



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Kylix 2 OE, Debian y componentes open source (33441 lectures)

Per **Mateu Batle Sastre**, [Mateu](http://www.mbatle.com/) (<http://www.mbatle.com/>)

Creado el 25/11/2001 23:20 modificado el 25/11/2001 23:23

*Kylix es el entorno de desarrollo de [Borland](#)⁽¹⁾ (o Inprise) para Linux. Es un producto comercial aunque hay una versión **Open Edition** gratuita (sólo binarios) que puede usarse para desarrollar proyectos GPL. La principal desventaja de esta versión es que se echan en falta algunos componentes esenciales para el desarrollo de aplicaciones comunes, acceso a base de datos, servicios Internet, etc. En este artículo se pretende dar a conocer esta herramienta, explicar como contrarrestar sus limitaciones iniciales y sacarle el máximo partido.*

Instalación Kylix 2 Open Edition

Para empezar debemos registrarnos en [Borland](#)⁽²⁾ para descargar la versión Open Edition y contestar un pequeño cuestionario de cortesía. Posteriormente recibiremos un e-mail de Borland con los datos para activar la Open Edition. El fichero de instalación que son unos 30 Mbytes aproximadamente.

La instalación es bastante sencilla, y no tiene mayores complicaciones en una Debian. Sólo he podido testarlo con la Debian sid, pero no he tenido ningún problema, aunque teóricamente no está en la lista oficial de distribuciones soportadas (que incluye Red Hat, SUSE y Mandrake). Para instalar basta ejecutar el *setup* contenido en el fichero **kylix2_open.tar.gz** que nos hemos descargado. Al ejecutar por primera vez Kylix 2 nos pedirá los datos de registro que Borland nos envió por e-mail, y registrará la aplicación por Internet, aunque creo que se puede retrasar el paso de registro y empezar a usar Kylix inmediatamente, aunque no lo he probado. La instalación se puede realizar como usuario normal sin permisos de root, en este caso Kylix se instala en el directorio home del usuario. También existe la posibilidad de instalar Kylix como root en un lugar accesible por todos los usuarios.

Una vez instalado, podemos ejecutar Kylix 2 directamente desde el menú principal del KDE. Ante nuestros ojos aparece el IDE con el editor de formularios, el Object Inspector, la paleta de componentes, etc. Todo es bastante parecido por no decir igual que un Delphi para Windows. Después de la euforia inicial, nos damos cuenta que la Open Edition nos proporciona una paleta de componentes bastante *light* ya que sólo tiene las páginas *Standard*, *Additional*, *Common Control* y *Dialogs*.

Aunque ya se pueden hacer aplicaciones con esto, echamos en falta algunos componentes esenciales para desarrollar aplicaciones, como son el acceso a base de datos y servicios Internet. Kylix 2 en versión Profesional (249 \$) y Enterprise (1999 \$) dispone de componentes de base de datos y muchos otros más, además de mejoras en el IDE.

Instalar FreeCLX

Las [FreeCLX](#)⁽³⁾ son una implementación open source de la librería CLX de Borland. CLX es la nueva librería de componentes cross-platform de Borland incorporada en Delphi 6 y Kylix. La librería usada anteriormente era la VCL, pero estaba más ligada a Windows. El problema es que en la versión Open Edition sólo viene una versión reducida de la CLX, por lo que deberemos usar FreeCLX para disponer de algunas características adicionales.

Para instalar las FreeCLX deberemos descargar la última versión del servidor CVS en [Sourceforge](#)⁽⁴⁾. Son unos 11 Megas de código fuente. Para bajar la última versión del servidor CVS usaremos los siguientes comandos (pulsar ENTER cuando pida la password).

```
cvs -d:pserver:anonymous@cvs.freeclx.sourceforge.net:/cvsroot/freeclx login
```



```

cvs -z3 -d:pserver:anonymous@cvs.freeclx.sourceforge.net:/cvsroot/freeclx co
freeclx

```

A continuación debemos compilar FreeCLX. Para ello podemos seguir los siguientes pasos:

