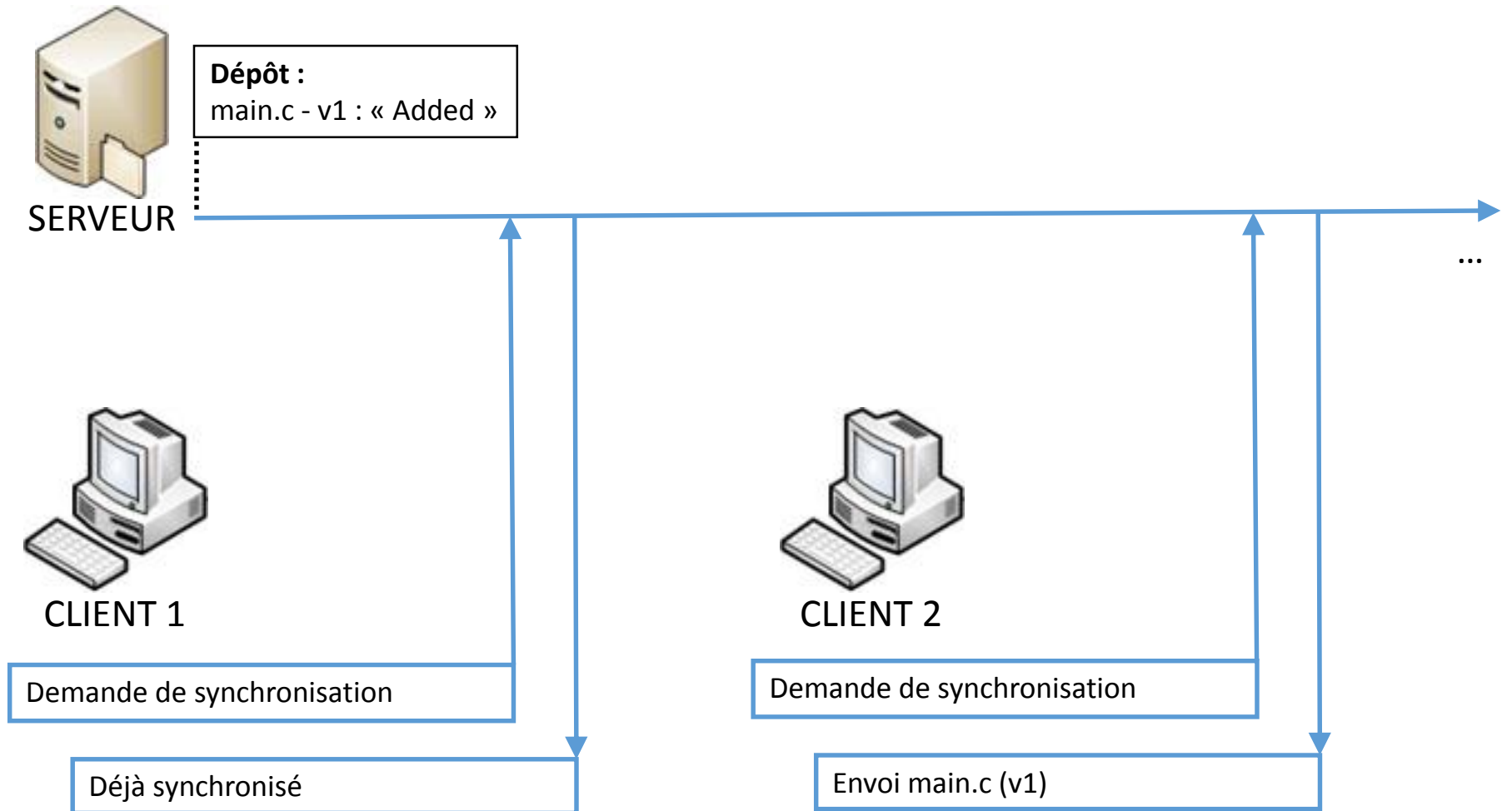


Conflit



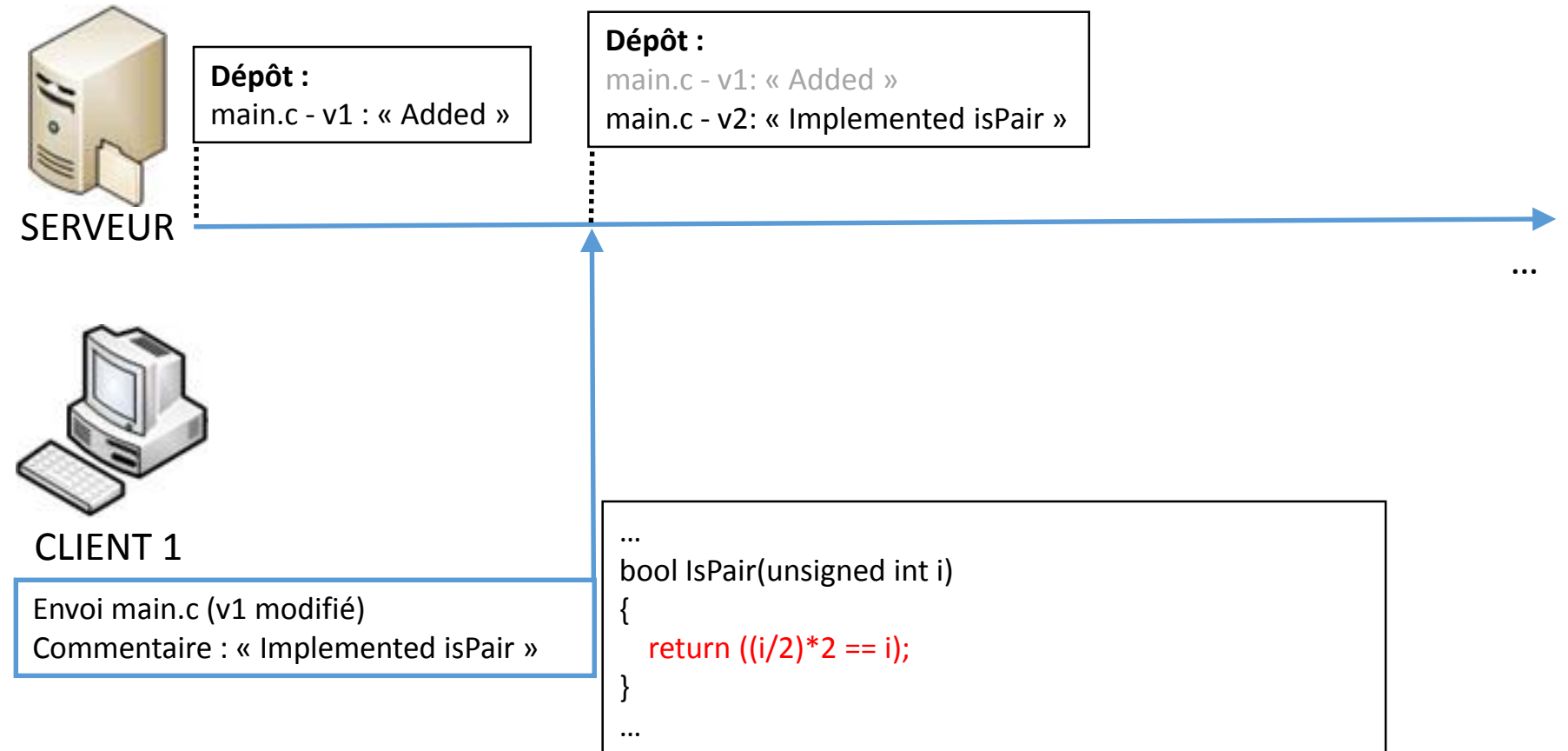
Dépôt d'un fichier 'main.c' par le Client 1, avec une fonction 'isPair' à implémenter

