

KDevelop 1.4

Utat a feltörekvőknek! – a KDevelop összetett fejlesztőkörnyezet programkészítésre, hibakövetésre és karbantartásra.

Egyetlen fejlesztőeszközben szenved hiányt a Nyílt Forráskód Közössége, egy profi IDE-ben (Integrated Development Environment – összetett fejlesztőkörnyezet). Szerencsére a KDevelop rendelkezésünkre bocsátott egy ilyen eszközt, amely egyesíti a már létező nyílt forráskódú termékeket és a hozzájáruló munkáit. Csakhogy vajon versenyre kelhet-e a KDevelop egy (többnyire valamilyen nem Unix-alapú) kereskedelmi IDE-vel?



Mi is az IDE?

Az IDE olyan (lehetőleg grafikus) környezet, amely programkészítésre, hibakövetésre és karbantartásra szolgál. Ennek a környezetnek három alapösszetevője: a programozók szövegszerkesztője, amely szerkezetérzékeny a programozási nyelvre; a GUI (grafikus felhasználói felület)-szerkesztő, ezt az alkalmazás grafikus felületének tervezéséhez használhatjuk fel; és végül a hibakereső-nyomkövető program, ami a kódban rejlő hibákat deríti fel. Ezek az IDE-vel szemben támasztott alapvető követelmények. Ahhoz azonban, hogy igazán előnyös eszközként használhassuk, további összetevőkkel kell kiegészíteni.

Telepítés

Mivel a nyílt forráskódú programok többnyire a feladat elvégzésére törekednek, és kevesebb figyelmet fordítanak arra, hogy felhasználóbarátok legyenek, a telepítés néha meglehetősen nehézkes és elkedvetlenítő lehet; különösen, ha figyelembe vesszük a számtalan Linux-változatot, valamint a folyamatosan változó könyvtárakat és eszközöket.

A KDevelop RPM binárisok letölthetők a KDevelop honlapról vagy felkutatathatók valamilyen keresőlap segítségével, például a <http://www.rpmfind.net/> címen.

Ehhez az ismertető megírásához teljesen új Linuxot telepítettem és megbizonyosodtam róla, hogy mindazt a csomagot és képességet tartalmazza, amit csak a terjesztés megenged. Még így is belefutottam azonban pár telepítési akadályba: néhány könyvtárnál olyan függőségekkel találok, amelyek az én Linux-telepítésem nem is léteztek. Egy kis internetes kitérő – a hiányzó könyvtárak letöltése – után viszont a gond hamar megszűnt.

A teljes telepítési idő (gyors Internet-eléréssel és némi szaktudással felvértezve) körülbelül félórát vett igénybe. Ez a telepítési módszer eszményi a kicsit is rendszergazdai vénával megáldott felhasználók számára. Néha a forrásból fordítás hasznos lehet a nem Linux/Unix operációs rendszert használó programozók vagy a leendő KDevelop-fejlesztők számára a testreszabott Linux-változatokhoz. De csak a tapasztalt vagy nagyon határozott fejlesztők próbálják meg forrásból lefordítani a KDevelopot!

Elsőként telepíteni kell az összes szükséges könyvtár fejlesztői változatát. Mivel nem áll a rendelkezésünkre egyszerű mód ezeknek

a függőségeknek a megállapítására, a forrás fordítása valószínűleg próbálkozások és hibák sorozata lesz.

A KDevelop szolgáltatása, hogy számos létező nyílt forráskódú eszközt fel tud használni. Az összes ilyen eszköz megléte nem kötelező, de ha azt szeretnénk, hogy a KDevelop olyan hatékony legyen, mint ahogyan azt elvárjuk, mindenképpen ajánlott. Amikor a KDevelop első ízben indul el, megadja a vonatkozó eszközök listáját, jelenlévőként vagy hiányzóként bejelölve az egyes elemeket (lásd az *1. képet*). A lista nem végleges, ugyanis a hiányzó eszközök később még telepíthetők.