1. Debemos definir la variable de entorno FREECLX al directorio donde hemos descargado. En mi caso he hecho un *export FREECLX=~/src/kylix/freeclx*.
2. Cambiamos el directorio actual a \$FREECLX/qt.
3. Debemos hacer un enlace simbólico en el directorio \$FREECLX/qt/lib a nuestra librería libqt.so.2.2.4. Como mi sistema tenía una versión posterior, en concreto la versión 2.3.1, lo probé con esta y parece que compila bien. El nombre del fichero que espera encontrar el makefile es libqt.so.2.2.4, por lo que o cambiamos el makefile o le damos un nombre simbólico falseado.
4. Ejecutamos make en el shell, que generará la librería libqtinf.so.2.2.4.3. En realidad se debería llamar libqtinf.so.2.3.1 en mi sistema, pero es culpa del makefile.
5. Copiamos la librería generada al directorio de kylix ~/kylix2/bin y actualizamos el enlace simbólico libqtinf-6.5-qt2.3.so para que apunte a esta nueva librería.
6. Ejecutamos Kylix. Vamos al menú Tools | Environment Options | Library y añadimos al principio de la lista de paths a librerías los siguientes directorios:
 - ◆ ~/src/kylix/freeclx/rtl/sys
 - ◆ ~/src/kylix/freeclx/rtl/linux
 - ◆ ~/src/kylix/freeclx/qt
 - ◆ ~/src/kylix/freeclx/clx
7. A continuación cerramos todos los proyectos abiertos en el IDE y vamos a Project | Options | Packages y nos aseguramos que la opción "Build with runtime packages" esté desactivada.

Instalando ZeosDB

[ZeosDB](#)⁽⁵⁾ es una librería open source para acceso a bases de datos desde Kylix. Permite acceso a diferentes sistemas de bases de datos, entre ellos Interbase, PostgreSQL, MySQL, Oracle, Microsoft SQL Server y IBM DB2. Para instalarlo es necesario instalar antes las FreeCLX (o la versión Profesional o Enterprise de Kylix). La instalación es bastante sencilla.

1. Vamos a www.zeoslib.org y descargamos la última versión de ZeosDBO (zeosdbo-5.3.0-beta5.zip en el momento de escribir el artículo). Se descomprime y desde Kylix se abre el fichero ZCommonK.dpk. Se borra el requerimiento dbdesign.dcp, compilamos e instalamos.

Otra posibilidad es coger el código directamente de CVS.

```

cvs -d :pserver:anonymous@cvs.zeoslib.sourceforge.net:/cvsroot/zeoslib
login
cvs -z3 -d :pserver:anonymous@cvs.zeoslib.sourceforge.net:/cvsroot/zeoslib
checkout .

```

2. Ahora abrimos desde Kylix el fichero ZDBwareK.dpk, borramos los requerimientos dbdesign y dataclx, compilamos e instalamos. Esto nos dará como resultado una nueva página con 3 nuevos componentes:



ZBatchSQL, ZMonitor y ZUpdateSQL.

3. Seguimos un proceso similar con ZMySQLK.dpk. Borrarnos dataclx de los requerimientos. Compilamos e instalamos. Obtenemos otros 4 componentes nuevos: ZMySQLDatabase, ZMySQLTransact, ZMySLTable y ZMySQLQuery. En la línea 29 de ZLibMySql.pas se define DEFAULT_DLL_LOCATION que es el path a libmysqlclient.so, debemos modificarlo si no es el correcto.
4. Repetimos el proceso con las demás bases de datos que nos interesen. Para Interbase instalamos el paquete ZibSQLK.dpk (da un error Invalid field registration) a no ser que comenteis la línea que contiene "RegisterFields([TZBCDField]);" en ZibSQLReg.pas, por lo que seguramente no podremos usar campos BCD. Si todo va bien tendremos 6 componentes nuevos: ZibSqlDatabase, ZibSqlTransact, ZibSqlTable, ZibSqlQuery, ZibSqlStoredProc y ZibSqlNotify.
5. De forma parecida instalamos el soporte para PostgreSQL, con el paquete ZPgSQLK.dpk.
6. Como último paso, hay que añadir los directorios zeosdb/, zeosdb/common y zeosdb/dbase. en el path de librerías de Kylix.