Conflit



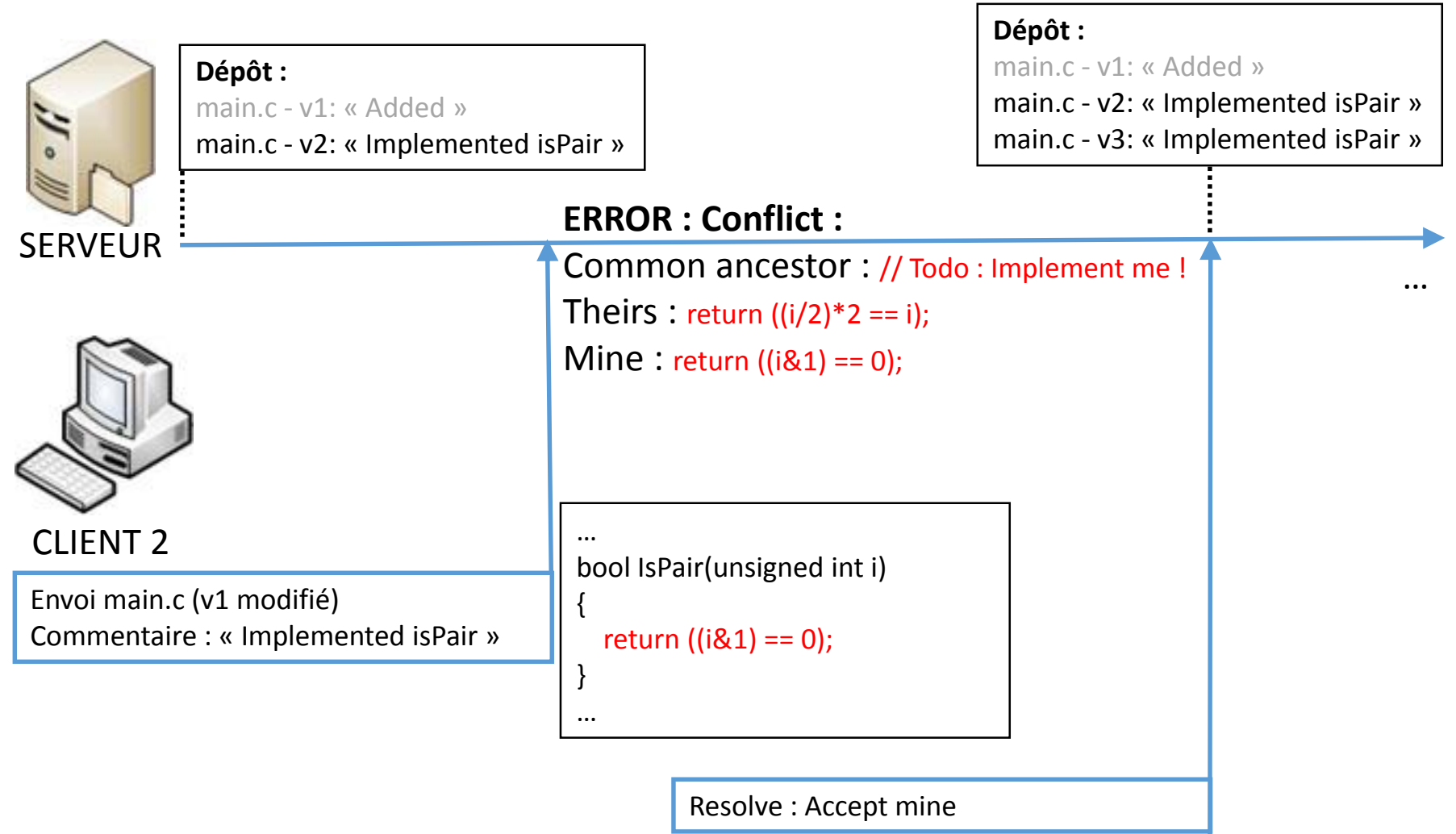
Le Client 1 ET le client 2 se synchronisent à la dernière version

Conflit



Le Client 1 implémente la fonction isPair et la redépose.

Conflit



Le Client 2 implémente lui aussi la fonction isPair, d'une manière différente, et la redépose.

- GESTION DE SOURCES
- CONTRÔLE DE VERSIONS
- Introduction
- Historique
- Concepts de base
 - Définitions
 - Cas d'utilisation
 - Commandes de base
 - Help
 - Checkout
 - Import
 - Update
 - Status
 - Add
 - Revert
 - Diff
 - Commit
 - Resolve
- Concepts avancés
- Outils graphiques

Help

« Affichage de l'aide »

Commande SVN :

Liste des commandes :
`svn help`

Aide relative à une certaine commande :
`svn help <Nom de la commande>`

Introduction

Historique

Concepts de base

Définitions

Cas d'utilisation

Commandes de base

Help

Checkout

Import

Update

Status

Add

Revert

Diff

Commit

Resolve

Concepts avancés

Outils graphiques

Check Out

« Récupérer les fichiers d'un dépôt en local pour la première fois »

Le résultat est une copie locale des fichiers du dépôt.

Commande SVN :

```
svn checkout URL
```

Import

« Placer dans le dépôt des fichiers locaux existants pour y créer un nouveau projet »

Commande SVN :

```
svn import -m "Initial import" . URL
```

Introduction

Historique

Concepts de base

Définitions

Cas d'utilisation

Commandes de base

Help

Checkout

Import

Update

Status

Add

Revert

Diff

Commit

Resolve

Concepts avancés

Outils graphiques

Update

« Récupérer la dernière version des fichiers du dépôt en local »

Commande SVN :

```
svn update <filename1> ... <filenameN>
```

```
svn update
```

Status

« Affiche le status d'un ou plusieurs fichiers locaux »

Commande SVN :

```
svn status <filename>  
svn status <filename1> ... <filenameN>
```

4 colonnes :

- Attribut [A|D|M|C|X|?|!]
- * si le fichier a été modifiée sur le serveur
- Numéro de révision
- Nom du fichier

Exemple :

```
$ svn status test.txt  
A          test.txt (fichier ajouté)
```


Introduction

Historique

Concepts de base

Définitions

Cas d'utilisation

Commandes de base

Help

Checkout

Import

Update

Status

Add

Revert

Diff

Commit

Resolve

Concepts avancés

Outils graphiques

Add

« Ajout d'un fichier dans le dépôt »

Commande SVN :

```
svn add <filename>
```

```
svn commit -m «Added filename» <filename>
```

```
svn add <filename1> ... <filenameN>
```

```
svn commit -m «Added some files» <filename1> ... <filenameN>
```

Revert

« Restaure l'état des fichiers et répertoires en annulant les modifications apportées localement et supprime les marques de conflit »

Commande SVN :

```
svn revert <filename1> ... <filenameN>
```

```
svn revert
```

Introduction

Historique

Concepts de base

Définitions

Cas d'utilisation

Commandes de base

Help

Checkout

Import

Update

Status

Add

Revert

Diff

Commit

Resolve

Concepts avancés

Outils graphiques

Diff

« Afficher les différences entre deux versions de fichiers »

Commande SVN :

```
svn diff <filename>
```

```
svn diff <filename1> ... <filenameN>
```

Introduction

Historique

Concepts de base

Définitions

Cas d'utilisation

Commandes de base

Help

Checkout

Import

Update

Status

Add

Revert

Diff

Commit

Resolve

Concepts avancés

Outils graphiques

Commit

« Mettre à jour le dépôt en fonction de la copie locale »

- Génère une nouvelle révision
- Ne fonctionne que si la copie locale correspond à la dernière version présente sur le dépôt (hors modifications locales).

En cas de modifications apportées au dépôt depuis la dernière update, une nouvelle update est nécessaire pour pouvoir effectuer le commit.
Cette nouvelle update peut engendrer des conflits de versions.

