GNOME, preguntas más frecuentes, PUF (FAQ)

Todd Graham Lewis

David "Gleef" Zoll

GNOME, preguntas más frecuentes, PUF (FAQ)

by Todd Graham Lewis and David "Gleef" Zoll

v1.0.6a Edition

Éstas son las PMF sobre GNOME, el entorno de trabajo en red orientado a objetos GNU ('Network Object Model Environment').

Table of Contents

Chapter 1. Introducción a GNOME

Esta sección abarca preguntas relacionadas con GNOME como concepto global.

1.1. ¿Qué significa GNOME?

GNOME es el GNU 'Network Object Model Environment', entorno de trabajo en red orientado a objetos. GNU significa GNU no es Unix ("GNU's Not UNIX).

1.2. ¿Qué es GNOME?

GNOME es el escritorio GUI del proyecto GNU (http://gnu.org) .

Según un extracto del anuncio original en comp.os.linux.announce, GNOME pretende ser un conjunto completo y gratuito de aplicaciones y accesorios de escritorio intuitivos para el usuario, similar a CDE o KDE pero basado enteramente en software libre.

A mí, el amable responsable de las PMF, me gusta definir GNOME como todo aquello que se espera disponer en un entorno moderno de programación. En este aspecto, es aproximademente equivalente a CDE, win32, OpenStep, o KDE. La gran diferencia es que, a diferencia de los ejemplos mencionados anteriormente, cada componente individual de GNOME es software "libre" o de "código abierto" (http://gnu.org/philosophy/free-sw.html) .

Para los cartógrafos de ahí a fuera, he aquí algunas de las más altas cumbres que encontrarás en la tierra de GNOME:

- GNOME usa la arquitectura universal de agentes de petición de objetos del grupo de manejo de objetos que permite a los componentes software interoperar 'sin costuras', independientemente del lenguaje en el que estén implementados, ¡e incluso de la máquina en la que se ejecuten! (Véase la sección CORBA)

- La comunidad GNOME está trabajando duro en el desarrollo de un modelo de objetos llamado ‘baboon’. Basado en CORBA y similar al enlazador y acoplador de objetos de Microsoft, v2 (OLE2), baboon permitirá a los programadores exportar y/o importar recursos ’componetized’. Ésto, por ejemplo, permitirá a los usuarios utilizar cualquier editor que deseen en su entorno de desarrollo, a condición de que su editor soporte a través de CORBA un interfaz de editor estándar (Véase la sección BABOON)
- GNOME no está atado a ningún gestor de ventanas. Puedes escoger tu gestor de ventanas favorito y usarlo con GNOME. (Véase la sección Gestor de ventanas)
- GNOME usa el kit de herramientas Gimp como kit gráfico para todas las aplicaciones gráficas. GTK tiene toneladas de ingeniosas funcionalidades, pero mis tres favoritas son: (1) soporte para multiples lenguajes, incluyendo C, C++, objective C, scheme, perl, y otros; (2) ’themability’, por la cual un usuario puede cambiar el aspecto de todas las aplicaciones GTK en una máquina mientras las aplicaciones están ejecutándose, lo que es increíblemente interesante; (3) y finalmente, la licencia de GTK es LPGL, lo que significa que es totalmente código abierto, o software "libre", al igual que el resto de GNOME. (Véase la seccion GTK+)
- Además de GTK, GNOME usa imlib, una biblioteca para X de tratamiento de imágenes que soporta multiples formatos de imagenes, desde XPM a PNG, y múltiples resoluciones, desde Color Real de 24-bit hasta B/N de 1-bit, todo de forma transparente al programador. (Véase la sección Imlib)
- Todas las aplicaciones GNOME pueden manejarse a traves de sesiones. Ésto significa, p.e., que si cierras el (todavía no escrito) procesador de textos GNOME y lo arrancas de nuevo, abrirá el documento que tenías abierto previamente y situará el cursor de nuevo en la misma posición. Ésto se logra a través del sistema de gestión de sesiones X, tal y como se implementa en el gestor de sesiones GNOME. (Véase la sección Gestión de sesiones)
- GNOME usa el estándar DocBook SGML para toda la documentación, lo que permite a los programadores escribir la documentación de una forma sencilla. Los documentos pueden verse con el visualizador de ayudas de GNOME, o tranformarse a HTML y verse con un visualizador de web, o convertirse a LaTeX o postscript e imprimirse. (Véase la sección SGML)

- GNOME soporta los métodos de 'localización' e internalización estándar de Uniforum, permitiendo incorporar soporte para nuevos lenguajes incluso sin tener que recompilar la aplicación. (Véase la sección Internationalización & Localización)
- las aplicaciones GNOME soportarán varios protocolos de "arrastar y soltar" (Drag & Drop) para una máxima interoperatibilidad con otras aplicaciones. (Véase la sección Arrastrar y soltar)
- GNOME soporta la programación 3-D usando OpenGL, permitiendo a la gente usar gráficos tridimensionales correctamente con sus aplicaciones GNOME, ya sean éstas cientificas o juegos. GNOME usa Mesa, que es una implementación de código abierto del estándar OpenGL. (Véase la sección OpenGL)

Si todo ésto parece ambicioso, ¡es porque lo es! ¡Sigue leyendo para enterarte de más cosas!

1.3. ¿Dónde puedo encontrar más información, ayuda o trucos acerca de GNOME?

La pagina de Gnome es, por supuesto un buen sitio para empezar: http://www.gnome.org (http://www.gnome.org) .

Puede tambien pensar en unirse a la lista de correo gnome-list, es un muy buen lugar para pedir ayuda.

1.4. Como uso esas listas de correo GNOME de las que he oido hablar?

El mejor lugar para conseguir ayuda e información puesta al dia sobre GNOME es en las listas de correo. Puede encontrar una lista de ellas en: http://www.gnome.org/mailing-lists/index.shtml (http://www.gnome.org/mailing-lists/index.shtml) . La lista de proposito general para ayudas y discusiones es gnome-list@gnome.org (mailto:gnome-list-request@gnome.org) . Para apuntarse a esta (o cualquier otra) lista, envie un mensaje a *request address* , esta es en general la direccion de la lista con "-request" añadido al nombre. Por ejemplo , this is generally the list address with "-request" added to the name. For example the request list for gnome-list is gnome-list-request@gnome.org (mailto:gnome-list-request@gnome.org) . The subject of the email *must* be **`subscribe`** .

En pocos minutos (posiblemente un poco más si no estás en Norte América), deberías obtener una respuesta automática pidíendote la confirmación de la suscripción. Sigue las instrucciones de este correo electronico y estarás suscrito. Finalmente, el sistema te mandará dos correos con ayuda sobre como usar el sistema (de listas).

1.5. ¿Cómo accedo a los ficheros de la lista de correo?

Hay dos maneras. La más fácil es ir a la página de la lista de correo en http://www.gnome.org/mailing-lists/index.shtml (http://www.gnome.org/mailing-lists/index.shtml) . Haciendo clic en cualquiera de la lista de direcciones de la izquierda, podrás visualizar los archivos de la lista.

Si necesitas realizar una búsqueda en los archivos, puedes usar la dirección de request (request address) de esa lista. Uno de los primeros correos que te envían cuando te suscribes detalla las instrucciones sobre como realizar esta tarea. Si has perdido estas instrucciones, email a gnome-list-request@gnome.org (mailto:gnome-list-request@gnome.org) con el subject de **`archive help`** . Contestará con instrucciones sobre como usar el sistema de archivos.

1.6. ¿Cómo pronuncio GNOME?

GNOME significa “ GNU Network Object Model Environment (Entorno de trabajo en red orientado a objetos GNU) ” . GNU significa “ GNU’s Not Unix (GNU’ No es Unix) ” , y ha sido siempre pronunciado “ guh-NEW ” para minimizar la confusión. Puesto que GNU es el primer nombre de GNOME, GNOME se pronuncia oficialmente “ guh-NOME ” .

Sin embargo, mucha gente pronuncian GNOME como “ NOME ” (como esos pequeños seres de leyenda, los gnomos), nadie te va a pegar si encuentras esta pronunciación más fácil.

1.7. ¿Por qué GNOME?

Primero, el sistema operativo GNU necesita un entorno de escritorio que sea 100% free software (software libre). GNOME satisface esta necesidad.

La segunda meta, por medio de la primera, es que la gente que use sistemas operativos libres tenga un agradable entorno de usuario en el que trabajar (Esto se aplica también a otros sistemas basados en UNIX no gratuitos).

Finalmente, GNOME tiene un montón de características realmente bien diseñadas que seguro que los ’geeks’ disfrutarán. En palabras de uno de los revisores internacionales de estas FAQ:

> Todd, GNOME tiene méritos tecnicos incluso más allá del "open source" y el "agradable entorno" y la "dominación mundial", como tú bien sabes...Por ejemplo, el clon de COM y el uso de CORBA para la inter-operación gives me a woody.

Esto es verdaderamente cierto (el hecho de que GNOME tenga un gran mérito técnico; descozco el aspecto en el woody front). ¡Pruébalo tu mismo y verás!

