

KDevelop

Ecole ENVOL 2010
Septembre 2010

F. Langrognat



PLAN

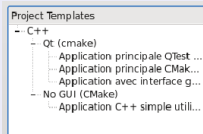
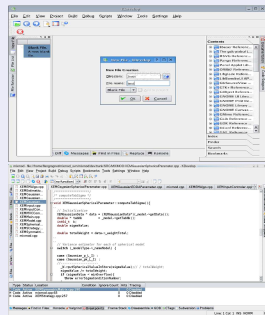
- 1 Introduction
- 2 Principales fonctionnalités
- 3 KDevelop 4
- 4 Alternatives à KDevelop - Conclusion

PLAN

- 1 Introduction
- 2 Principales fonctionnalités
- 3 KDevelop 4
- 4 Alternatives à KDevelop - Conclusion

Historique

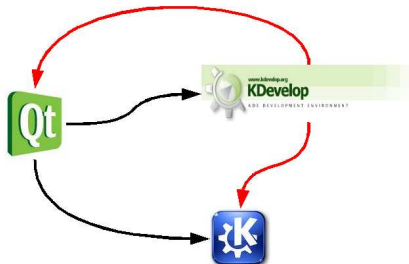
- **1998** : Mise en place du projet KDevelop
 - ▶ Objectif : bâtir un IDE pour KDE facile à utiliser
 - ▶ Langage : uniquement C/C++
- **2001** : KDevelop 2.0
- **2004** : KDevelop 3.0 (complètement réécrit)
 - ▶ Langages supportés : C++, Java, Python, PHP, ...
 - ▶ Plugins
- **1er mai 2010** : KDevelop 4.0.0
Complètement réécrit - encore !
Beaucoup de nouveautés et fonctionnalités améliorées
Version 3.9.95 (ubuntu 9.10) : disponible uniquement pour le C++



Ecosystème

Ecosystème naturel

Développement en C++ sous Linux/KDE



Mais aussi ...

- 15 langages supportés
- Version windows (cygwin) et Mac OS

PLAN

- 1 Introduction
- 2 Principales fonctionnalités**
- 3 KDevelop 4
- 4 Alternatives à KDevelop - Conclusion

Aperçu des fonctionnalités

Langages / Fonctionnalités

Feature/Language	Ada	Bash	C/C++	Fortran	Haskell	Java	Pascal	Perl	PHP	Python	Ruby	SQL
Class wizard	Y	N/A	Y	N/A	N/A	Y	N	N	Y	N	N	N/A
Attribute/Method wizard	N	N/A	Y	N/A	N/A	Y	N	N	N	N	N	N/A
Qt UI subclassing	N/A	N/A	Y	N/A	N/A	Y	N/A	N/A	N/A	N	N/A	N/A
Language parser	ANTLR	Internal	eggparser	Internal	N	ANTLR	ANTLR	Internal	Internal	Internal	N	N
Memory class store	CodeModel	CodeModel	CodeModel	CodeModel	N	CodeModel	CodeModel	CodeModel	CodeModel	CodeModel	N	N
Persistent class store	N	N	Cache	N	N	N	N	N	N	N	N	N/A
Code completion	N	Y	Y	N	N	Y	N	N	Y	N	N	N
Problem reporter	Y	N	Y	N	Y	Y	Y	N	Y	N	N	N
Application templates	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Application import templates	Y	N	Y	Y	N	Y	Y	Y	Y	Y	Y	N
Source file templates	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	N
Code abbreviation templates	N	N	Y	N	N/A	N	Y	N	Y	N	N	N
Source code formater	N	N	Y	N	N	Y	N	N	N	N	N	N
Documentation topics	Y	Y	Y	N	Y	Y	N	Y	Y	Y	Y	N
Debugger	N	N	Y	N	N/A	Y	N	N	N	N	N	N
Compiler plugins	N	N/A	gcc	PCNPL PCT2	Gaspow	N	Dolpin, Free Pascal	N	N	N	N	N