Controles db-aware

Con lo que tenemos ahora se pueden acceder fácilmente a las bases de datos existentes. Pero dónde están los controles típicos db-aware: DBEdit, DBText, DBComboBox, ... ? Están implementados en las FreeCLX, pero no han sido registrados en la paleta de componentes. Para ello podemos usar el paquete [dbControls](#)⁽⁶⁾.

Abrimos el paquete Kylix dbControls.dpk, compilamos e instalamos como hemos hecho anteriormente sin ningún problema. Como resultado obtenemos 3 páginas más de componentes:

- Data Access: DataSource, ClientDataSet y DataSetProvider.
- dbExpress: SQLConnection, SQLDataSet, SQLQuery, SQLTable, SQLMonitor y SQLClientDataSet.
- Data Controls: DBGrid, DBNavigator, DBText, DBEdit, DBMemo, DBImage, DBListBox, DBComboBox, DBCheckBox, DBRadioGroup, DBLookupListBox y DBLookupComboBox.

Instalar componentes Internet

Uno de los paquetes de componentes de servicios Internet más conocido es [Indy \(Internet Direct o Winshoes\)](#)⁽⁷⁾ de [Nevrona](#)⁽⁸⁾. Se distribuyen bajo una licencia open source basada en BSD License o Mozilla Public License (a elegir). Este paquete de componentes es muy completo ofreciendo tanto servidores como clientes de la mayoría de protocolos Internet conocidos, incluso creo que se incluye en las versiones de pago de Kylix. Los protocolos soportados son: TCP, UDP, daytime, DNS, echo, finger, FTP, gopher, HTTP, ICMP, ident, IMAP4, IP Multicast, IRC, LPR, NNTP, POP3, QOTD, rexec, rsh, SMTP, SNMP, SNPP, SNPT, syslog, telnet, time, TFTP, whois, chargen, dict, discard, etc. Además hay componentes adicionales para SSL, cookies y codificadores / descodificadores entre otros.

Aunque anuncian que sus componentes son directamente instalables en Kylix, la verdad es que he tenido algunos pequeños problemas pero han sido fáciles de solucionar. La que he intentado instalar es la Indy 9 que aún está en versión beta (9.0.2). También he echado un poco de documentación sobre su instalación en Kylix, pero no he encontrado ninguna referencia.



Para instalar [Indy v9.0.2](#)⁽⁹⁾ nos bajamos la última versión, y a continuación deberemos compilar (no instalar) el indy.dpk. Es posible que tengamos algunos problemas con los nombres de las unidades (mayúsculas / minúsculas), las unidades referenciadas siempre deben tener el mismo case que los ficheros correspondientes. He tenido que cambiar algunas referencias a mano para que compilase correctamente. Es posible que exista alguna opción de configuración para evitar este problema, pero no la he encontrado. Adicionalmente he encontrado un problema en el código en el fichero IdTunnelSlave.pas, *Id := Thread.Handle*, resulta que Handle no existe en TThread. Yo lo cambié por ThreadId sólo para que compilara y proseguir con la instalación, aunque seguramente no sea lo correcto. A continuación sólo resta compilar e instalar el paquete dclindy.dpk.

Una vez instalado nos aparecerán 3 nuevas páginas: Indy Clients, Indy Servers y Indy Misc. Realmente me pierdo con la cantidad de componentes añadidos, afortunadamente existe documentación en HTML.

Enlaces relacionados con Kylix

Aquí cito algunos enlaces de repositorios de componentes y información sobre Kylix (y Delphi).