1.8. ¿Por qué no CDE o KDE o GNUStep?

¡Porque GNOME es mejor! Ya más enserio, cada uno de estos entornos plantean cuestiones que nos animaron a comenzar (to start fresh) con el proyecto GNOME.

CDE no es Free Software, ni de lejos se parece. Mucha gente además lo encuentra carente de características especiales, rendimiento y funcionalidad.

En el momento en el que surge GNOME, KDE tenía serios problemas de licencia, que todavía están intentando solucionar. KDE está basado en una librería no libre llamada Qt, que tiene una licencia muy liberal, pero que no es Free Software (http://gnu.org/philosophy/free-sw.html) . TrollTech (la empresa propietaria de Qt) ha prometido cambiar la licencia de Qt, pero todavía no lo han hecho. Además habían problemas de compatibilidad entre la licencia de KDE y la de Qt. Mejor que insistir en este tema, es leer Posición de RedHat sobre Qt (http://charlotte.redhat.com/redhat/qtlicense.html) y Posición de Debian sobre KDE (http://www.debian.org/News/1998/19981008) . A la gente de GNOME nos gusta la gente de KDE, y consideramos esto una situación desafortunada en proceso de ser arreglada. Con un poco de suerte, esto dejará pronto de ser un tema actual, y GNOME y KDE podrán competir amigablemente en méritos técnicos y diseño.

Este tema ha sido hashed out una y otra vez en la lista de correo general de GNOME. Plantear esta cuestión en la 'gnome-list' es un comportamiento totalmente deasconsejado. Si quieres ir a algún sitio y empezar una flame war sobre este asunto, entonces por favor hazlo en algún lugar muy muy lejano donde no tengamos que oirte.

Hay importantes diferencias de diseño entre KDE y GNOME. La primera en la lista es la diferencia en el widget set. Encontramos a GTK+ más agradable, más adaptable, más amigable para los desarrolladores en varios lenguajes, y más flexible que Qt; otros puede que no estén de acuerdo. En conjunto, los proyectos son lo suficientemente diferentes como para que ambos puedan coexistir e incluso colaborar entre sí.

GNUStep es otro entorno de escritorio que tiene muchas cosas buenas que le funcionan muy bien. ***has a lot of good things going for it***. Otro cosa respecto a GNUStep es que están intentando reimplementar el escritorio y API OpenStep, con GNOME preferiríamos desarrollar algo nuevo y bueno más que simplemente rehacer algo viejo, incluso algo viejo y bueno. Además, así como Qt empuja con fuerza al los desarrolladores hacia C++, GNUStep empuja a los desarrolladores hacia Objective C. GNOME quiere permanecer agnostico frente a los lenguajes, y soportar desarrolo en cualquier lenguaje que desees utilizar.

1.9. ¿Qué plataformas?

GNOME fue iniciado por varias personas bien conocidas en las comunidades Linux y GNU, pero está concebido para funcionar en cualquier moderno y funcional sitema basado en Unix. Su funcionamiento ha sido probando en los siguientes:

- GNU/Linux
- BSD (FreeBSD, NetBSD y OpenBSD)
- Solaris
- IRIX
- HP-UX
- AIX
- probablemente en algunos otros; a los que no sigo la pista. ***I have not really kept track.

Si pruebas GNOME en otras plataformas y funciona, entonces por favor asegúrate de hacérmelo saber para que pueda incluirlas aquí. Si lo pruebas en otra plataforma y tienes problemas, entonces por favor envíanos las dificultade que encontraste, para que los desarrolladores de GNOME puedan trabajar en el problema. (Mira en Informando sobre Bugs .)

1.10. ¿Puedo todavía ejecutar mis programs en GNOME?

GNOME no reemplaza tu sistema, funciona sobre él. Todos los programas que funcionan sin GNOME deberían correr bien junto a GNOME.

1.11. ¿Qué es un gestor de ventanas, y qué

tiene que ver con GNOME?

Un gestor de ventanas es un programa dentro del entorno X Windows que controla la posición y apariencia de las ventanas en la pantalla. Mira un xterm o algo. La caja del terminal de texto es el actual xterm, el borde, barra de título, botones y demás estan controlados por el gestor de ventana. Algunos gestores de ventanas tienen aplicaciones extras, pero todos se encargan de la decoración y posición de la ventana.

Hay muchos gestores de ventana ahí fuera, y GNOME debería funcionar correctamente con todos ellos. Hay unas cuantas características de GNOME (como el manejo de sesiones, y el GNOME pager), que no funcionarán con todos ellos.that won't work with all of them. Para obtener los mejores resultados, hazte con un gestor de ventanas compatible con GNOME. Los dos más compatibles actualmente son Enlightenment y IceWM, con FVWM, SCWM, Window Maker, AfterStep y QVWM dando los últimos retoques a la compatibilidad con GNOME mientras lees esto. * finishing touches *****on GNOME-compliance as we speak.

1.12. Has dicho que GNOME es Free Software. ¿Qué significa Free Software?

La mayoría de licencias de software están diseñadas para limitar la libertad de los usuarios a usar, examinar, modificar y distribuir el software. En el proyecto GNOME, y en la comunidad del Free Software en general, creemos que esas restricciones no son éticas, y nos esforzamos por producir software de calidad que esté libre de esas restricciones. Esto es lo que llamamos Free Software. Tradicionalmente, se ha referido a software que permite tres tipos de libertad:

- Libertad para estudiar cómo funciona el programa y adaptarlo a tus necesidades.
- Libertad para redistribuir copias para que puedas compartirlo con tu vecino.
- Libertad para mejorar el programa, y presentar tus mejoras al público, para que toda la comunidad se beneficie.

Más recientemente, una cuarta libertad ha comenzado a ser restringida por las licencias, la libertad para usar el software. Debido a que las licencias de software están basadas el la ley de propiedad (copyright), y esta ley sólo se dirige a la distribución, no al uso, asumimos que la la libertad de uso estaba implícita. law only addresses distribution, not use, we assumed that freedom of use goes without saying. *****Suffice it to say que mientras la libertad de uso no es parte de la definición tradicional del Free Software, el concepto sí incluye esa libertad.

* *Note to translators, please translate the following note, including all instances of the word "free" carefully, except leave the* `free` *in English.*

> **Note:** Si el siguiente párrafo es difícil de entender en la traducción, perdonadme. A diferencia de muchos idiomas, en inglés la palabra usada en " free beer (cerveza gratuita) " y " free speech (libertad de expresión****) " es `free` . Esto causa alguna confusión cuando se está hablando de Free Software.

El Free Software es un asunto de libertad, no de precio. Para entender el concepto, piensa en "Free Speech" y no en "Free Beer". La comunidad Free Software no está de ningún modo en contra del concepto de vender copias del software, o de que los programadores se ganen la vida. No estamos en contra del software comercial, estamos en contra del software propietario. Para más información, mira los artículos en http://gnu.org/philosophy (http://gnu.org/philosophy) .

En el proyecto GNOME project, protegemos la libertad del usuario usando la GNU General Public License (GPL) y la GNU Lesser General Public License (LGPL, anteriormente conocida como la GNU Library GPL). Estas licencias son cuidadosamente diseñadas con el legal advice para producir un " copyleft " . Que es un trabajo con copyright que garantiza que será siempre redistribuible libremente. Esto se consigue asegurándonos que las modificaciones y trabajos derivados están también cubiertos bajo la GPL.

Las principales aplicaciones de GNOME (y unas pocas librerías) están cubiertas bajo la GPL. La mayoría de las librerías core de GNOME usan la LGPL, para permitir que puedan ser utilizadas otras licencias de software libres (y propietarias) cuando se realize software para GNOME, aunque siempre protegiendo las librerías de ser restringidas.

1.13. ¿Es GNOME Open Source(tm) Software?

En primer término se puede decir que sí. La GPL y LGPL usadas para GNOME están explicitamente listadas como ejemplos de la definición de "Open Source". Sin embargo, intentamos evitar usar el término Open Source para describir a GNOME, ya que no sólo esconde este término deliberadamente la libertad inherente en el Free Software, sino que además el término está siendo utilizado para describir software que no es Free (Software).

El término fue creado para intentar acercar el Free Software a los negocios que no querían oir hablar de Libertad (Freedom). Personalmente, he descubierto que a la mayoría de negocios les gusta oir hablar sobre libertad. La libertad para personalizar su software a sus necesidades, la libertad para cambiar a los desalloradores de software sin cambiar su software, la libertad para dar copias de su software a sus clientes y subcontratistas, son todas estas libertades que el Free Software ofrece, y el propietario normalmente no, las que hacen un *good business sense* . Para saber más sobre la distinción entre Open Source y Free Software, le el artículo de Richard M. Stallman en http://gnu.org/philosophy/free-software-for-freedom.html (http://gnu.org/philosophy/free-software-for-freedom.html) .

Chapter 2. Consiguiendo GNOME

Hay muchas maneras de introducir GNOME en tu sistema. Esta sección cubre las preguntas más comunes que implica el proceso.

2.1. ¿Cuáles son los Requisitos de Sistema para GNOME?