Legend:

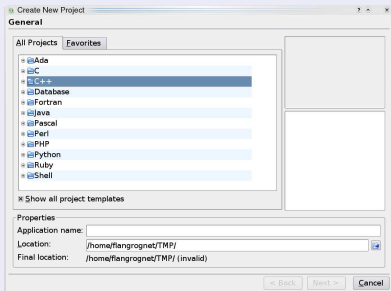
 	Feature has not been implemented
 	Feature is in alpha state, the development has only been started or the feature is turned off at the moment
 	Feature is in beta state or it is unfinished
 	Feature is considered to be complete or complete enough for the release

Fonctionnalités (1)

Présentation des fonctionnalités de KDevelop 3.5.3 (et ... 4.0)

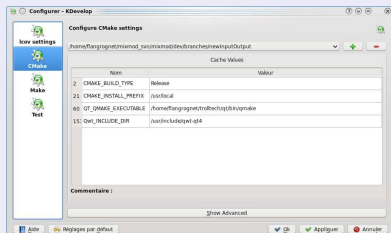
15 langages

Ada, C, C++, Objective-C, SQL,
Fortran, Haskell, Java, PHP, Pascal,
Perl, Python, Ruby, Bash



Construction de projet

- autotools
- qmake
- **cmake**
- ant (java)



Fonctionnalités (2)

Edition (texte/classes)

- Coloration syntaxique
- Indentation automatique
- Complétion automatique
- Aide à l'ouverture/fermeture de parenthèses, accolades
- Utilisations de variables (KDevelop4)
- Aide à la création de classe
- Visualisation de classes :
 - ▶ Classe mère, classes dérivées
 - ▶ Membres privés, publics, protégés
 - ▶ Liste des utilisations des méthodes/fonctions

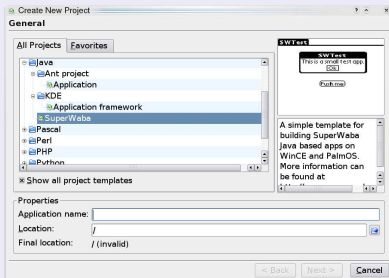
The screenshot shows the KDevelop 4 IDE interface. The main editor displays the constructor of the `XEMClusteringMain` class, which takes an `XEMClusteringInput` object as input. The code includes various member variables like `_nbCriterion`, `_nbModelType`, `vector<int64_t>`, `_nbNbCluster`, `_nbModelType`, `_nbSelection`, `_nbEstimation`, `int64_t`, and `ModelType`. It also shows the initialization of `inputData`, `workingData`, and `knownPartition`.

A popup window titled "Uses of XEMClusteringMain::XEMClusteringMain(XEMClusteringInput * cInput)" is visible on the right. It shows the "Declaration" of the constructor and a "Contexte" section listing several lines of code where the constructor is used, such as `Line 45 XEMClusteringMain::XEMClusteringMain(XEMClusteringInput * cInput) {` and `Line 81 inputKnownPartition = cInput->getKnownPartition();`.

Fonctionnalités (5)

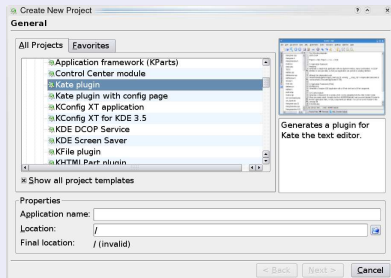
Kits de développement

- QT (QT Designer, QT Linguist)
- GTK
- wxWidgets
- superwaba
- GBA



Développement de plugins

- KDE
 - ▶ Kate
 - ▶ KFile
 - ▶ Konqueror
 - ▶ Noatun
 - ▶ ...
- KDevelop



PLAN

- 1 Introduction
- 2 Principales fonctionnalités
- 3 KDevelop 4**
- 4 Alternatives à KDevelop - Conclusion

Quelques améliorations (parmi d'autres)

- Nouveau langage : **C#**
- Nouveaux SGV (décentralisés) :
 - ▶ **git**
 - ▶ **mercurial**
 - ▶ **bazaar**
- Tests
 - ▶ **CppUnit, QTest**
- **Edition** (coloration, navigation dans les classes, ...)