- [Kylix en Borland](#).⁽²⁾
- [Torry](#)⁽¹⁰⁾. Uno de los mejores y más surtidos repositorios de componentes.
- [VCL Crawler](#)⁽¹¹⁾. Otro buscador de componentes.
- [Club Delphi](#)⁽¹²⁾. Excelente página sobre Delphi.
- [Nexus](#)⁽¹³⁾
- [Open Source dbexpress](#)⁽¹⁴⁾
- [Indy \(Indirect Direct\)](#)⁽¹⁵⁾
- [ZeosDB](#)⁽⁵⁾
- [FreeCLX](#)⁽³⁾
- [Cetus Links sobre Delphi](#)⁽¹⁶⁾
- [Links de About sobre Kylix](#)⁽¹⁷⁾
- [Synapse](#)⁽¹⁸⁾. Otra librería de servicios Internet para Kylix.
- [TPlot](#)⁽¹⁹⁾: componente para hacer todo tipo de gráficos.
- [Delphi Super Page](#)⁽²⁰⁾
- [Delphi Source](#)⁽²¹⁾
- [Delphi Pages](#)⁽²²⁾
- [MT DORB y MT REPORT](#)⁽²³⁾. CORBA y informes en Kylix.
- [HADP](#)⁽²⁴⁾
- [Delphree](#)⁽²⁵⁾: open source initiative.
- [UNDU](#)⁽²⁶⁾: Unofficial Newsletter of Delphi Users.
- [Freepascal](#)⁽²⁷⁾: compilador open source de Object Pascal.
- [Lazarus](#)⁽²⁸⁾: proyecto open source de portar Delphi a Linux (anterior a Kylix).
- [FastReport](#)⁽²⁹⁾
- [Delphi 3000](#)⁽³⁰⁾
- [Delphiseek](#)⁽³¹⁾
- [Delphi City](#)⁽³²⁾
- [Swiss Delphi Center](#)⁽³³⁾
- [Delphi Forum](#)⁽³⁴⁾
- [Delphi Spirit](#)⁽³⁵⁾
- [Kylixnet](#)⁽³⁶⁾
- [IB Phoenix](#)⁽³⁷⁾
- [Morphtek](#)⁽³⁸⁾
- [Dr Bob 42](#)⁽³⁹⁾
- [Kylix.sk](#)⁽⁴⁰⁾

Concluyendo, Kylix 2 Open Edition y los componentes que lo complementan constituyen una alternativa a tener en cuenta en el desarrollo de aplicaciones GPL. Ahora a picar código ;-)

--Mateu

Lista de enlaces de este artículo:



1. <http://www.borland.com/>
2. <http://www.borland.com/kylix>
3. <http://sourceforge.net/projects/freeclx/>
4. <http://www.sourceforge.net/>
5. <http://www.zeoslib.org/>
6. <http://www.ubicum.it/aposoft/download/dbcontrols.tgz>
7. <http://www.nevrona.com/indy>
8. <http://www.nevrona.com/>
9. <http://www.nevrona.com/Indy/download90.html>
10. <http://www.torry.net/>
11. <http://www.vclcrawler.com/>
12. <http://www.clubdelphi.com/>
13. <http://sourceforge.net/projects/nexusproject/>
14. <http://sourceforge.net/projects/open-dbexpress/>
15. <http://www.nevrona.com/Indy>
16. http://www.cetus-links.org/oo_delphi.html
17. <http://delphi.about.com/cs/kylix/index.htm?terms=kylix>
18. <http://www.ararat.cz/synapse/>
19. <http://chemware.hypermart.net>
20. <http://delphi.icm.edu.pl/>
21. <http://www.delphisource.com/>
22. <http://www.delhipages.com/>
23. http://www.millennium-group.ru/index/lang/eng/parent_id/30/level/1
24. <http://www.hadp.org/kylix.htm>
25. <http://delphree.clexpert.com/pages/default.htm>
26. <http://www.undu.com>
27. <http://www.freepascal.org/>
28. <http://lazarus.freepascal.org/>
29. <http://www.fast-report.com>
30. <http://www.delphi3000.com/>
31. <http://www.delphiseek.com/>
32. <http://www.delphicity.net/>
33. <http://www.swissdelphicenter.ch>
34. <http://www.delphiforum.com/>
35. <http://www.delphispirit.com>
36. <http://www.kylixnet.de/eng/index.shtml>
37. <http://www.ibphoenix.com>
38. <http://www.morphtek.com>
39. <http://www.drbob42.com/kylix/>
40. <http://www.kylix.sk/>

E-mail del autor: mbatle_ARROBA_mbatle.com

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=1024>