A KDevelop által felhasznált segédeszközök: g++2.7.2, g++2.8.1 vagy egcs 1.1 (én a g++2.9.2-t ajánlom); make; perl 5.004; autoconf 2.12; automake 1.2; flex 2.5.4; gettext; Qt 2.2.X (magába foglalja a Qt designert és a uic-t) és a KDE 2.X.

A választható eszközök: encrypt, Ghostview vagy KGhostview, Glimpse 4.0, htdig, sgmltools 1.0, KDE-SDK (KDE software development kit), KTranslator, KDbg, KIconedit és Qt Linguist. Bár választhatók, a legjobb, ha ezek az eszközök is mind elérhetők.

Képességek

A KDevelop teljesíti az IDE három alapvető követelményét (tartalmaz szövegszerkesztőt, GUI-szerkesztőt és -hibakeresőt – lásd a *2. képet*, és más olyan képességekkel is fel van vértezve, amelyek hatékony és megbízható, akár üzleti feladatokra is alkalmazható eszközzé teszik.

Egy összetett program mind a kezdők, mind a profik számára rémisztő lehet, ezért nagyon fontos a jó programleírás. A KDevelophoz készült leírás kimerítően jó, közvetlenül elérhető súgót szolgáltat – sajnálatunkra azonban a pillanatképek és egyéb grafikai elemek nem túl sűrűn fordulnak elő benne. A tartalomérzékeny sűgő az eszközsorból és a *What's this?* kurzormódból egyaránt elérhető.

A KDevelop hivatkozik a KDE Lib és a Qt leírásaira is. Lehetőség nyílik továbbá könyvjelzők elhelyezésére, ami egyszerűvé teszi a visszatérést a tárgyhoz tartozó leíráshoz. A KDevelop honlapján további tanítófájlok és leírások is elérhetők.

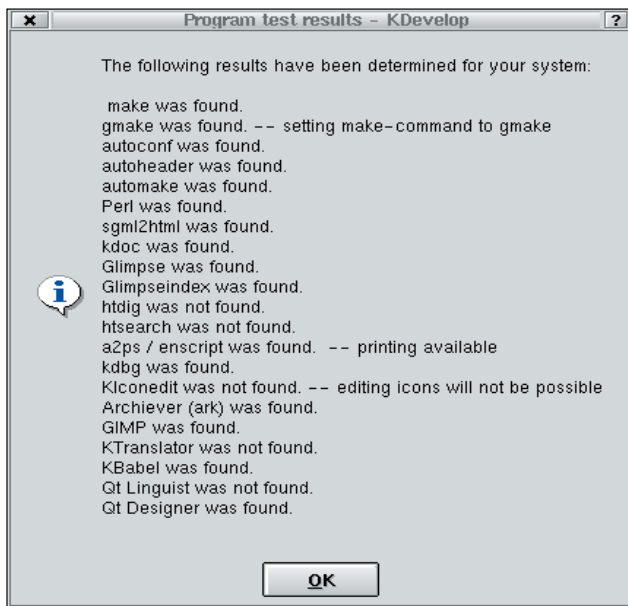
A KDevelop beépített HTML-böngészővel rendelkezik, így a dokumentáció elérése gördülékeny és külső böngésző használata sem szükséges.

Az alapfelület elemei a következők:

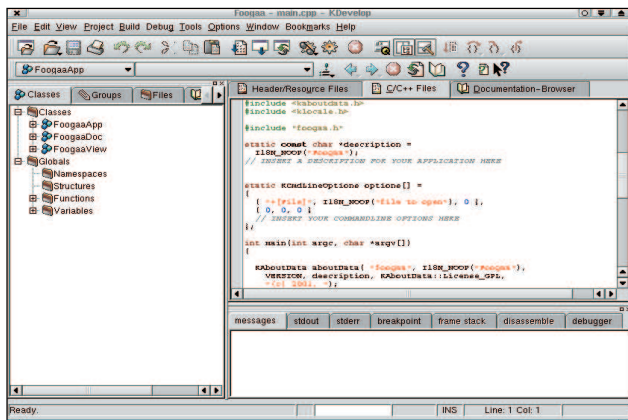
Tree View, amely a class, a groups, a file, a books és a watch nézeteket tartalmazza;