Actualmente, necesitas una máquina con un sistema operativo Unix (o basado en Unix) instalado, con el Sistema X Window (X11R5 o superior). Necesitas al menos 16MB de RAM, aunque supongo que serás más feliz con 32MB o más. Si sólo estás instalando los binarios, necesitarás cerca de 30MB de espacio en el disco, no esperes necesitar menos 200MB de espacio de disco si vas a compilarlo desde los fuentes.

2.2. ¿Cuál es el mejor formato para conseguir GNOME en mi máquina?

Depende de la clase de ordenador que tengas, y lo que quieras hacer con GNOME. Si tienes dudas, el método oficial y con soporte para instalar GNOME es desde el último paquete de tarballs (los ficheros acabados en ****the officially supported way to install GNOME is from the latest set of tarballs (the files ending in ***** “ .tar.gz ”) desde ftp://ftp.gnome.org (ftp://ftp.gnome.org) o sus servidores réplica (mirrors). Busca en el directorio /pub/GNOME/sources para comprobar los diversos paquetes existentes. La lista completa de sitios FTP réplica está en http://www.gnome.org/ftpmirrors.shtml (http://www.gnome.org/ftpmirrors.shtml) , para una emergencia, prueba con ftp://gnomeftp.wgn.net/pub/gnome (ftp://gnomeftp.wgn.net/pub/gnome) .

****If you want the up to the minute, bleeding edge development version of GNOME, CVS is the way to go. Nadie te asegura No hay garantías de que la versión CVS pueda como mínimo ser compilada, ni mucho menos que funcione, pero los bug reports son muy apreciados. Si te parece qeu esto es para tí, puedes encontrar mas información sobre como obtener la versión CVS en http://www.gnome.org/devel/whatiscvs.shtml (http://www.gnome.org/devel/whatiscvs.shtml) y http://www.cinternet.net/~soren/gnome/cvs.shtml (http://www.cinternet.net/~soren/gnome/cvs.shtml) .

Si no quieres ocuparte de compilar GNOME, y estás en un sistema común, deberían haber paquetes binarios precompilados disponibles para tí. Actualmente hay paquetes para sistemas RedHat, SuSE 6.0, Debian y LinuxPPC. Hasta hace poco habían paquetes para Solaris packages también, tan pronto como tengamos unos buenos ofreceremos estos también. Más sobre las diversas distribuciones en http://www.gnome.org/start/getting.shtml (http://www.gnome.org/start/getting.shtml) .

2.3. ¿Qué necesito en mi ordenador antes de instalar GNOME?

Hay unas pocas cosas con las que GNOME cuenta que tienes, si no las tienes, o tienes una versión más antigua, instala o actualiza los siguientes paquetes antes de comenzar el proceso de instalación de GNOME.

libjpeg (6b o superior)
libpng (1.0.1, 1.0.3 o superior *no 1.0.2*
libungif (3.0 o superior)
libtiff (3.4 o superior)
libgr (2.0 o superior)
ImageMagick (4.0.5 o superior)
zlib (1.1.2 o superior)
guile (1.2 o superior)
Berkeley DB (1.85 o superior)

Además, hay unas pocos librerías que, aun no siendo tecnicamente una parte de GNOME, son habitualmente distribuidas con él. Asegúrate de que tienes:

glib (1.2 o superior)
gtk+ (1.2 o superior)
ORBit (0.4.0 o superior)
Imlib (1.9.3 o superior)

Algunos de los bells and whistles requieren un gestor de ventanas compatible con GNOME para funcionar. Actualmente estos son Enlightenment, IceWM, y Window Maker. Además tanto FVWM como SCWM están dando sus últimos retoques a su soporte para GNOME. GNOME no incluye un gestor de ventanas, y funcionará correctamente con cualquier gestor de ventanas, estos simplemente aportan unos pocos extras.

Últimamente, hay algunas librerías que son necesarias para unos pocos, no críticos programas en GNOME. Si te estas quedando si espacio o tiempo, puedes saltar esto sin ningún riesgo. Si quieres que suenen todas las bells and whistles, querás también lo siguiente:

libgtop (0.99 o superior)
esound (0.2.8 o superior)
gtk-engines

2.4. ¿En qué orden me ocupo de los paquetes de GNOME?

No hay una única respuesta para esta cuestión, pero hay un orden que debería funcionar bien:

glib
gtk+
imlib
orbit
gtk-doc
libgtop
gtk-engines

gnome-libs
gnome-core
control-center
gnome-print
mc
gnome-guile
gnome-python
gnome-objc
gtk--
anything else, as needed, in whatever order

2.5. ¿Como consigo los Tarballs de GNOME?

Las nuevas versiones de los Tarball (p.e. archivos comprimidos del código usando tar y gzip) de GNOME aparecen en ftp://ftp.gnome.org (ftp://ftp.gnome.org) bajo el directorio /pub/GNOME/ (ftp://ftp.gnome.org/pub/GNOME) . Estos archivos son entonces copiados en los servidores FTP réplica, listados en http://www.gnome.org/ftpmirrors.shtml (http://www.gnome.org/ftpmirrors.shtml) . Si tanto el sitio ftp y web están caídos, un mirror relativamente seguro está en ftp://gnomeftp.wgn.net/pub/gnome (ftp://gnomeftp.wgn.net/pub/gnome) .

Estos tarballs son las únicas versiones oficiales de GNOME, y la única versión de la que los desarrolladores tienen garantizado un control directo ***and the only version that the developers can be guaranteed to have direct control over. Actualmente, las versiones más recientes de cada paquete pueden encontrarse en sources/latest (ftp://ftp.gnome.org/pub/GNOME/sources/latest) , el actual grupo de tarballs para la version GNOME-1.0 están en gnome-1.0/sources (ftp://ftp.gnome.org/pub/GNOME/gnome-1.0/sources) .

Si quieres los últimos tarballs en desarrollo de GNOME, Jim Pick Software (http://jimpick.com) envía snapshots diarias del árbol de desarrollo en formato tarball ***of the development tree in tarball form. Puedes conseguirlos en ftp://ftp.jimpick.com/pub/gnome/snap (ftp://ftp.jimpick.com/pub/gnome/snap) .

Generalmente ****he keeps a few key days of each package on file, and diff files so that you can recreate the snapshots of any other day. Note that these are not releases, they are unstable snapshots, and should not be used in a production system unless you know exactly what you are doing.

2.6. ¿Cómo consigo GNOME desde el CVS?

El CVS es una herramienta muy útil, no solo para los desarrolladores, sino también para los usuarios que quieran tener acceso a lo último de lo último ***up-to-the-second*** del árbol del desarrollo. ***However, using CVS is a little involved for this section of the FAQ. Dos buenos sitios para empezar son la sección " What is CVS? (¿Qué es el CVS?) " del sitio web de GNOME en http://www.gnome.org/devel/whatiscvs.shtml (http://www.gnome.org/devel/whatiscvs.shtml) , y la siguiente sección Cómo hacerse un GNOMEr .

2.7. Mi único acceso a internet es el correo electrónico. ¿Cómo consigo GNOME?

Si estás tan desconectado de internet que sólo tienes acceso al correo, existen remailers en la red que te permiten usar el FTP via correo e. Envía un mensaje escribiendo en el cuerpo **help** , sin comillas o signos de puntuación a bitftp@pucc.princeton.edu o webmail.ucc.ie. Puede ser lento, pero te permitirá bajarte los tarballs de GNOME via email. Visita http://www.csdl.tamu.edu/~cchung/cpsc689/ftpmail/ftpmtoc.html (http://www.csdl.tamu.edu/~cchung/cpsc689/ftpmail/ftpmtoc.html) o http://www.cix.co.uk/~net-services/mrcool/stats.htm (http://www.cix.co.uk/~net-services/mrcool/stats.htm) para conseguir una lista actualizada de los sitios FTPMail disponibles.

Los sitios FTP a los que tendrías que acceder están detallados más arriba, en " ¿Como consigo los Tarballs de GNOME? " , simplemente tendrías que utilizar el interfaz de FTPMail más que hacer directamente un FTP al sitio. ***** Aquí no se si lo que quiere hacer es resumir la diferencia entre lo que seria usar un FTP y un FTPMail you would just use the FTPMail interface rather than directly FTPing to the site.

2.8. Yo utilizo RedHat Linux, ¿dónde consigo los RPMs?

De forma periódica, RedHat Labs produce los RPMs de las versiones de GNOME para RedHat Linux, y los ponen en su web. La última versión está actualmente en gnome-1.0/redhat (ftp://ftp.gnome.orgpub/GNOME/gnome-1.0/redhat) . Se crean para plataformas alpha, sparc y i386. Asegúrate también que consigues los RPMs del directorio noarch, sea cual sea la plataforma en la que estás. Puedes encontrar instrucciones paso a paso para estos RPMs en http://www.gnome.org/start/getting_rh.shtml (http://www.gnome.org/start/getting_rh.shtml) .