Commande SVN :

```
svn commit -m "My Commit" <filename> ... <filenameN>
```

Introduction

Historique

Concepts de base

Définitions

Cas d'utilisation

Commandes de base

Help

Checkout

Import

Update

Status

Add

Revert

Diff

Commit

Resolve

Concepts avancés

Outils graphiques

Resolve

« Résolution le(s) conflit(s) en utilisant certaines règles »

Commande SVN :

```
svn resolve [options] [path]
```

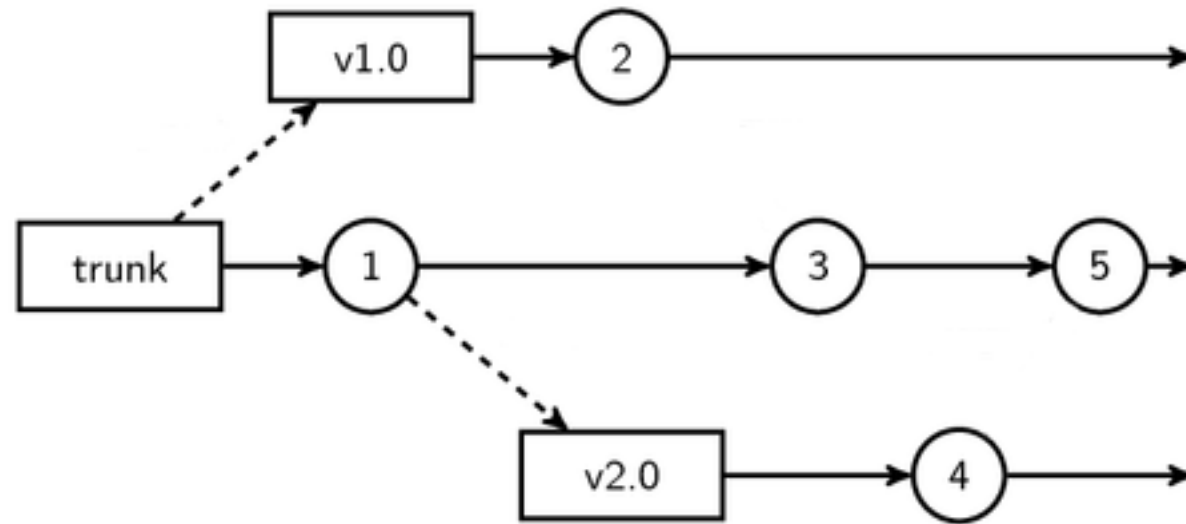
Par exemple :

```
svn resolve --accept working main.c
```

```
svn resolve --accept mine-full main.c
```

```
svn resolve --accept theirs-full main.c
```

Introduction



Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité

avec synchronisation

Versions logicielles

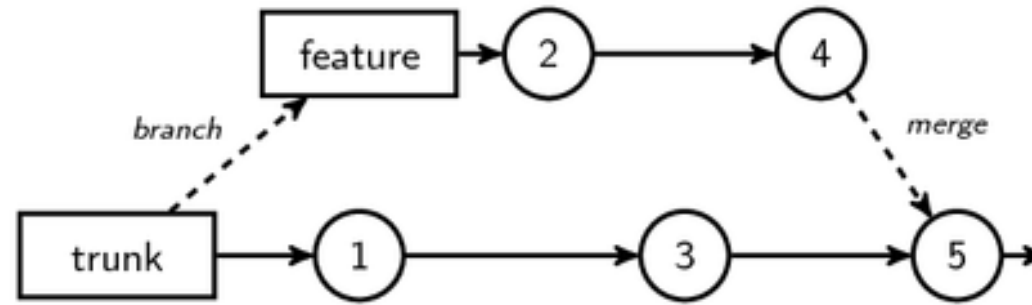
Modèle Base/work/dev

Portage parallèle

Commandes

Outils graphiques

Ajout de fonctionnalités



Un programme existant

Une fonctionnalité 'dangereuse' en termes de stabilité à implementer

Création d'une branche dans laquelle on implémente la fonctionnalité

Intégration une fois la fonctionnalité implémentée et testée

Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité

avec synchronisation

Versions logicielles

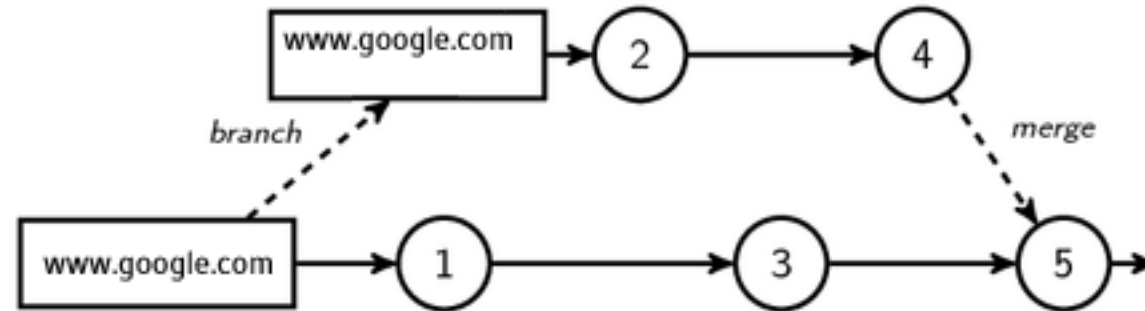
Modèle Base/work/dev

Portage parallèle

Commandes

Outils graphiques

Ajout de fonctionnalité – Exemple concret



Site web existant / contenu évolutif

Site web mobile à développer

Problème : Comment développer le site mobile de manière itérative, sans affecter la stabilité du site existant ?

→ Développement du site mobile dans une branche avant d'intégrer les changements dans le site déployé

Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité

avec synchronisation

Versions logicielles

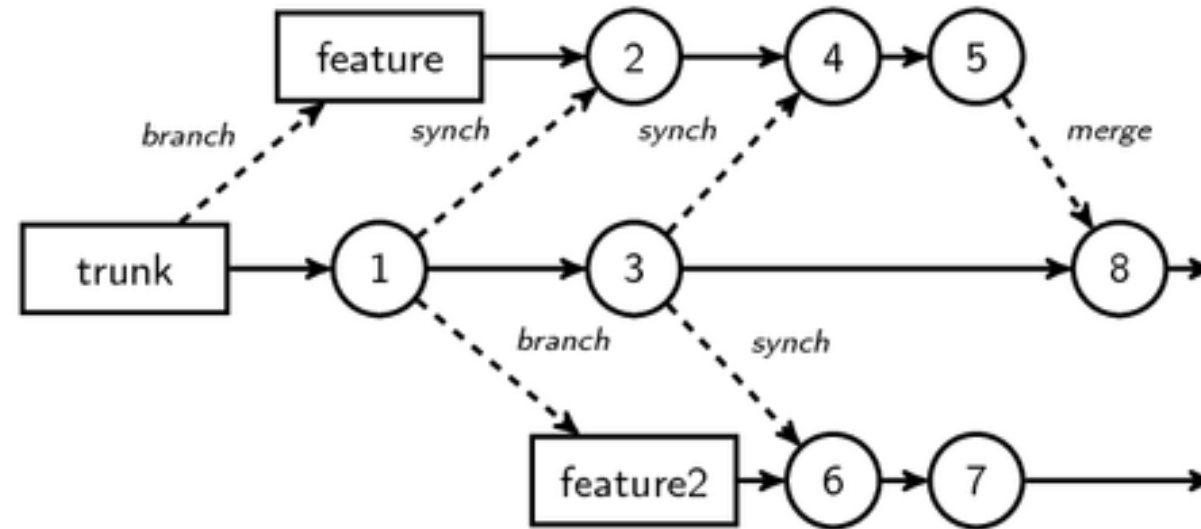
Modèle Base/work/dev

Portage parallèle

Commandes

Outils graphiques

Ajout de fonctionnalités avec synchronisation



Attention : Si le temps de développement du (ou des) fonctionnalité(s) est important, penser à synchroniser de temps en temps...

L'opération de merge n'en sera que plus aisée !

Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité

avec synchronisation

Versions logicielles

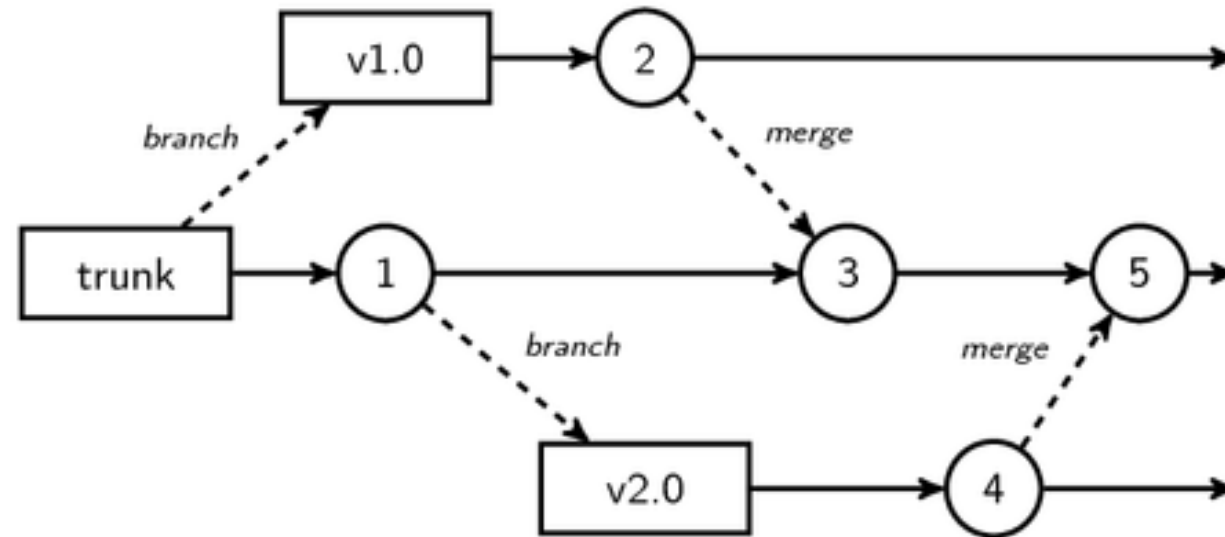
Modèle Base/work/dev

Portage parallèle

Commandes

Outils graphiques

Versions logicielles



Exemple d'un logiciel déployé en v1.0 et v2.0 chez des clients différents, avec un tonc commun.

Par exemple, un logiciel de gestion de stock, déployé chez deux clients qui manipulent des données différentes, et qui ont besoin de fonctionnalités additionnelles par rapport au modules de base.

Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité
avec synchronisation

Versions logicielles

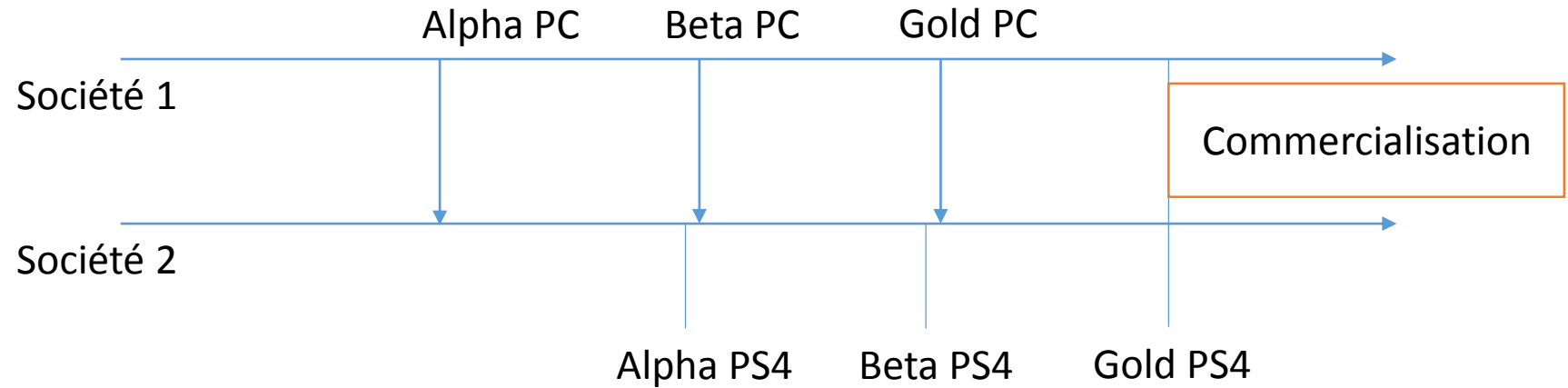
Modèle Base/work/dev

Portage parallèle

Commandes

Outils graphiques

Portage parallèle



Deux sociétés de jeux vidéos :

- La société 1 travaille sur un produit (PC)
- La société 2 travaille sur son portage (PS4)
- Deux branches, qui sont décalées de 2 mois dans le temps

Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité
avec synchronisation

Versions logicielles

Modèle Base/work/dev

Portage parallèle

Commandes

Copy

Merge

Outils graphiques

Copy

« Copie d'un fichier ou d'une arborescence à l'intérieur d'un dépôt »

Commande SVN :

```
svn copy [source] -r ? [destination]
```

Par exemple :

```
svn copy file:///project_1 file:///project_1_branch -m  
"ma_branche"
```

Introduction

Historique

Concepts de base

Concepts avancés

Notion de branches

Introduction

Etude de cas

Ajout de fonctionnalité

Ajout de fonctionnalité
avec synchronisation

Versions logicielles

Modèle Base/work/dev

Portage parallèle

Commandes

Copy

Merge

Outils graphiques

Merge

Commande SVN :

```
svn merge --reintegrate [url]
```

Par exemple :

```
svn merge --reintegrate file:///project_1_branch  
svn commit -m "Merge branch back into trunk!"
```

Introduction

Historique

Concepts de base

Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge

Tortoise SVN

- Intégration à l'explorateur (Windows)
- Beaucoup plus ergonomique que la ligne de commande
- Permettent de parcourir le dépôt beaucoup plus rapidement, avec un report visuel des status des différents fichiers

<http://tortoisesvn.net/>

GESTION DE SOURCES

CONTRÔLE DE VERSIONS

Introduction

Historique

Concepts de base

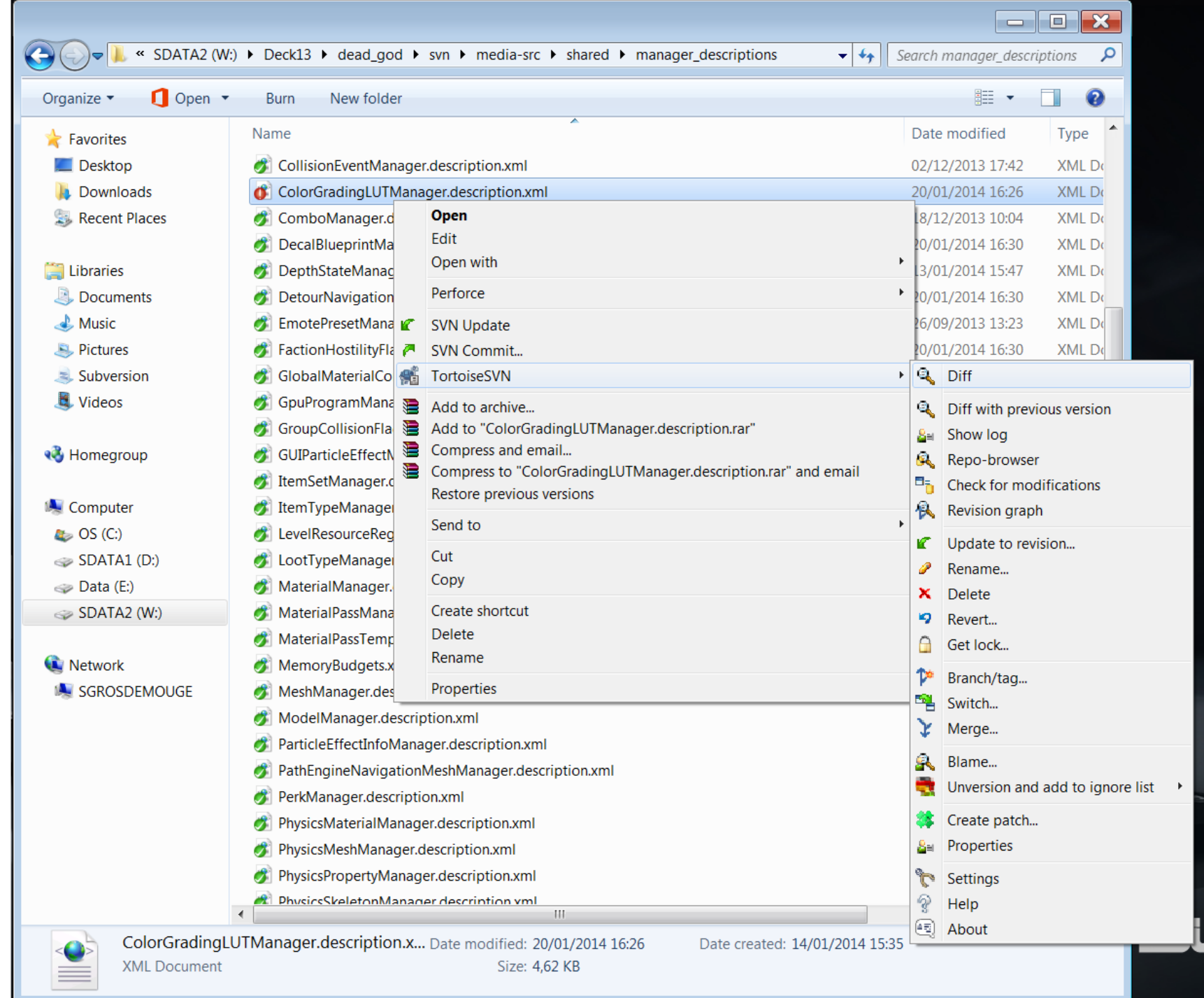
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