Output View (Kimeneti nézet), ez a következő kimenetek üzeneteit jeleníti meg: stdout, stderr, debugger breakpoints, debugger frame stack, debugger disassembly és debugger messages;

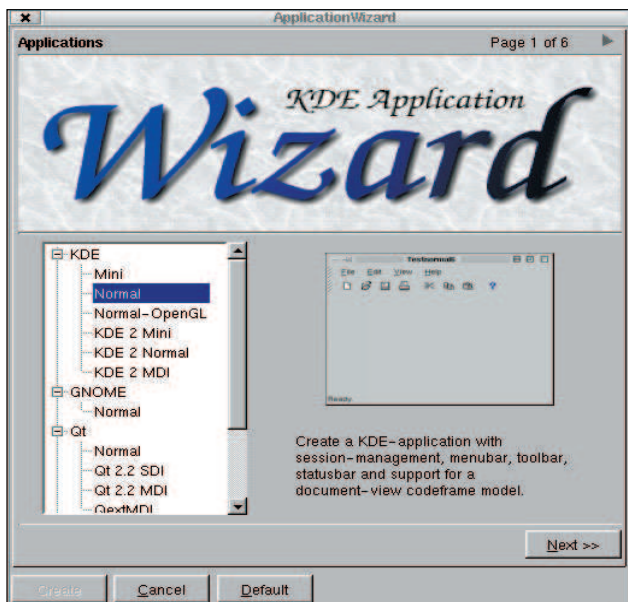
Szövegszerkesztő és Dokumentáció, melybe beleértendő a fejléc- és erőforrás-szerkesztő, C/C++ fájl szerkesztő és a dokumentációböngésző; *Eszköztár* azaz a főmenüparancsok ikonos megfelelője.



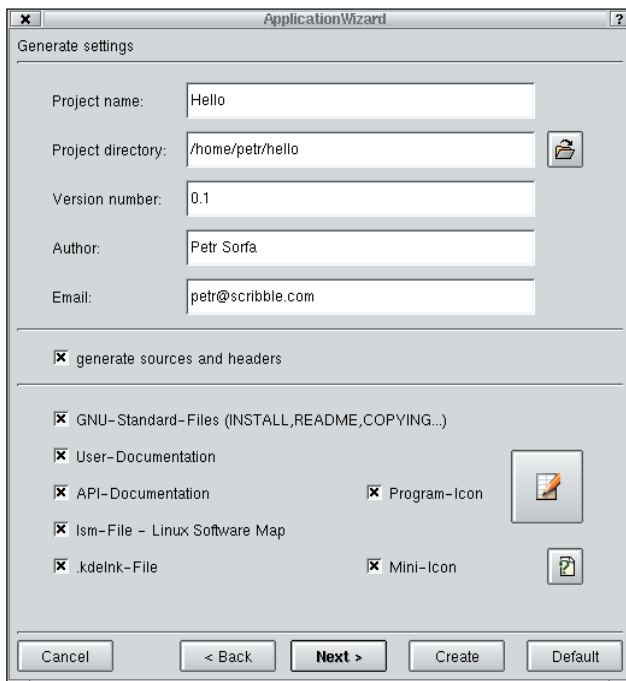
1. kép A KDevelop első indulásakor észleli a telepített elemeket