Estos RPMs deberían funcionar correctamente en Mandrake Linux, y también en las distribuciones MacMillian y CheapBytes. Es muy posible que estos RPMs tengan problemas con Caldera, S.u.S.E., y otras distribuciones basadas en RPM; sigue leyendo para obtener más ayuda para estos sistemas.

Estos RPMs están compilados con soporte para sonido, lo que es fantástico a menos que no tengas tarjeta de sonido en tu sistema. Si alguien quiere mantener una distribución de RedHat Linux compilada *sin* soporte para sonido, con mucho gusto lo pondremos en el FTP también.

2.9. Yo utilizo Debian GNU/Linux, ¿cómo consigo los paquetes .deb?

* *¡Gracias Havoc! :-)*

Los paquetes Debian más actulizados son distribuidos en los sitos Debian (y mirrors) en debian-unstable (http://www.debian.org/Packages/unstable/x11/) . Instrucciones paso a paso en http://www.gnome.org/start/getting_debian.shtml (http://www.gnome.org/start/getting_debian.shtml) .

Para obtener los paquetes Debian, haz lo siguiente:

```
$

apt-get update


$


apt-get install gnome-control-center gnome-panel gnome-
session gnome-games gnome-terminal gnome-utils
```

Los maintainers de los paquetes Debian son voluntarios, y siempre prueban las cosas a fondo antes de lanzar una versión, por lo que si no ves la última versión en formato deb en seguida, por favor se paciente. Mejor aún, si quieres ayudar a preparar los paquetes Debian para su distribución, hay una lista de correo en debian-gtk-gnome@lists.debian.org (mailto:debian-gtk-gnome@lists.debian.org) , e instrucciones sobre cómo acceder a la staging area en http://www.debian.org/~jim/debian-gtk-gnome/README (http://www.debian.org/~jim/debian-gtk-gnome/README) .

Debido a que los paquetes Debian de GNOME ya están siendo distribuidos con Debian, actualmente no replicamos aquí. No parece haber necesidad de ello. Si existiera una necesidad real de replicar estos paquetes, simplemente háznolo saber y lo haremos.

2.10. Yo utilizo S.u.S.E. Linux, ¿cómo consigo los paquetes RPM?

Hay una excelente página de GNOME para SuSE en http://gnome.linuxbe.org (http://gnome.linuxbe.org) (Bélgica) y http://www.tu-harburg.de/skf/Pub/ifmpc118.ifm.uni-hamburg.de/gnome.html (http://www.tu-harburg.de/skf/Pub/ifmpc118.ifm.uni-hamburg.de/gnome.html) (Alemania). Actualmente tienen las instrucciones y los RPMs para SuSE v6.0, con la v5.3 de los RPMs en camino. Visítala para más información.

2.11. Yo uso Caldera OpenLinux, u otras distribuciones de Linux basadas en RPM, ¿cómo consigo los paquetes RPM?

Simplemente porque dos distribuciones usen RPM no quiere decir que puedan usar los mismos RPMs, por ejemplo, RedHat y Caldera usan distintas versiones de libc, por lo que probablemente los binarios RPM de RedHat no funcionarán en Caldera. Un truco que puede funcionar es coger los SRPMS de la distribución de RedHat, y usar **rpm –rebuild** en ellos, y entonces instalar los RPMs binarios resultantes. De otra manera, probablemente tengas que seguir las instrucciones de los tarballs.

2.12. Yo utilizo FreeBSD, ¿cómo puedo conseguir GNOME via Ports?

* *¡Gracias Stanislav Shalunov! :-)*

FreeBSD tiene portes y paquetes para varias aplicaciones de GNOME (incluyendo Gnumeric) así como un porte para el proyecto GNOME entero que instalará todos los programas específicos de GNOME via dependencias. Si tienes el CDROM montado bajo `/cdrom` (el nombre del punto de montaje *es* importante) o tienes una conexión de internet en marcha y el tiempo para bajar durante un rato, puedes instalar GNOME en tu FreeBSD box (3.1 o superior, posiblemente inferior también) escribiendo:

```
$
cd /usr/ports/x11/gnome && make install
```

Para saber más sobre la Colección de Portes de FreeBSD puedes visitar http://www.freebsd.org/ports (http://www.freebsd.org/ports) . Si no tienes conexión a internet, pero tienes una distribución de docs ***but do have a docs distribution, puedes también encontrar más información en `/usr/share/doc/handbook23.html` . Este documento también te explicará como instalar GNOME con paquetes pre-built (binarios) usando **pkg_add** , si tienes el CDROM. Actualmente (para la 3.1-RELEASE), necesitarás instalar los siguientes paquetes: `gnomelibs` , `gnomeaudio` , `gnomecore` , `gnomegames` , `gnomeprint` , `gnumeric` , y `libgtop` . Un listado de las últimas versiones disponibles se encuentra en http://www.freebsd.org/ports/gnome.html (http://www.freebsd.org/ports/gnome.html) .

2.13. Yo uso Solaris, AIX, HP-UX, BSDI, SCO, IRIX, u otros sitemas Unix o similares, ¿puede conseguir los paquetes binarios?

Existían unos paquetes binarios para Solaris que alguien hizo, pero los paquetes parece que han desaparecido (de todas formas era una versión atigua de GNOME). Si aparece un buen conjunto de paquetes binarios para alguno de estos sistemas, probablemente los podamos poner en el sitio FTP.

En ausencia de una versión binaria, vas a necesitar usar los tarballs. Si tu sistema no tiene un compilador, el GNU Project (http://gnu.org) ofrece el GNU C Compiler (gcc) Libremente (for Free) ****on pretty much every platform out there. Ve a http://gnu.org/software/gcc/gcc.html (http://gnu.org/software/gcc/gcc.html) para más información.

2.14. Yo utilizo Microsoft Windows, ¿puedo conseguir GNOME?

* *¡Gracias David Wheeler!*

Actualmente la respuesta es no, GNOME no ha sido portado a la la plataforma Windows, hay ya mucho trabajo como para hacer ese porte posible. ****there is an effort going on to make such a port possible. Si realmente necesitas una versión para Windows, cuenta con hacer mucho trabajo y ***count on doing lots of work and coordinating with related efforts.

Tor Lillqvist ha estado portando varias librerías gráficas usadas por GNOME a la plataforma Win32 (Microsoft Windows 95/98/NT). Puedes encontrar más información sobre sus esfuerzos en http://www.gimp.org/~tml/gimp/win32 (http://www.gimp.org/~tml/gimp/win32) . Ya ha portado glib, gtk+, gdk_imlib, y unas pocas otras librerías relacionadas con la generación de gráficos a Win32.

Su meta es hacer funcionar Gimp totalmente en la plataforma (ya funciona parcialmente), pero su trabajo pone a GNOME mucho más cerca de ser portable a Windows. Varias aplicaciones de GNOME necesitan sus librerías (posiblemente con un poco más de debugging), libpng, gnome-libs y ORBit. Si alguien portara lo anterior, ****If someone were to port these, muchas aplicaciones de GNOME podrían ser compiladas y ejecutadas en Windows.

Si estás interesado en trabajar en dicho proyecto, asegúrate de leer cuidadosamente la documentación de Autoconf (http://gnu.org/software/autoconf/autoconf.html) , ya que así es como diferenciamos entre diferentes capacidades de sistema en GNOME. También, presta atención a las librerías y utilidades de Cygnus Solution’s CygWin32 (http://sourceware.cygnus.com/cygwin/) . Harán esta tarea mucho más fácil.****They will make such a task much easier.

Chapter 3. Compilando GNOME

3.1. ¿En qué orden compilo los paquetes?

Es importante que compiles e instales los paquetes de GNOME en los ordenes correctos. Hay algunas dependencias muy importantes dentro del árbol de GNOME. Este es el orden que sugiero, en el que deberías compilar e instalar cada paquete antes de continuar con el siguiente. En sistemas basados en ELF (como GNU/Linux y FreeBSD), deberías ejecutar también **ldconfig** después de cada paquete, simplemente para asegurarte de que todas las nuevas librerías son detectadas.

glib
gtk+
imlib
orbit
gtk-doc
libgtop
gtk-engines
gnome-libs
gnome-core
control-center
gnome-print
mc
gnome-guile
gnome-python
gnome-objc
gtk–
anything else, as needed, in whatever order

3.2. ¿Como configuro el código fuente para mi

sistema? (o... ¿Dónde se esconden los Makefiles?)

GNOME usa configure (generado por Autoconf) para configurar el código fuente para ser compilado correctamente en tu sistema ***to set up the source code to compiler properly on your system. Configure investigará como está configurado tu sistema, y definirá constantes de compilación para añadir o quitar diversos trozos de código según sea necesario. ****Configure will investigate how your system is set up, and define compiler constants to add or remove various pieces of code as needed.

Configure acepta varias opciones, escribe **./configure –help** para una lista completa. La más importante es **`–prefix`** , que le dice al sistema donde se va a instalar GNOME. Si omites el prefix, configure asume que lo harás en `/usr/local` . La mayoría de RPMs usan el prefix `/usr` . Muchos usuarios prefieren usar el prefix **`/opt/gnome`** , lo que requiere unos retoques previos, ***which requires a little advance preparation, descritos más abajo.