Introduction

Historique

Concepts de base

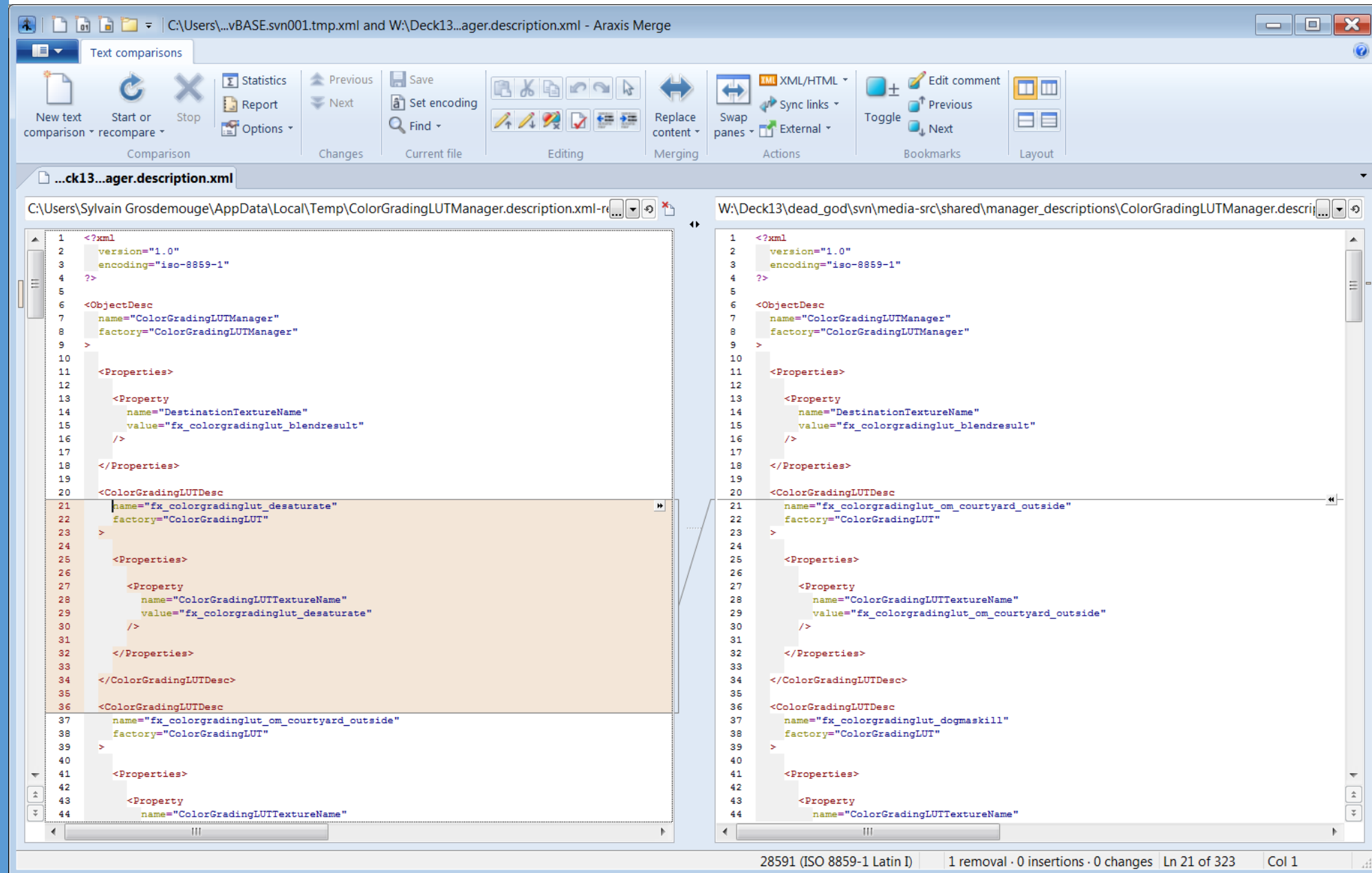
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



Introduction

Historique

Concepts de base

Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge

The screenshot shows the TortoiseSVN interface with a file history table on the left and an XML diff view on the right. The file history table lists revisions, authors, and line numbers. The XML diff view shows the content of the file, with changes highlighted in different colors (yellow for additions, green for deletions, and blue for modifications).

Revision	Author	Line
14821	tlange	1
14821	tlange	2
14821	tlange	3
14821	tlange	4
15819	tlange	5
14821	tlange	6
21814	mvdkloet	7
21814	mvdkloet	8
21814	mvdkloet	9
21814	mvdkloet	10
21814	mvdkloet	11
17851	lkessler	12
14821	tlange	13
17851	lkessler	14
14821	tlange	15
14821	tlange	16
14821	tlange	17
14821	tlange	18
14821	tlange	19
14821	tlange	20
14821	tlange	21
17851	lkessler	22
14821	tlange	23
17851	lkessler	24
14821	tlange	25
14821	tlange	26
17851	lkessler	27
14821	tlange	28
17851	lkessler	29
14821	tlange	30
14821	tlange	31
17851	lkessler	32
15794	tlange	33
17851	lkessler	34
15794	tlange	35
15794	tlange	36
15794	tlange	37