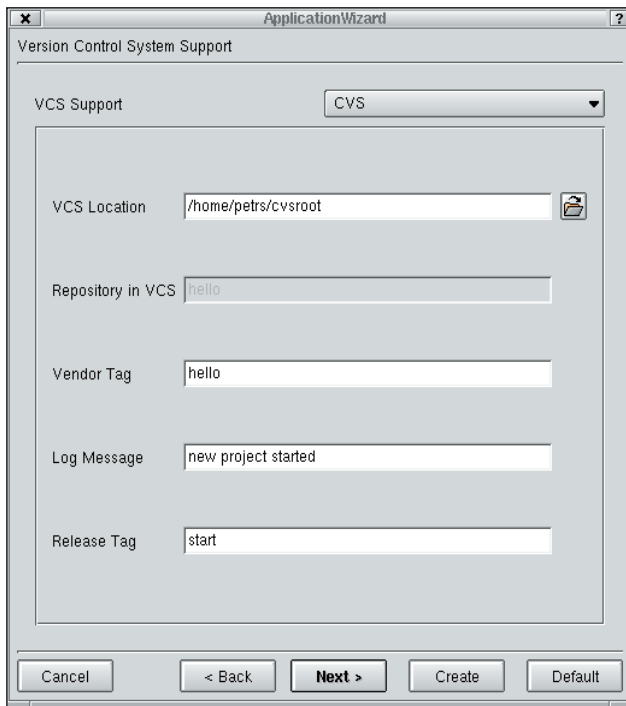
2. kép A KDevelop 1.4 működés közben



3. kép Az alkalmazásvarázsló



4. kép Általános projektbeállítások



5. kép A változatkezelő rendszer kiválasztása

A KDevelop alatt a legegyszerűbb dolgok egyike a projektkészítés, melyet az *Application Wizard* segítségével, három lépésben tehetünk meg:

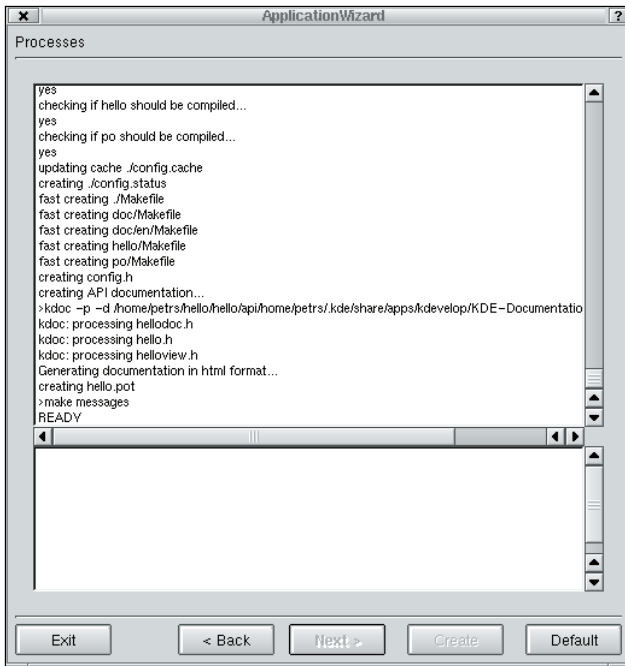
1. *Application Type* (lásd a 3. képet)

E lépés során a felhasználó különféle sablonok közül választhat: KDE 2 Mini; KDE 2 Normal; KDE 2 MDI GNOME (Normal); Qt (Normal, Qt 2.2 SDI, Qt 2.2 MDI, QextMDI); Terminal, például szöveges (C, C++) és egyéb (saját készítésű).

© Kiskapu Kft. Minden jog fenntartva



6. kép Fejlécsablonok



7. kép A projektkészítés kezdete

2. Általános beállítások (lásd a 4. képet)

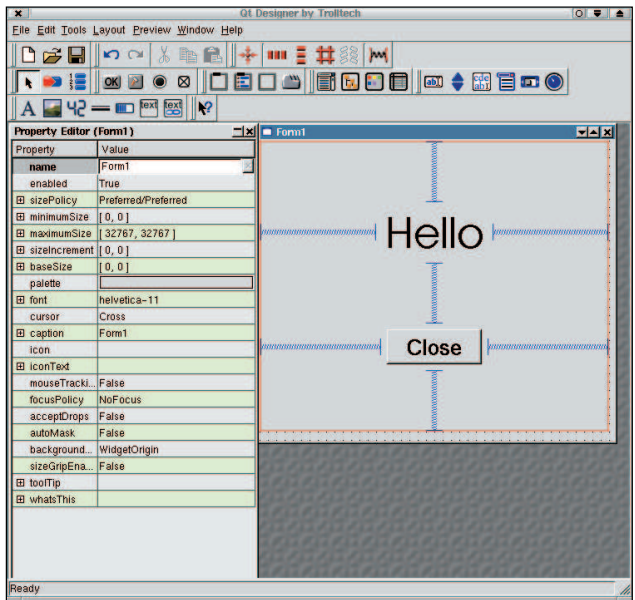
Itt lehet megadni a projekt nevét, elérési útját, kezdeti változat-számát, alkotójának nevét és levélcímét. Lehetőség nyílik a projekt-hez kötődő fájlok létrehozására is, például forrásfájlok, fejlécek, GNU-alapfájlok, ikonok; továbbá a projekthez tartozó leírás szintén e helyütt készíthető el.