Configure también chequea ciertas variables de entorno. Pon tus CFLAGS a igual sean cuales sean las opciones de compilación que deseas para asegurarte que las pasas, por ejemplo ***to equal whatever compiler options you want to make sure get passed, for example **`–O6`** maximizará la optimización, **`–g`** incluirá signos de debugging, etc.

3.3. ¿Dónde se esconde configure?

Estás obteniendo el código del CVS, ¿verdad? ****You're getting your code from CVS, aren't you? Lee la sección Becoming a GNOMEr que se encuentra un poco más arriba. Si conseguiste un tarball, configure debería estar en el directorio superior del fuente. Si no está ahí, eso es un bug. Mira la pregunta ¿Cómo informo de un bug? .

3.4. De acuerdo, ahora tengo un Makefile.

¿Cómo consigo instalar GNOME en mi sistema?

Desde aquí, todo es bastante sencillo. Escribes **make** , seguido de **make install** . En muchos sistemas (incluido GNU/Linux), esto debería ser seguido de **ldconfig** . Si tienes mucha prisa, a veces puedes saltar **make** , ya que **make install** normalmente hará todo lo que no halla sido hecho todavía. Si quieres empezar todo el proceso y olvidarte de ello, y usas la Bourne shell o sus derivadas (p.ej. ksh o bash), pon los cuatro comandos en una línea separados por `&&` . Por ejemplo:

```
$

./configure –
prefix=/usr/local && make && make install && ldconfig
```

El `&&` se asegura de que el primer comando ha finalizado correctamente antes de seguir con el siguiente.

Una vez que has hecho **make install** (y **ldconfig** si fuera necesario), puedes continuar con el siguiente paquete.

3.5. ¿Cómo pongo GNOME en algún sitio especial, como `/opt/gnome` ?

A mucha gente le gusta poner GNOME a parte, en su propio directorio. ****Many people like to put GNOME off by itself, in its own directory. Esto hace que sea más fácil de desinstalar ante una emergencia, más fácil de controlar quien lo usa, te permite exportar el directorio de gnome a otros sistemas, etc, etc. Hacer esto requiere una cierta preparación, y conocimiento de tu sistema. En este caso, pondré GNOME en el directorio `/opt/gnome` . Simplemente reemplaza `/opt/gnome` con cualquier otro directorio que quieras.

La cuestión principal ***the biggest issue*** es asegurarse de que el sistema puede encontrar los binarios de GNOME en `/opt/gnome/bin` , las librerías en `/opt/gnome/lib` , y los archivos man en `/opt/gnome/man` .

Para decirle al sistema que busque en `/opt/gnome/bin` para encontrar los binarios de GNOME, simplemente asegúrate de que está incluido en la variable de entorno PATH . por ejemplo: **`PATH="/bin:/usr/bin:/opt/gnome/bin"`** .

Después, normalmente hay que indicar al ***loader donde se encuentran las librerías. En la mayoría de Unix comerciales, añades `/opt/gnome/lib` a la variable de entorno LD_LIBRARY_PATH , de la misma manera que anteriormente al PATH . En algunos sistemas, en vez de la anterior se usa LD_RUN_PATH . En los sistemas que usan binarios en formato ELF ***ELF format binaries, como GNU/Linux o FreeBSD, añades el path de la librería al archivo `/etc/ld.so.conf` , y ejecutas **ldconfig** . En sistemas estáticamente linkados ****On statically linked systems, no tienes que preocuparte de nada de lo anterior, sino comprar más espacio de disco, que seguro que necesitarás. Consusta la documentación de tu sistema si no estás seguro de como decirle al ***loader en que directorios buscar.

También deberías indicarle a man donde encontrar las páginas man instaladas por GNOME. Simplemente inserta una línea en `/etc/man.config` , donde ponga **`MANPATH /opt/gnome/man`** .

La otra cuestión es que hay un par de directorios en los que GNOME pondrá cosas, y que estarían mejor en cualquier otro lugar. ***The other issue is there are a couple of directories which GNOME will put things in, which would be better put elsewhere. La manera más sencilla de resolver esto es mediante links simbólicos. Escribe lo siguiente:

```
$

mkdir /opt/gnome/share


$


cd /opt/gnome/share
```

```
$

ln -s /usr/share/locale locale

$

ln -s /usr/share/aclocal aclocal
```

Ahora ya está todo preparado para usar `/opt/gnome` como ***prefix al compilar. Para hacer esto, simplemente da en el comando **./configure** la opción `-prefix=/opt/gnome` . Por supuesto, si quieres algún sitio especial diferente a `/opt/gnome` , simplemente reemplazalo con en directorio a utilizar para todo lo mencionado anteriormente.

3.6. ¿Cómo compilo GNOME con todos los ficheros en el lugar correcto para ser Compatible con FHS? ¿Cómo recoloco los directorios etc y var ***to easily network GNOME?

* *¡Gracias Raja R. Harinath!*

Por defecto, GNOME pone todos sus archivos bajo su ***prefix directory. Aunque que esto hace que GNOME sea fácil de aislar, no es necesariamente el mejor lugar para ciertos archivos en algunas situaciones. En particular, si buscas conectar GNOME, o para una estricta compabilidad con ***if you are looking to network GNOME, or for strict FHS (http://www.pathname.com/fhs/2.0/fhs-toc.html) compliance, puede que quieras recolocar algunos directorios. Para recolocar el directorio etc de GNOME, usa la opción `–sysconfdir` . Para recolocar el directorio var, usa la opción `–localstatedir` .

Por ejemplo, una instalacion de GNOME con FHS podría tener el prefix `/opt/gnome` . Sin embargo, sus archivos etc deberían estar en `/etc/opt/gnome` , e igualmente, sus archivos var deberían estar en `/var/opt/gnome` . Esto tiene la ventaja de permitirte ***allowing you to network el directorio `/opt/gnome` , mantenerlo como sólo lectura (read-only), y permitir a unas máquinas tener configuración local e información sin preocuparse de otras máquinas. Para hacer esto, el comando de configuración sería el siguiente: **`./configure –prefix=/opt/gnome –sysconfdir=/etc/opt/gnome –localstatedir=/var/opt/gnome`** .

3.7. Ya tengo este módulo, ****how do I get rid of all the object files and other cruft from the source tree?

Tienes tres opciones, **make clean** , **make distclean** o borrar el directorio. Usar **make clean** borrará todos los ******object files, built executables, y miscellaneous cruft, es sencillo y seguro. La siguiente opción, **make distclean** debería borrar todo lo que no estubiera en el tarball original. Si haces esto, no podrás desinstalar el módulo . Y lo mismo para borrar el source directory. Basicamente, **make clean** es bueno para un uso general, las otros deberían ser consideradas medidas drásticas.

3.8. ¿Cómo desinstalo un módulo?

Si estás usando un sistema de manejo de paquetes (como RPM), simplemente sigue su documentación para desinstalar.

Si has instalado un módulo desde el código fuente, y piensas que quizás quieras desinstalarlo, *asegúrate que mantienes tu source tree* . Simplemente ve al fuente del módulo y ***go to the module's source and ejecuta **make uninstall** . Debería funcionar sin problemas y borrar todos los archivos que instaló.

Chapter 4. Y ahora ya tengo GNOME, ¿qué hago con él?

Ahora estamos hablando de dar los primeros pasos con GNOME, y las cosas que puedes hacer con él.

4.1. ¿Dónde miro primero?

Hay un tour ligeramente anticuado pero todavía útil en http://www.gnome.org/start/gnometour.shtml (http://www.gnome.org/start/gnometour.shtml) . Realmente vale la pena echarle un vistazo.

También, asegúrate que visitas la Guía de Usuario GNOME en http://www.gnome.org/users-guide (http://www.gnome.org/users-guide) ***************************** Y en GNOME-es ***************************** .

4.2. ¿Cómo inicio GNOME?

La forma correcta de iniciar GNOME con todas su características es poner la línea **`exec gnome-session`** al final de tu X startup script. Este script puede llamarse `~/.xinitrc` , `.xsession` o `.Xclients` , dependiendo de tu plataforma y configuración. Por supuesto, asegúrate de que los binario de GNOME están en tu path, y no hay ninguna línea que empieze por **`exec`** antes de la que acabas de añadir. Con las gnome-core versions 1.04 y posteriores, puedes seleccionar el gestor de ventanas a ejecutar poniendo la variable WINDOW_MANAGER en la que tu elijas.

Ejecutar **gnome-session** tal cual iniciará la sesión por defecto, la primera vez. Esto incluye el gestor de ventanas de tu elección (***or a good guess si no lo especificaste), el panel, GNOME Midnight Commander, y el visor de ayudas. A medida que uses GNOME, el manejador de sesiones guardará tu escritorio actual. Si sales (log out), y vuelves a entrar (log back in) más tarde, tu escritorio más reciente quedará como cuando saliste. *****your most recent desktop will be restored.