KDevelop4 - Edition (1)

Coloration syntaxique

```
.....
Exemple description
-----
...../
#include <iostream>
using namespace std;

/** Commentaire Doxygen
@author F. Langrognat
@return : un entier
@param : les paramètres ...
**/
int maFonction(bool monBooleen, double monDouble, char monChar){
// Mes commentaires
string maChaine = "ceci est une chaîne de caractères";
cout<<maChaine<<endl;

int monEntier = 2;
long monLong = monDouble / monEntier;
monEntier++;

if (monBooleen == true){
monEntier = 2;
monChar = 'a';
monDouble = 3.14;
}

//TODO A faire !!

return 0;
}
```

```
.....
Exemple description
-----
...../
#include <iostream>
using namespace std;

/** Commentaire Doxygen
@author F. Langrognat
@return : un entier
@param : les paramètres ...
**/
int maFonction(bool monBooleen, double monDouble, char monChar){
// Mes commentaires
string maChaine = "ceci est une chaîne de caractères";
cout<<maChaine<<endl;

int monEntier = 2;
long monLong = monDouble / monEntier;
monEntier++;

if (monBooleen == true){
monEntier = 2;
monChar = 'a';
}

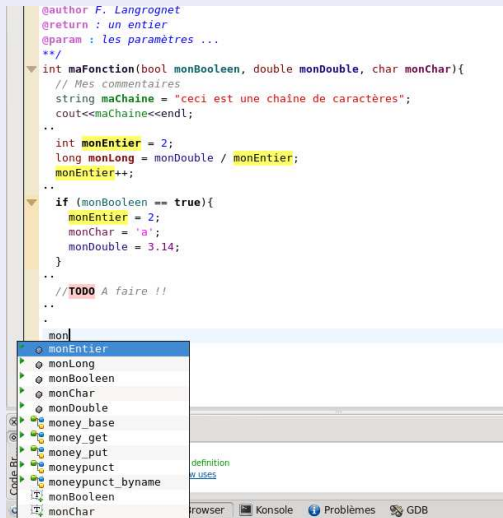
//TODO A faire !!

return 0;
}
```

Contexte					Normal	Sélectionné	Fond
Normal	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Mot clé	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Type de données	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Décimale / Valeur	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Entier en base N	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Nombre à virgule flottante	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Caractère	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Chaine	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Commentaire	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Autres	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Alerte	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Fonction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Marqueur de région	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune
Erreur	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>			Aucune

Complétion automatique

```
@author F. Langrognat
@return : un entier
@param : les paramètres ...
**/
int maFonction(bool monBooleen, double monDouble, char monChar){
    // Mes commentaires
    string maChaine = "ceci est une chaine de caractères";
    cout<<maChaine<<endl;
    ..
    int monEntier = 2;
    long monLong = monDouble / monEntier;
    monEntier++;
    ..
    if (monBooleen == true){
        monEntier = 2;
        monChar = 'a';
        monDouble = 3.14;
    }
    ..
    // TODO A faire !!
    ..
    mon
```



The screenshot shows the KDevelop4 IDE with a code editor displaying C++ code. The code defines a function `maFonction` with parameters `monBooleen`, `monDouble`, and `monChar`. The function body includes comments, a string declaration, a cout statement, variable declarations and assignments, and an if-statement. The cursor is positioned at the end of the line `mon`. A dropdown menu is open, showing a list of variables: `monEntier`, `monLong`, `monBooleen`, `monChar`, `monDouble`, `money_base`, `money_get`, `money_put`, `money_punct`, `money_punct_byname`, `monBooleen`, and `monChar`. The `monEntier` option is highlighted. The IDE interface includes a sidebar with 'Code Browser', a bottom toolbar with 'Konsole', 'Problèmes', and 'GDB', and a status bar at the bottom.