3. Változatkezelő rendszer (lásd az 5. képet)

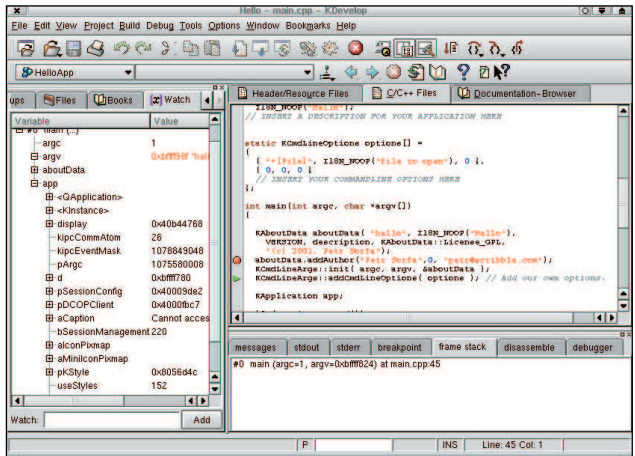
A Változatkezelő rendszer beállításait módosíthatjuk. Ez a rész Linux-változathoz, de általában a CVS eszköz található itt.

4. Fejléc és kódfájlokhoz használható fejlécsablonok (lásd a 6. képet)

A fejlesztők itt dönthetik el, hogy milyen sablonokat használ a



8. kép A KDevelop 1.4 GUI-szerkesztője: a Trolltech Qt Designer



9. kép Hibakeresés a projektben

rendszer a program-fájlokhoz, illetve fejlécfájlokhoz. Ezek a fejlécek tagkibontásos módszerrel testreszabhatók, néhány adat (szerző, fájlnev, dátum) pedig önműködően kitöltésre kerül.

5. Projektkészítés (lásd a 7. képet)

A projektkészítés végső lépéseként, az automake és configure eszközök segítségével elkészülnek a megfelelő projektfájlok és könyvtárak. Figyelem! Ha a szükséges eszközök hiányoznak a Linux-változathól, a létrehozás folyamata megszakadhat. Amennyiben ilyesmi történik, a leghelyesebb feltelepíteni a hiányzó összetevőket, és újra felépíteni a projektet, ugyanis egy projektleállási hibát hihetetlenül nehéz helyreállítani.

Amint a projekt létrejött, megkezdődhet a fejlesztés. Ezen a ponton nagyon hasznos, ha projektet megpróbáljuk felépíteni és végrehajtani a, hogy ezáltal felderíthessük az esetleges összeállítási hibákat.

Qt Designer

A KDevelop 1.4-es változata a Trolltech Qt Designerét használja. A Qt Designer profi felülettel rendelkezik, ahol a legtöbb Qt vizuális elem (widget) használata grafikus felületen keresztül oldható meg, és nagyon hasznos eszköz lehet a GUI-elemgyűjtemény és -összetevők



10. kép Lebegő hibakereső eszköztár

Előnyök

- Kitűnő leírás
- Nagyszerű támogatottság
- Az IDE legfontosabb szolgáltatásait támogatja
- Üzembiztos és használható valós többfejlesztős projekteken
- Testreszabható



Hátrányok

- Vizuáliselem-támogatottság (widget support)
- A programozási nyelvek és fordítók támogatása
- Létező projektek importálása
- Nincs RAD-eszköz (egyelőre)



egymás közötti kapcsolatainak kialakításában (akár vizuális programozásnak is nevezhetnénk).