Si no quieres manejo de sesiones, GNOME funcionará bien sin él. Pon la línea **exec gnome-wm** al final de tu X startup script, como anteriormente. Utilizará la variable de entorno WINDOW_MANAGER para seleccionar tu gestor de ventanas, pero no te molestará con la gestión de sesiones. Ejecuta el panel, gmc, o lo que tú quieras, bien manualmente, o en el script de inicio antes de la linea de exec. Por ejemplo:

```
panel &
gmc &
exec gnome-wm
```

4.3. ¿Cómo ejecuto los programas?

Primero de todo, los programas de GNOME pueden ser ejecutados como cualquier otro programa. Vas a la linea de comandos, escribes su nombre, y el programa se inicia.

Lo segundo, puedes ejecutar programas desde el menu del panel. La mayoría de programas GNOME (y algunos no GNOME también) se instalan en el Foot menu en el panel de GNOME. Algunos programas a destacar en este menú; el Gestor de Archivos ejecuta la versión para GNOME del Midnight Commander, Settings->Editor de Menues te permitirá añadir o quitar cosas del Foot (y otros menus), Settings->Centro de Control GNOME mostrará el programa para configurar la mayoría de GNOME (y algunas cosas como Enlightenment), Juegos->Gnobots es un juego muy adictivo.

4.4. ¿Cómo ejecuto el Panel de GNOME?

El panel de GNOME arranca normalmente desde **gnome-session** , si no ocurre esto, simplemente teclea **panel** en cualquier linea de comandos para iniciar el panel.

Añade cosas al panel mediante el **Foot's Panel menu, o mediante el menu que aparece haciendo clic con el botón derecho en el panel. La Guía de Usuario GNOME (http://www.gnome.org/users-guide) tiene un Capítulo (http://www.gnome.org/users-guide/paneluse.html) entero sobre como usar el Panel.

Chapter 5. ¡¡¡¡¡¡¡Ayuda!!!!!!!

Incluso aunque hayas seguido todas las instrucciones al piede la letra, algunas veces surgen problemas imprevisibles. Cualquier cosa puede ir mal, desde un malentendido en las instrucciones, *** ¿debido a la traduccion ... :) ? comentario personal de javi *** a un bug en el software, o un bug en el software de otra persona. Aquí, intentamos hacer frente a problemas que han tenido con GNOME algunas personas.

5.1. Ayuda General.

Aquí estan las cuestiones más generales que surgen a menudo. Los problemas específicos se tratarán más adelante, en este capítulo.

5.1.1. Algo no funciona, estoy confundido, ¿qué hago?

Bueno, lo primero que debes hacer es una pausa, relájate, y respira profundamente. Seguidamente, revisa estas FAQ, la Guía de Usuario, y cualquier otra documentación que tengas a mano y mira a ver si alguna se aproxima a la resolución de tu problema. Lo siguiente es mirar en los archivos de la lista de correo (las instrucciones se dieron anteriormente) para ver si alguien más ha comentado algo sobre tu problema. Después prueba a ver, sea lo que sea lo que estabas haciendo *una vez más* , si la solución de repente aparece por si misma. Si nada de eso funciona, es hora de preguntar en la lista.

Antes de nada, estás suscrito a la gnome-list, ¿verdad? Si no estás, las instrucciones sobre como suscribirte se dieron anteriormente . Ahora, cuando pidas ayuda en la GNOME-list, *da muchos detalles* . Escribir***Posting " El panel no funciona, ¡esto es una mierda! ***The panel doesn't work, this sucks! " no ayuda ni a tí ni a nosotros. Aun tendrás el mismo problema, y nosotros tendremos un posible bug del cual no tendremos información. Es más útil (pero todavía insuficiente) decir " Cada vez que selecciono el foo del Foot menu, el panel entero desaparece de repente; ****Whenever I select the foo item from the Foot menu, the entire panel disappears suddenly; por cierto, ¿qué es este fichero que GNOME no deja de poner en mi directorio personal, y que se llama core ?. ”

Lo ideal, es que tengamos cierta información necesaria sobre tí. Lo primero, exactamente ¿que clase de sistema usas? No, no todo el mundo tiene el mismo sistema operativo que tú. Necesitamos el tipo de procesador: Sparc, MIPS, i386 (esto incluye Pentiums, AMD, Cyrix, etc), PPC, Alpha, ARM, 680x0, et al; ***¿que es el "et al" anterior?*** necesitamos el sistema operativo: distribución Linux, FreeBSD, HURD, OpenBSD, Solaris, IRIX, CrayOS, OS/400, o lo que sea. También incluye la versión del sistema operativo. Por ejemplo, la máquina en la que estoy ahora mismo es una GNU/Linux RedHat/i386 Rawhide machine, un amigo mio usa una máquina Solaris Sparc 2.6.

Seguidamente, necesitamos algunas vitales y relevantes estadísticas de que hay en tu sistema. *****we need some relevant vital statistics of what is on your system. En un sistema GNU/Linux, el número de versión del kernel y de libc son casi siempre importantes. En la mayoría de problemas de GNOME necesitamos las versiones de glib, gtk+, Imlib, ORBit y gnome-libs. Si no estás seguro si necesitas algunas de las anteriores, inclúyelas de todas formas. Otras cosas que a veces son importantes son las versiones de Perl, X11, Python, Guile y libpng, a menos que veas un mensaje de error que mencione alguna, puedes no incluirlas. Otra cosa importante es cómo instalaste GNOME. ¿Usaste binarios? si es así ¿cuáles? ¿Usaste tarballs? ¿Usaste el CVS?, si es así que fechas y/o tags? Si usaste una mezcla de los anteriores, menciónalo también.

Por último, necesitamos el problema en sí. Necesitamos saber exactamente qué hiciste para que surgiera el problema. Necesitamos saber si el problema es permanente o intermitente. Necesitamos saber si el programa mostró algo con altelación al problema. Necesitamos saber exactamente que sucede cuando ocurre el problema. Si quieres ser ambicioso, prueba con algunas herramienta de diagnostico de las que hablo más adelante, como backtrace, e incluye esa información en el mail.

Ahora antes de te prepares para mandar el correo, necesitamos encontrar un buen subject. La Gnome-list suele puede tener muchos mensajes, y tú quieres que el tuyo resalte que necesitas ayuda, y que pase desapercibido entre la gente que no pueda ayudar. **** messages on it, you want yours to stand out to the people you need help from, and be safely passed over by people who can’t help. Un subject como "GNOME está roto" es candidato a ser ignorado; uno como "GSM es una mierda (sucks)" pondrá a la gente de malas, y no con ganas de ayudar; algo como "La mayoría de los iconos del panel en HP-UX son sólo cuadros negros" es más probable que tenga la clase de atención que necesita.

Pon todo esto junto en un correo electrónico largo y envíalo a la lista. Si seguiste mi consejo, es muy probable que obtengas una rápida, *** se refiere a "atenta" o a "meditada"*** thoughtful response. Si por algún motivo no obtienes una rápida respuesta, no te lo tomes como algo personal, probablemente el mail se perdión en el montón. ***the mail probably just got lost in the volume. Espera un par de días, mira a ver si puedes añadir algún detalle, o una quizás una cabecera más descriptiva, y envíalo otra vez.

5.1.2. ¿Cómo informo de un bug?

Si encuentras un bug, queremos saber sobre él, queremos seguirlo, y queremos arreglarlo. Para hacer este proceso lo más eficiente posible, hemos tomado prestado de Debian su excelente sistema de seguimiento de bugs. Instrucciones completas sobre como utilizarlo en http://bugs.gnome.org/Reporting.html (http://bugs.gnome.org/Reporting.html) .

Hay unas pocas cosas que hay que tener en cuenta al enviar bugs al sistema. Primero asegúrate de comprobar la página ****to check the web front end at http://bugs.gnome.org (http://bugs.gnome.org) para asegurarte de que tu bug no se conoce todavía en el sistema. Si es un nuevo bug, envíalo a submit@bugs.gnome.org (submit@bugs.gnome.org) . Asegúrate de que incluyes las cabeceras ***headings*** del Paquete y Version como se describe en las instrucciones, esto asegura que será dirigido rápida y automaticamente a la persona adecuada. Asegúrate que incluyes todos los detalles. Mira la cuestión previa para tener una idea de lo que es útil incluir. Si un backtrace o strace es apropiado, desde luego que lo has de incluir. Las instrucciones sobre como hacer esto se dan más adelante.

Después de que envies el bug al sitema. Obtendrás comentarios sobre el bug via email. También puedes seguir la evolución del bug mediante el sistema de seguimiento de bugs en formato web. ****You also can track the status of the bug through the bug tracking system's web front end.

5.1.3. **** Esta pregunta deberia ser contentada antes de la seccion de compilacion ya que alli aparece y yo no sabia lo que era*** ¿Qué es un prefix (o un $prefix, ${PREFIX}, o <prefix>)?

Cuando se compila GNOME, necesitas especificar donde se situará en el sistema de archivos. La opción específica para usar esto se llama `–prefix` , por lo que la llamamos the GNOME Prefix, o prefix para abreviar. La localización de los archivos de GNOME debería ser la misma de un sistema a otro, excepto por el prefijo, por lo que un archivo en `/usr/share/gnome/pixmaps` en un sistema podría estar en `/opt/gnome/share/gnome/pixmaps` en otro, y nos referiríamos a la localización como `$prefix/share/gnome/pixmaps` .