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ObjectDesc name="ColorGradingLUTManager" factory="ColorGradingLUTManager">
  <Properties>
    <Property name="DestinationTextureName" value="fx_colorgradinglut_blendresult" />
  </Properties>
  <ColorGradingLUTDesc name="fx_colorgradinglut_desaturate" factory="ColorGradingLUT">
    <Properties>
      <Property name="ColorGradingLUTTextureName" value="fx_colorgradinglut_desaturate" />
    </Properties>
  </ColorGradingLUTDesc>
  <ColorGradingLUTDesc name="fx_colorgradinglut_om_courtyard_outside" factory="ColorGradingLUT">
    <Properties>
      <Property name="ColorGradingLUTTextureName" value="fx_colorgradinglut_om_courtyard_outside" />
    </Properties>
  </ColorGradingLUTDesc>
  <ColorGradingLUTDesc name="fx_colorgradinglut_dogmaskill" factory="ColorGradingLUT">
    <Properties>
      <Property name="ColorGradingLUTTextureName" value="fx_colorgradinglut_dogmaskill" />
    </Properties>
  </ColorGradingLUTDesc>
  <ColorGradingLUTDesc name="fx_colorgradinglut_ragemode" factory="ColorGradingLUT">
    <Properties>
      <Property name="ColorGradingLUTTextureName" value="fx_colorgradinglut_ragemode" />
    </Properties>
  </ColorGradingLUTDesc>
  <ColorGradingLUTDesc name="fx_colorgradinglut_om_courtyard_staircase" factory="ColorGradingLUT">
    <Properties>
      <Property name="ColorGradingLUTTextureName" value="fx_colorgradinglut_om_courtyard_staircase" />
    </Properties>
  </ColorGradingLUTDesc>
  <ColorGradingLUTDesc name="fx_colorgradinglut_ui" factory="ColorGradingLUT">
    <Properties>
      <Property name="ColorGradingLUTTextureName" value="fx_colorgradinglut_ui" />
    </Properties>
  </ColorGradingLUTDesc>
  <ColorGradingLUTDesc name="fx_colorgradinglut_skill_ram_fadein" factory="ColorGradingLUT">
```

GESTION DE SOURCES

CONTRÔLE DE VERSIONS

Introduction

Historique

Concepts de base

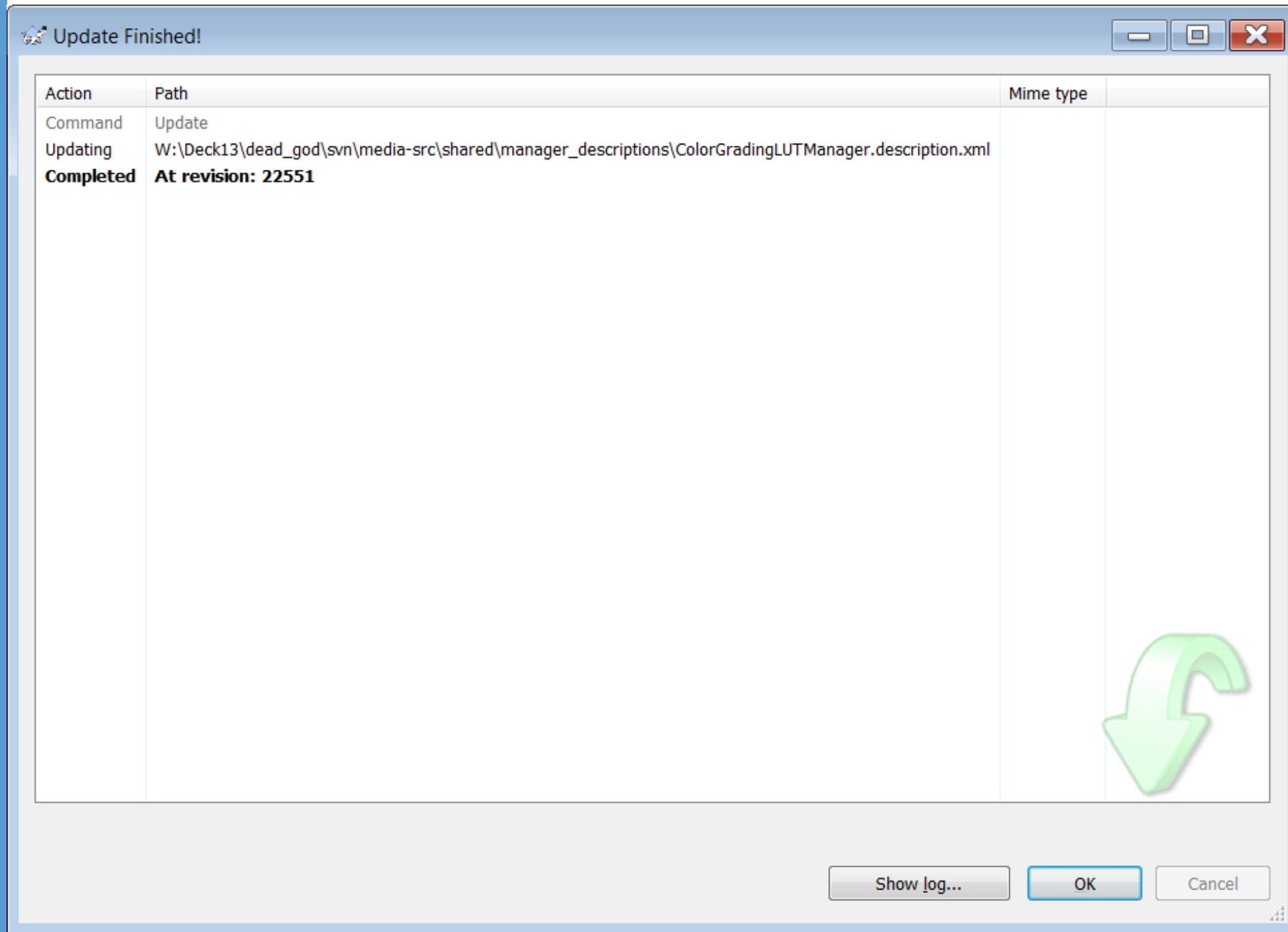
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



GESTION DE SOURCES

CONTRÔLE DE VERSIONS

Introduction