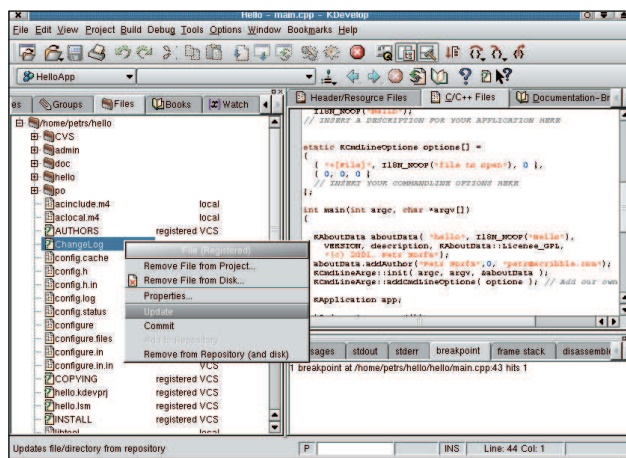
Foglaljuk össze röviden a KDevelop alatti Qt Designer segítségével végzett grafikus felület készítés lépéseit (lásd a 8. képet)! A Qt Designer komolyabb termék, képességeinek és használatának teljes leírása önmagában is megtöltene egy cikket, ezért most csak a KDevelopdal való kapcsolatára térünk ki. A Qt Designer lehetővé teszi a tervezőkörök (layout tools) használatát és valamennyi vizuáliselem-tulajdonsághoz hozzáfér. Képes az egyes vizuális elemek közötti függőségi viszonyok kiépítésére is, például: egy gomb megnyomásakor záródjon be az adott ablak.

A Qt Designer csak egy a párbeszédablakot leíró köztes XML .ui fájl készítő. A tényleges forráskódot egy másik eszköz, az uic készíti el a .ui fájlból. A KDevelop 1.4 támogatja ugyan az .ui fájlokat, de azokat a felhasználónak kell a projekthez adnia. Amikor a felhasználó make-et vagy rebuild-et indít, a KDevelop a szükséges forráskód elkészítéséhez önműködően meghívja az uic-t. Sajnálatos módon az uic az összes elkészült forrásfájlt újraírja, valahányszor a felhasználó a Qt Designerben az .ui állományt megváltoztatja. Emiatt a felhasználó ezeket a forrásfájlokat tulajdonképpen nem szerkesztheti. Az uic eszköz által készített kód kihasználásához a létrehozott kódsztyálokat törölni kell, mielőtt saját képességeket vihetnénk a programba.

Hibakeresés

Hibakereső szolgáltatások nyújtásához a KDevelop a GDB-t használja fel (lásd a 9. képet). A szerkesztőablak bal oldali oszlopára kattintva töréspontokat állíthatunk be az elkészült kódban. A töréspontok akkor is beállíthatók, ha a program nem fut vagy lefordíthatatlan állapotban van. Ezek az úgynevezett kötetlen (lazy) töréspontok. A KDevelop a kötetlen töréspontokat kékkel, míg a működőket vörös színnel jelöli. A megfelelő forrásor utáni kis zöld nyíl a végrehajtás pillanatnyi pontját mutatja. A KDevelop a legtöbb alapvető hibakereső szolgáltatást támogatja. Képes az egyszerű végrehajtásra, a következő sorra ugrásra és a programmegszakításra. A felhasználónak lehetősége nyílik a hibakeresőt lebegő eszköztár formájában előhívni, ami megkönnyíti a hibakereső parancsok elérését (lásd a 10. képet). Fanézetben a változófa a jelenleg elérhető változókat mutatja be.

A hibakeresőhöz tartozó adatok a hibakereső (debugger), assembly, verem (frame stack) és töréspont kimeneti ablakokban jelennek meg. Az alapértelmezett hibakereső hátránya, hogy hiába lenne szükség esetleg finomhangolt hibakeresésre, a GDB-t nem lehet közvetlenül



11. kép A forrásvezérlő rendszer használata

elérni. A másik gond, hogy a felhasználó a változók értékeit a Watch input sorban nem magától értetődő módon változtathatja meg. A KDevelop úgy is beállítható, hogy külső hibakeresőt használjon, legyen az a népszerűsített kdbg vagy xgdb.