Si instalaste desde tarballs (o CVS), ****but never gave **configure** (or **autogen.sh**) a prefix, por defecto es `/usr/local` . Si instalaste los RPMS de RedHat, estos usan el prefix `/usr` . Si todavía no estás seguro, escribe **which panel** , y quita `/bin/panel` del final. Lo que queda es tu prefix.

5.1.4. ¿Qué significa `***Segmentation Fault` ?

La mayoría de sistemas Unix y similares tienen una característica llamada protección de memoria. Esto significa que un programa puede acceder a algunas partes de la memoria (a menudo llamadas segmentos) *****(often called segments), pero no a otras. Si el programa intenta acceder a una zona fuera de los segmentos que tiene permitidos, el sistema detecta el problema y muestra un error. Este error se denomina un Segmentation Fault (abreviado “ segv ” algunos sistemas), y generalmente, el programa se cerrará de repente, normalmente dejando tras de sí un core dump (ver bajo).

Todo esto se reduce a la existencia de un bug sea cual sea el programa que estabas ejecutando. ****What it all boils down to is there is a bug in whatever program you were running. El bug podría estar en el código fuente de GNOME, podría estar en las librerías compartidas que usa tu programa, o podría estas en el modo en que se compiló el programa. Si no estás seguro si es un bug en GNOME o en otra parte, pregúntalo en la lista. Si estás seguro que el bug recae en GNOME, por favor asegúrate de que está en el sistema de seguimiento de bugs.

5.1.5. ¿Qué es un core dump? ¿Qué es este fichero llamado core que no paro de encontrarme? ***What is this file named core that I keep running into?

En los primeros días de los ordenadores electrónicos, la memoria de los ordenadores era un manojo de aros magnéticos llamados cores, montados sobre cables. Este tipo de memoria se llamaba memoria core. Cuando un programador quería ver lo que estaba pasando en su programa, extraía una copia de lo que había en la memoria core para así analizarla. Esta copia se llamaba un core dump, y el término ha permanecido hasta mucho después de que las memorias core se quedaran obsoletas. *****In the early days of electronic computers, computer memory was a bunch of magnetic rings called cores, mounted on wires. This type of memory was called core memory. When a programmer wanted to see what was going on in their program, they would output a copy of what was in their core memory so they could analyze it. This output was called a core dump, and the term has remained long after core memory became obsolete.

En muchos sistemas Unix y similares, el sistema producirá automaticamente un core dump cuando un programa falle catastroficamente, por ejemplo cuando se produce un Segmentation Fault (ver arriba). Generalmente pone el core dump en un fichero llamado `core` en el directorio donde estuvieras cuando ejecutaste el programa. Este fichero es muy útil para hacer un debugging y saber por qué fallo el programa, y puede ser usadoo para producir un backtrace (ver abajo). ***This file is very useful for debugging why the program failed, and can be used to produce a backtrace (see below). Por otra parte, este fichero puede ser bastante grande, particularmente si no lo usas para el debugging. ***this file can be quite large, particularly if you aren’t using it for debugging. Su no puedes permitirte tener core dumps, consulta la documentación de tu sistema para averiguar como desabilitar esta característica.

5.1.6. ¿Qué es un backtrace? Alguien me pidió que le enviara uno, ¿cómo lo hago?

Una herramienta de debugging incleiblemente útil es el backtrace. En esencia es un ***snapshot of exactly where in the program you are. Se realiza más a menudo cuando un programa ***crashes (cierra inesperadamente, a menudo con un segmentation fault u otro error) o ***hangs (deja de funcionar pero no se cierra). En esta respuesta, voy a tener que asumir que estás usando el debugger GDB del proyecto GNU, ya que ese es el más común incluido con GNOME, y el que yo he usado. Si usas otro debugger, consulta tu documentación.

Hay dos maneras de hacer un backtrace. Puedes usar un archivo core que dejó un programa cuando crashed, o puedes ejecutar el programa desde el debugger. La experiencia me demuestra que ejecutar el programa desde el debugger es más seguro, pero también comentaré como usar los archivos core. ***I have found that running the program within the debugger is more reliable, but I will discuss using core files as well.

Para ejecutar un programa dentro de GDB, ve a un xterm (o equivalente), y teclea **`gdb <nombre completo y localización del programa>`** , por ejemplo, para hacer un debug sobre gtalk, deberías hacer ***to debug gtalk, you might do **`gdb /usr/local/bin/gtalk`** . GDB no busca el path, por lo que tienes que especificarle donde está el progama. Si quieres ejecutar el programa con argumentos, *no los pongas aquí* . Puedes especificar argumentos de linea de comando después. ***You can specify command like arguments later.

GDB se identificará, y mostrará un prompt `(gdb)` . Desde aquí, escribe **`run`** para arrancar el programa. Si quieres ejecutar el programa con argumentos, ponlos aquí. Por ejemplo, para hacer un debug sobre gtalk con el sonido desactivado, deberías escribir ****to debug gtalk with sound disabled, you would type **`run -disable-sound`** . Entonces el programa arrancará. Ten presente que el programa irá más lento de lo normal, y usará mucha memoria.

Ahora intenta recrear el problema que tuviste. Haz lo que fuera que hicieres para causar el problema anterior. ***Do whatever it is that caused the problem before. Una vez que el programa ***crashes, deberías ver inmediatamente el material en la sesión de debug, y un prompt ***Once the program crashes, you should immediately see stuff in the debug session, and a `(gdb)` . Si el programa se cuelga, espera hasta estar seguro de que se colgó en el lugar correcto, y entonces aprieta Control-C en la sesión de debug. Tambíen aquí debería aparecer material de debugging, seguida de un prompt ***It too should reply with a bunch of stuff, followed by a `(gdb)` . En este prompt (sea cual sea la forma en que llegaste), escribe **`bt`** . Entonces respoderá con el backtrace propiamente dicho. Then it will reply with the actual backtrace.

Aquí tienes un ejemplo de una sesión de debug. Si te piden un backtrace, deberías enviarlo todo:

```
$

gdb /usr/local/bin/gtalk


GNU gdb 4.17
Copyright 1998 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public Li-
cense, and you are
welcome to change it and/or distribute copies of it under cer-
tain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show war-
ranty" for details.
This GDB was configured as "i386-redhat-linux"...


(gdb)


run -disable-sound


Starting program: /usr/local/bin/gtalk -disable-sound
Program received signal SIGINT, Interrupt.
0x4049290d in __poll (fds=0x808c110, nfds=3, timeout=-1)
at ../sysdeps/unix/sysv/linux/poll.c:45
```

```
../sysdeps/unix/sysv/linux/poll.c:45: No such file or directory.
```

```
(gdb)
```

bt

```
#0  0x4049290d in __poll (fds=0x808c110, nfds=3, timeout=-1)
at ../sysdeps/unix/sysv/linux/poll.c:45
#1  0x403c54e9 in g_main_poll (timeout=-1, use_priority=0, priority=0)
at gmain.c:991
#2  0x403c516e in g_main_iterate (block=1, dispatch=1) at gmain.c:789
#3  0x403c58a1 in g_main_run (loop=0x80882e8) at gmain.c:912
#4  0x401cbf1d in gtk_main () at gtkmain.c:475
#5  0x804a629 in main (argc=2, argv=0xbffffaa4) at main.c:54
#6  0x40407c77 in __libc_start_main (main=0x804a570 <main>, argc=2,
argv=0xbffffaa4, init=0x8049e10 <_init>, fini=0x804c7dc <_fini>,
rtld_fini=0x40009c10 <_dl_fini>, stack_end=0xbffffa9c)
at ../sysdeps/generic/libc-start.c:78
```

```
(gdb)
```

Hacer un backtrace con un core dump es muy parecido. Ve al directorio donde está el core dump, y entonces ejecuta **gdb -c core <nombre completo y localización del programa>** . De nuevo, GDB no mira en tu path, tienes que especificarle donde encontrar el programa. Todo lo que tienes que hacer es escribir **bt** en el prompt, y enviar la sesión entera.

5.1.7. ¿Qué es una strace? Alguien me pidió que le enviara una, ¿cómo lo hago?

El comando **strace** realiza una "Signal Trace". ******Using it will allow you to run a program while outputting system calls and signals. This is a much more verbose debugging tool than the backtrace, but sometimes you need the extra information an strace can give you. Obtener un strace es fácil, simplemente escribe **`strace -o <archivo de salida> <comando con argumentos>`** en un xterm. Por ejemplo, para hacer un strace al programa gtalk con el sonido deshabilitado, yo haría ***to do an strace on the gtalk program with sound disabled, I might do **`strace -o /tmp/gtalk.strace gtalk -disable-sound`** . Esto guardaría la salida del strace en `/tmp/gtalk.strace` . Strace sí que usa el path, y acepta argumentos de programa en la linea de comandos.