Historique

Concepts de base

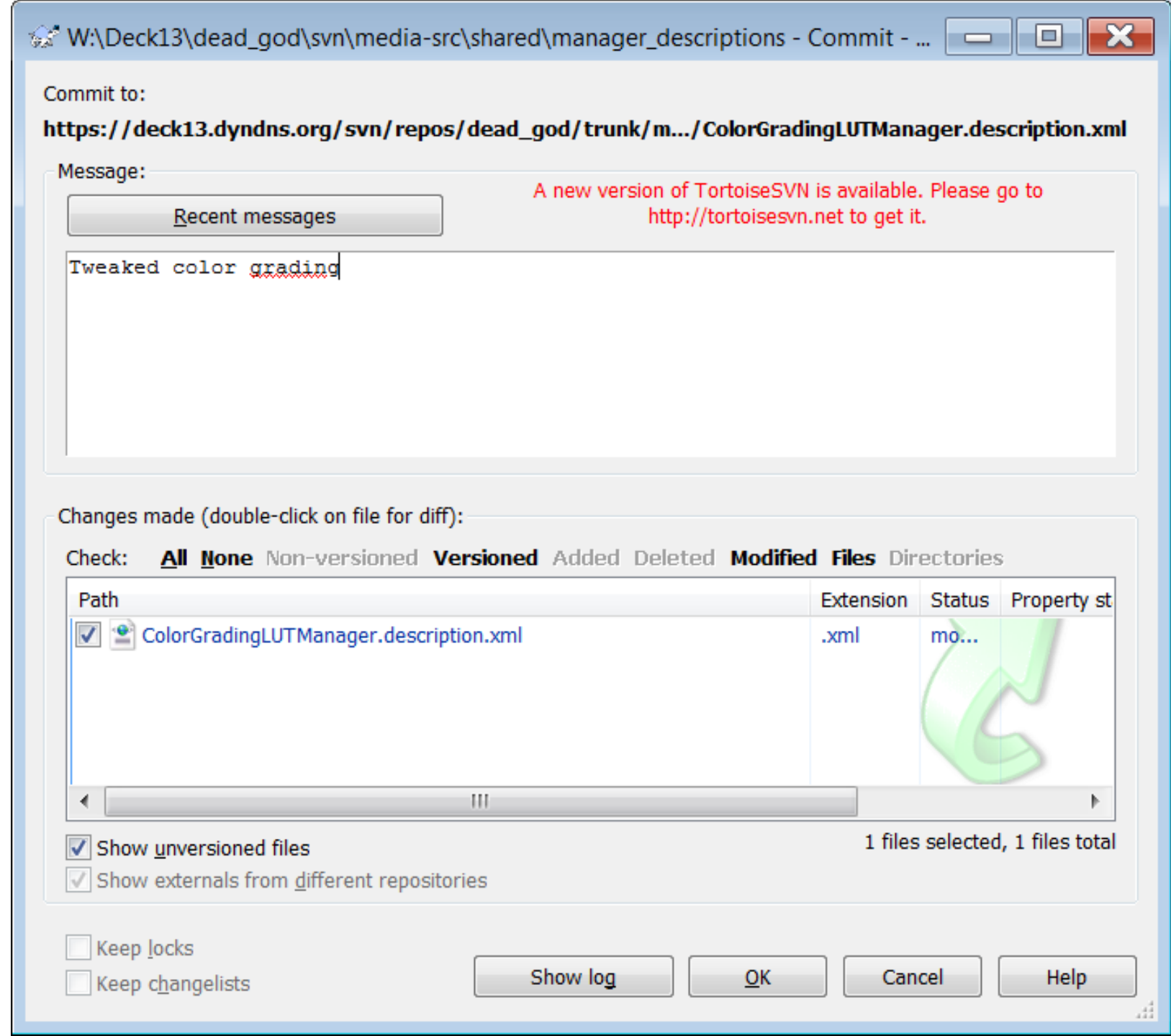
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



GESTION DE SOURCES

CONTRÔLE DE VERSIONS

Introduction

Historique

Concepts de base

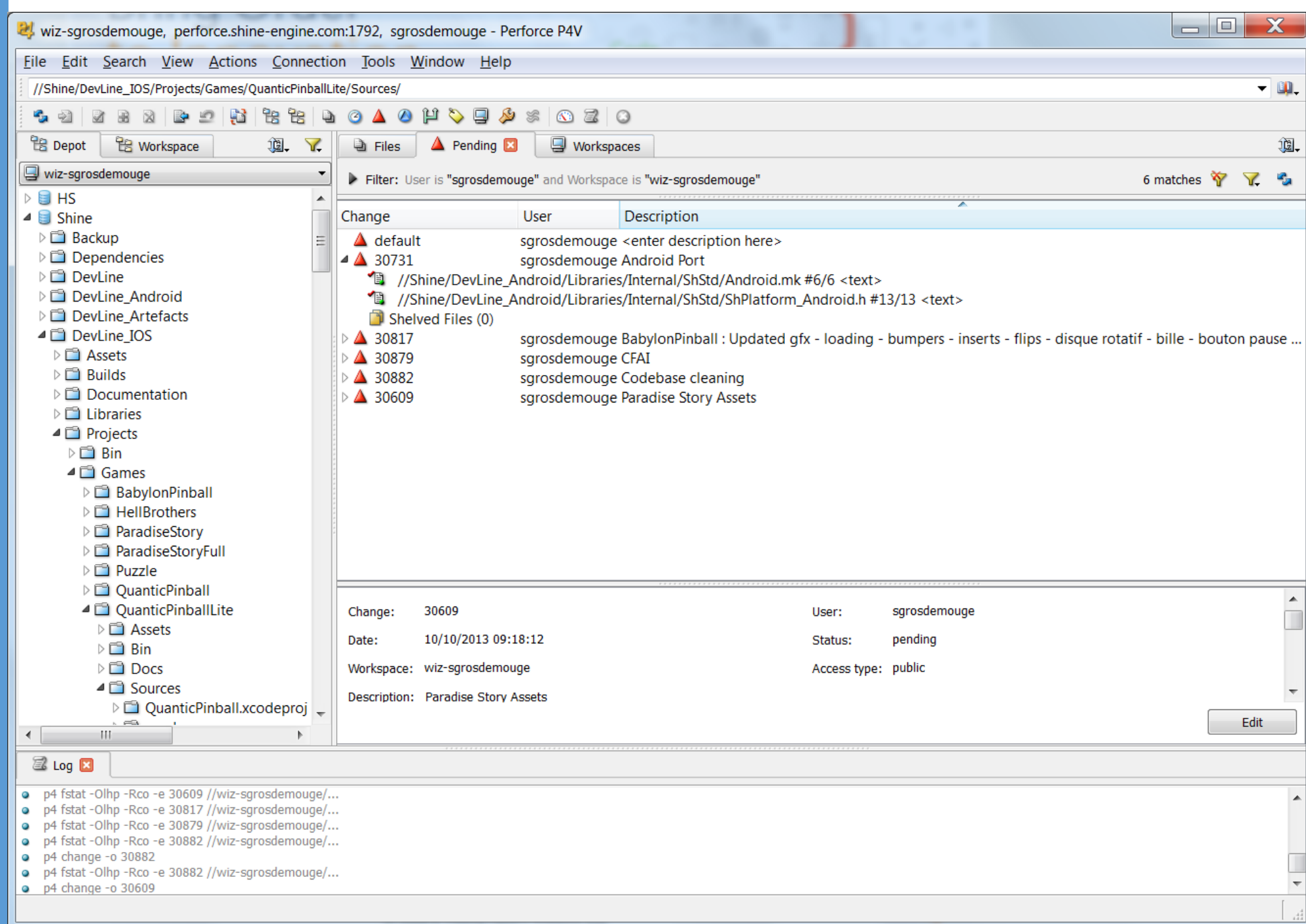
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



Introduction

Historique

Concepts de base

Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge

The screenshot displays the Perforce P4V client interface. The title bar indicates the connection to 'wiz-sgrosdemouge, perforce.shine-engine.com:1792, sgrosdemouge - Perforce P4V'. The main window is divided into several panes:

- Left Pane:** A tree view showing the project structure. The 'Sources' folder is selected, showing a list of source files including 'CShEventReceiver.cpp', 'CShEventReceiver.h', 'CShPinballBackground.cpp', 'CShPinballBackground.h', 'CShPinballBallLockerManager.cpp', 'CShPinballBallLockerManager.h', 'CShPinballBallManager.cpp', 'CShPinballBallManager.h', and 'CShPinballBlocTransition.cpp'.
- Top Right Pane:** A table showing the revision history. The table has columns for 'Revision (Chai)', 'Date Submitted', 'Submitted By', and 'Description'. The current revision, 30877, is highlighted.
- Bottom Right Pane:** A 'Details' pane showing the details for the selected revision (30877). It includes the revision number, date submitted, submitted by, and description.
- Bottom Pane:** A 'Log' pane showing a list of commands executed, such as 'p4 fstat -Olhp -DI -F ^headAction=delete & ^headAction=move/delete //Shine/DevLine_IOS/Projects/Games/QuanticPinball/*'.

Revision (Chai)	Date Submitted	Submitted By	Description
30877	10/01/2014 16:48:31	sgrosdemouge	Quantic Pinball / Quantic Pinball Lite : Merged
30871	10/01/2014 14:00:48	sgrosdemouge	Quantic Pinball : Updated P4 w/ last source code (v1.00)
30787	27/11/2013 18:50:05	rgrimmer	Quantic Pinball : Removed insert on delta table + fixed invaders movements (ki...
30786	27/11/2013 17:12:57	rgrimmer	Quantic Pinball : Fixed kill them all mode (canon position)