Külső alkalmazások

A KDevelop KDE-alkalmazások futtatását teszi lehetővé saját keretrendszeren belül. Néhány alkalmazás, például a Gimp, az Ark és a KBabel alapértelmezés szerint be is van állítva, további elemeket pedig az Options tools menü alatt adhatunk a rendszerhez.

Fordítás, felépítés és terjesztés

A projekt fordítása és felépítése több menüszolgáltatáson keresztül is elérhető. Ezek a Make, Clean, Rebuild, Clean for distribution és Auto configuration menüpontok. A KDevelop, ha szükséges, még a futtatás előtt rákérdez, hogy újra akarjuk-e fordítani a programot.

Project Make Distribution (Terjesztéskészítő)

A Source.tgz menüpont forrásváltozatok készítését teszi lehetővé. Úgy tűnik, a hasznosabb csomagolást nyújtó RPM vagy RPM spec fájlok önműködő készítésére sajnos nincs lehetőség.

Configuration Management (Beállításkezelő) Source Control (Forrásvezérlő)

Ha projektkészítéskor kiválasztottuk a változatkezelő rendszert, a kijelölt fájlokat a forrásvezérlő (source control) rendszerhez adhatjuk (lásd a 11. képet). Ezt úgy tehetjük meg, hogy Group vagy File nézetben a fájl Add to Repository helyi menüt (Pop Out) választjuk. A változásokat a Commit pont kiválasztásával érvényesíthetjük, más fejlesztők változtatásait pedig az Update pont választásával hozhatjuk vissza.

Mivel a CVS távoli tárlóhelyeket is támogat, a KDevelop alatt egyszerre több fejlesztői projektünk is lehet. A KDevelop azonban nem támogatja a CVS által nyújtott valamennyi szolgáltatást, többek között a fájlmegetkintési és -szerkesztési jogokat sem.

Felhasználói leírás

A KDevelop a kdcc és a doxygen rendszeren keresztül program API-leírások készítésére is képes. Ezalatt a felhasználó folyamatosan láthatja a felhasználói API-leírást. Különösen a nagy, többfejlesztős projektek esetén igen hasznos ez a szolgáltatás. Ha projektkészítéskor a felhasználói leírást kiválasztottuk, akkor a felhasználói kézikönyv HTML-ürlapja önműködően elkészül. Innentől már csak a felhasználótól függ, kitölti-e az adatokat valamilyen HTML-szerkesztővel.

© Kiskapu Kft. Minden jog fenntartva

Támogatottság

A nyílt forráskódú projektek egyik leggyakoribb hátránya az alacsony támogatottság. Néha a projekt meg is szakad – így tulajdonképpen lehetetlenné válik a kapcsolatteremtés, ha gondok merülnek fel vagy hibákat találunk. A KDevelop azonban szerencsére pezsgő levelezőlistával rendelkezik, melyet néhány fejlesztője folyamatosan felügyel. A közvetlen leírások széles választékával a KDevelop olyan szintű támogatottságot tudhat magáénak, amellyel még a legtöbb kereskedelmi termék sem versenyezhet.

Hiányzó kívánások listája

Annak ellenére, hogy a KDevelop hatékony és hasznos eszköz, néhány felhasználási terület még hiányos vagy fejlesztésre szorul:

- Kívánatos volna egy okos szövegszerkesztő, amely önműködően befejezné a kódot, például kitöltené az adott függvény értékeit.
- A KDevelop 1.4 nyelv támogatása mindössze a C++ és C alkalmazásokra, valamint gcc/g++ fordítóra korlátozódik.

- Más GUI-szerkesztők (például a Gnome GUI-szerkesztő, a Glade) támogatottsága lehetne jobb.
- Egy létező projekt KDevelopba illesztése nem éppen egyszerű feladat.
- Nincsenek Rapid Application Development (RAD) -elemek, amelyek adatbázis-kapcsolatot szolgáltatathatnának, és alapját képezhetnék a vállalati szintű fejlesztéseknek.