Information contextuelle

```
.....  
Exemple description  
-----  
.....  
#include <iostream>  
using namespace std;  
  
/** Commentaire Doxygen  
@author F. Langrognnet  
@return : un entier  
@param : les paramètres ...  
**/  
int maFonction(bool monBooleen, double monDouble, char monChar){  
    // Mes commentaires  
    string maChaine = "ceci est une chaine de car  
    cout<<maChaine<<endl;  
    ..  
    int monEntier = 2;  
    long monLong = monDouble / monEntier;  
    monEntier++;  
    ..  
    if (monBooleen == true){  
        monEntier = 2;  
        monChar = 'a';  
        monDouble = 3.14;  
    }  
    ..  
    //TODO A faire !!  
    ..  
    return 0;  
}
```

Uses of `maFonction::monDouble`

Project: No project File: /home/flangrognnet/TMP/exemple_coloration.cpp (2)

Declaration
[Line 13](#) int maFonction(bool monBooleen, double monDouble, char monChar){

Contexte [maFonction\(...\)](#) (2 uses)
[Line 19](#) long monLong = monDouble / monEntier;
[Line 25](#) monDouble = 3.14;

KDevelop4 - Edition (4)

Edition de classes, navigation

The screenshot displays the KDevelop4 IDE interface. The top menu bar includes options like 'Fichier', 'Édition', 'Affichage', 'Navigation', 'Project', 'Exécuter', 'Code', 'Signets', 'Window', 'Outils', 'Configuration', and 'Aide'. The toolbar contains icons for 'New', 'Code Area', 'Debug Area', 'Test Area', 'Construire', 'Execute', 'Debug', 'Stop', 'Continue', and 'Quick Open...'. The main window shows the file path: `/home/flangrognet/mixmod_svn/mixmod/dev/branches/newInputOutput/SRC/MIXMOD/XEMGaussianEDDAParameter.h - KDevelop`. The editor displays the source code for `XEMGaussianEDDAParameter.h`, with a context menu open over the `getPdf` function. The context menu options are: 'Rename XEMGaussianEDDAParameter::getPdf', 'Find Uses', 'Show Declaration', and 'Show Definition'. The 'Uses' panel at the bottom shows the following information:

```
Uses of XEMGaussianEDDAParameter::getPdf
Project: newInputOutput File: SRC/MIXMOD/XEMGaussianEDDAParameter.h (0 uses)
Declaration
Line 79 double getPdf(int64_t iSample, int64_t kCluster) const;
Project: newInputOutput File: SRC/MIXMOD/XEMParameter.h (0 uses)
Declaration
Line 87 virtual double getPdf(int64_t iSample, int64_t kCluster) const = 0;
Project: newInputOutput File: SRC/MIXMOD/XEMBinaryParameter.h (0 uses)
Project: newInputOutput File: SRC/MIXMOD/XEMGaussianParameter.h (0 uses)
Project: newInputOutput File: SRC/MIXMOD/XEMGaussianHDDAParameter.h (0 uses)
```

PLAN

- 1 Introduction
- 2 Principales fonctionnalités
- 3 KDevelop 4
- 4 Alternatives à KDevelop - Conclusion**

Et si l'on parlait des besoins ?

Un IDE pour C++



Critères de choix :

- Gratuit, Payant
- Linux, Windows
- Fonctionnalités :
complétion de code,
gestionnaire de version,
debugger, ...

Conclusion

KDevelop

- IDE très complet
- Nombreux outils pour le C++
valgrind, KCacheGrind, Doxygen, ...
- Création facile de projets divers
QT, WxWidgets, ...
- Edition de texte, navigation classes,
fichiers de grande qualité (KDevelop4)
- Debugger performant
- Projet actif et pérenne
- Utilisation sous Windows,
MacOS très marginale
- Cross compilation ?
- Création de package ?

KDevelop

FIN

Merci de votre attention

Florent Langrognat