30772	26/11/2013 19:56:52	sgrosdemouge	Quantic Pinball : Updates IOS (Size / 2 for video in menu on all versions)
30765	26/11/2013 15:10:22	sgrosdemouge	Quantic Pinball : Icons update / IOS 7 support
30764	26/11/2013 10:31:26	sgrosdemouge	Quantic Pinball : Removed obsolete icons / Added final ones
30762	25/11/2013 14:20:00	sgrosdemouge	Quantic Pinball : Updated unlock sprites + locked levels
30761	25/11/2013 00:54:43	rgrimmer	Quantic Pinball : Reduced small flips' power
30760	25/11/2013 00:24:29	rgrimmer	Quantic Pinball : Added "ball saved" sound & message on multiball
30758	24/11/2013 21:33:01	rgrimmer	Quantic Pinball : Fixed sniper mode
30754	21/11/2013 17:11:43	rgrimmer	Quantic Pinball : Fixed special table (2 balls at start)
30753	21/11/2013 15:48:14	rgrimmer	Quantic Pinball : Show hiscore menu after making a new score + fixed smash-u...
30749	08/11/2013 00:23:07	rgrimmer	Quantic Pinball : Improved ball manager (pool)
30743	04/11/2013 18:18:34	sgrosdemouge	Quantic Pinball : Added support for PCH

Introduction

Historique

Concepts de base

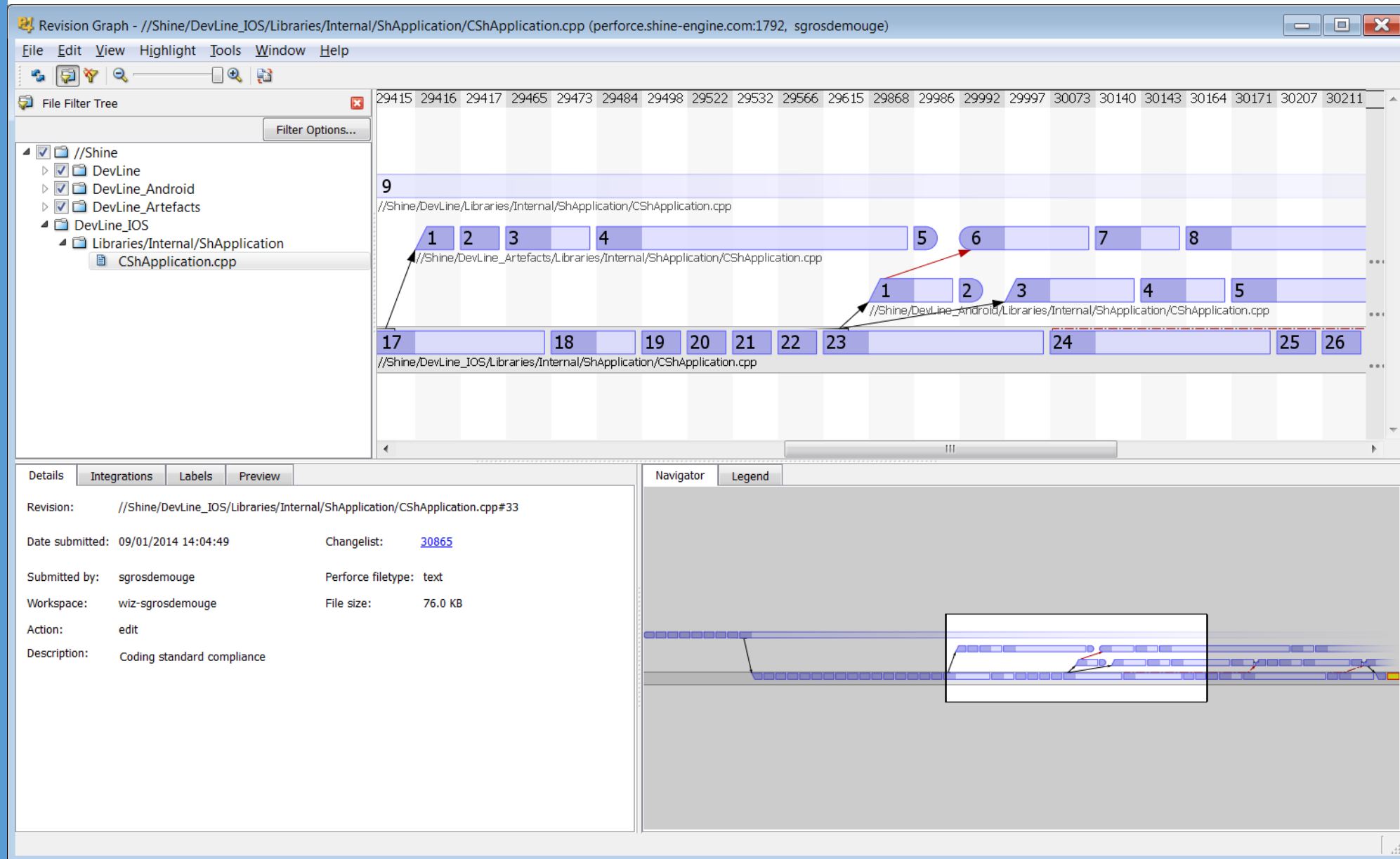
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



Introduction

Historique

Concepts de base

Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge

Araxis Merge

- Outil permettant d'effectuer des opérations de merge sur deux ou trois fichiers, ainsi que sur des répertoires complets.
- S'intègre dans la plupart des logiciels de contrôle de sources
- Permet d'effectuer les opérations de différences et de merge visuellement, de façon efficace

GESTION DE SOURCES

CONTRÔLE DE VERSIONS

Introduction

Historique

Concepts de base

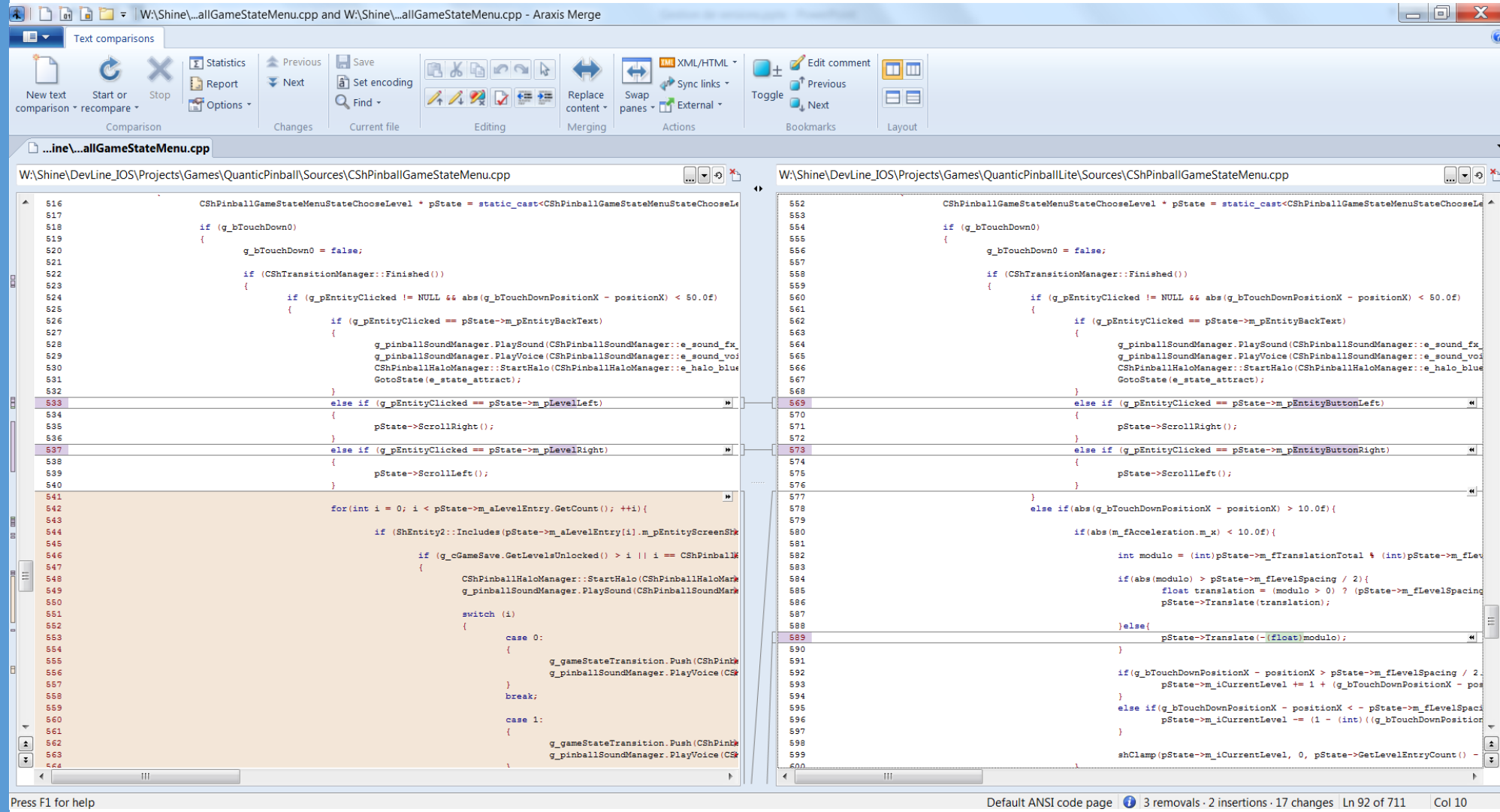
Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge



Press F1 for help

Default ANSI code page 3 removals · 2 insertions · 17 changes Ln 92 of 711 Col 10

GESTION DE SOURCES

CONTRÔLE DE VERSIONS

Introduction

Historique

Concepts de base

Concepts avancés

Outils graphiques

Tortoise SVN

P4V (Perforce Visual Client)

Araxis Merge