Mivel a KDevelop nyílt forráskódú program, ezek a hiányzó lehetőségek hamarosan talán nem jelentenek gondot.

Összefoglalás

A KDevelop egy középszintű kereskedelmi IDE képességeivel rendelkezik. Igen jól illeszkedik a linuxos felülethez, számos nyílt forráskódú eszközt kihasznál, és olyan szintű támogatottsággal bír, amit nehéz túlszámolni. Ennek ellenére van még hely a további fejlesztések számára. A KDevelop képességei megfelelően elégitik ki a kis- vagy közepes méretű projektek és fejlesztőcsapatok fejlesztőkörnyezet-igényeit.

Kapcsolódó címek

KDevelop honlap ➔ <http://www.kdevelop.org/>
 Linux RPM-csomagok jegyzéke ➔ <http://www.rpmfind.net/>
 Nyílt forráskódú programok tájékoztató jegyzéke ➔ <http://www.freshmeat.net/>
 A Qt, Qt Designer és a Qt Linguist készítői ➔ <http://www.trolltech.com/>
 KDE honlap ➔ <http://www.kde.org/>

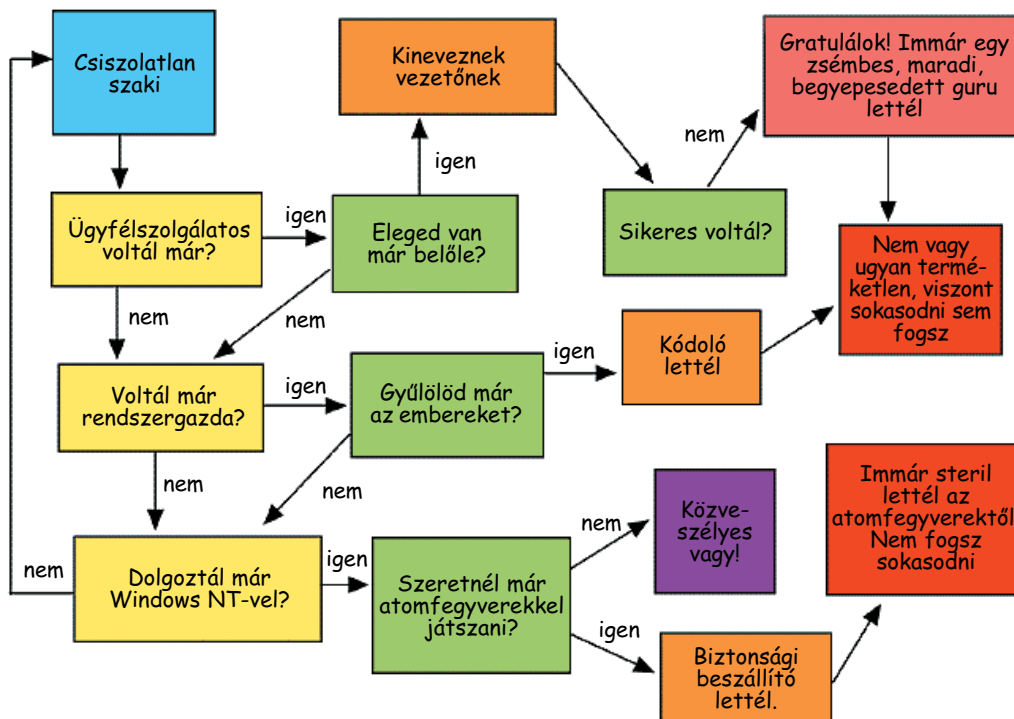


Petr Sorfa

(petrs@sco.com) a Santa Cruz Operation's Development Systems Group tagja, ahol cscope és a Sar3D nyílt forráskódú projektek felelőseként tevékenykedik. Főiskolai végzettséget szerzett a Cape Town és a Rhodes University-n. Érdeklk a nyílt forráskódú fejlesztések, a számítógépes grafika, a fejlesztőrendszer, továbbá a képregény-rajzolás (sequential art).

A szaki-karrier folyamatábrája

Itt kezd!



COPYRIGHT(C) 2000 ILLIAP

HTTP://WWW.USERFRIENDLY.ORG/