Advertencia, strace produce una gran salida (con el anterior ejemplo de strace se obtuvieron 5.498 lineas en cinco segundos). No envies straces a las listas de correo, y solamente envíaselas a los desarrolladores si te las piden especificamente *****Warning, strace produces lots of output (the above strace example came up with 5,498 lines in five seconds). Don't send straces to the mailing lists, and only send them to the developers if they specifically request it.

En sistemas Solaris, el comando **strace** hace algo diferente. El comando **truss** es probablemente lo que espere la persona que te pidió el strace.

5.2. Problemas Específicos de Compilación

5.2.1. ¿Qué significa `undefined reference to 'compress'` ?

En general, significa que se estan teniendo problemas para encontrar o linkar la librería de compresión (libz). ***In general, it means that it is having trouble finding or linking to the compression library (libz). Primero, asegúrate de que está instalada. Si no es así, entonces encuentrala e instálala. Debería ir incluida en todas las distribuciones más importantes de sistemas Unix y similares.

Si está instalada correctamente, y todavía aparece el error, entonces comprueba si existe alguna otra copia de libz danzando por ahí. Un lugar donde se esconde a menudo cuando la gente tiene problemas es en `/usr/X11R6/lib` , ya que XFree86 pone una copia allí. Es seguro simplemente borrarla, pero si no estás seguro, renómbrala como algo parecido a `libz.so.renamed` , de esa forma el linker no la encontrará, y si tienes problemas en otra parte por haberla renombrado, siempre puedes volverla a renombrar.

5.2.2. ¿Qué es `AC_TRY_RUN` , y por qué está mi máquina advirtiendome de ello?

Algunas veces, cuando ejecutamos aclocal, autogen o make, aparecerán lineas parecidas a: `configure.in:171: warning: AC_TRY_RUN called without default to allow cross compiling` . No es algo para estar preocupado, y puedes ignorarlo sin riesgos (a menos que estés cross compiling). ****This is nothing to be alarmed about, you can safely ignore it (unless you are cross compiling).

Si todavía sientes curiosidad, un cross compiler es un compilador basado en una plataforma que puedes usar para crear código para ser ejecutado en una plataforma diferente. Por ejemplo, alguna gente tiene una versión de GCC instalada en una box GNU/Linux con procesadores Intel que crean binarios para boxes SGI con procesadores MIPS. En este caso estaríamos hablando de un cross compiler. ****If you're still curious, a cross compiler is a compiler based on one platform that you can use to create code to run on a different platform. For example, some people have a version of GCC installed on GNU/Linux boxes with Intel processors which makes binaries for SGI boxes with MIPS processors. This would be a cross compiler.

La macro AC_TRY_RUN se llama para intentar compilar y ejecutar un fragmento de código. Se usa para comprobar si un determinado trozo de código tiene la posibilidad de funcionar en otra máquina o no, y ajustar la configuración consecuentemente. Sin embargo, si estás cross compiling, no puedes ejecutar el fragmento de código, ya que es de una máquina diferente a la que estás. La macro AC_TRY_RUN incluye un parámetro para que diga que hacer en ente caso, pero no hemos probado a cross-compiling GNOME todavía, por lo que sería prematuro configurar este valor.

Debido a la ausencia de este valor, aparece un warning. *****The AC_TRY_RUN macro is called to attempt to compile and run a code fragment. It is used to test to see whether a particular piece of code is likely to work on your machine or not, and adjust the configuration accordingly. If you are cross-compiling, however, you can't run the code fragment, since it's on a different machine than the one you are on. The AC_TRY_RUN macro includes a parameter to tell it what to do in this case, but we haven't tested cross-compiling GNOME yet, so it would be premature to configure this value. Because the value is absent, it gives the warning.

5.3. Problemas con el Runtime

5.3.1. ¿Cómo reseteo las propiedades del manejo de sesiones?

Mucha gente encuentra el manejo de sesiones muy útil. Sin embargo, algunas veces el manejo de sesiones puede despistarse, y hacer cosas indeseadas como recordar tus propiedades después de que tu hermano pequeño atacara tu ordenador. ****However, sometimes session management can go astray, and do undesired things like remember your settings after your kid brother attacked your computer. Si GNOME está correctamente instalado, volver tus propiedades a las por defecto es fácil.

Primero, sal completamente de X. Si estás usando el manejo de sesiones, deberías ser capaz de usar la opción `Log out` del ***Foot menu para salir. Si eso no funciona, aprieta las teclas control y alt, y presiona retroceder (borrar). Si eso tampoco a funcionado, y ahora aparece un ***login prompt tipo XDM o GDM **** seria conveniente explicar que es ****, cambia a un login modo texto (control-alt-F1 en muchos sistemas, consulta la documentación si no estás seguro, loguea como un usuario diferente y usa ****If that still doesn't work, and you are now looking at an XDM or GDM login prompt, switch to a text login (control-alt-F1 on many systems, consult your system documentation if you aren't sure, log in as a different user and use **su** en un xterm si estás realmente en un apuro ***in an xterm if you are really in a bind).

Después, para quitar tus propiedades, estando fuera de X, quita el archivo `~/.gnome/session` . La próxima vez que arranques las X, o la próxima vez que loguees desde GDM, deberías empezar con el gestor de ventanas, panel por defecto, el gmc y visor de ayuda.

5.3.2. Inicié GNOME con manejo de sesiones, ¿por qué me aparece esta pantalla gris (ni panel, ni nada)?

Si usas el manejo de sesiones, y no se puede encontrar una sesión, esto es lo que aparece. Primero, ¿has probado a resetear las propiedades del manejo de sesiones (Ver arriba)? Si esto no lo arregla, entonces **gnome-session** tiene un problema para encontrar el archivo `default.session` . En las versiones gnome-core 1.03 y posteriores el archivo debería estar en `$prefix/share/gnome` . ****Prior to that it should be in `$prefix/share` . Si el fichero no está, probablemente deberías instalar el gnome-core.

5.3.3. ¿Cómo le indico al manejo de sesiones que gestor de ventanas usar?

Esta es una pregunta más difícil de lo que parece, ya que la respuesta depende de que versión de GNOME estés utilizando. La respuesta más fácil es que uses las propiedades del apartado Window Manager en **control-center** . Si por alguna razón no te funciona, aquí tienes como hacerlo.

En la versión 1.03 del gnome-core y anteriores, el gestor de ventanas se selecciona poniéndolo en un archivo llamado `default.wm` . El archivo que crea el sistema por defecto debería estar en el directorio ****The system default for this file should be in either the `$prefix/share/gnome` o en el (or **en el original) `$prefix/share` , cualquiera que sea el que contiene al directorio ****whichever one has the file `default.session` ****** esta frase anterior no me queda nada clara ****.

El usuario puede poner una versión específica de `default.wm` en el directorio `~/.gnome`. El archivo `default.wm` debería parecese a:

```
[Default]
WM=foo
```

Donde **`foo`** debería ser remplazada por el gestor de ventanas que quieras. Si quieres cambiar el gestor de ventanas por defecto del sistema, también tendrás que cambiar el archivo `default.session`.

En las versiones 1.04 y posteriores de gnome-core, es mucho más fácil. Pon la variable de entorno WINDOW_MANAGER con el gestor de ventanas que quieras, entonces arranca X. ****If that variable is not set, GNOME ejecutará un script para tratar de encontrar con acierto un buen gestor de ventanas ****to make a good guess as to your window manager. Puedes tambier editar el archivo `~/.gnome/default.wm` como en las versiones antiguas.

5.3.4. Todo parece funcionar, pero la mayoría de mis iconos son simplemente cuadrados negros, o manchas**or random spots*. ¿Cómo hago aparecer mis iconos?

La mayoría de iconos de GNOME son imágenes de formato gráfico PNG (http://www.cdrom.com/pub/png) , mostradas via libpng y la librería de graficos Imlib. *****displayed via libpng and the Imlib graphics library. El problema es, que esta parte de Imlib parece ser muy melindrosa *** finicky, segun el diccionario tocho que tengo *** The trouble is, this part of Imlib seems to be very finicky, y expresa su desagrado mostrando basura (los cuadrados negros son basura más limpia). ****and it shows its displeasure by displaying garbage (the black squares are just cleaner garbage).

Hay unas cuantas cosas que pueden ser la causa de que esto ocurra. El problema más común es una libpng mala, confundida o que falta. ****The most common problem is a bad, missing or confused libpng. Si tienes libpng versión 1.0.2, asume que es mala.

Sólo usa las versiones 1.0.1, 1.0.3 o posteriores. Que no esté es fácil, si no existe ningún archivo llamdo "libpng.so" o "libpng.so.2", y estás en un sistema con librerías dinámicas, entonces te falta, consigue una copia e instálala. Para ver si está confundida, mira a ver si tienes más de una copia de "libpng.so" o "libpng.so.2", si es así, borra las dos y reinstala la correcta.

Otra posibilidad es la que tengo yo en una de mis máquinas (Una box RedHat con un procesador AMD5x86 y egcs). Si uso alguna optimización cuando compilo glib o imlib, aparece el problema descrito (imlib usa el gmodule portion de glib para cargar el código PNG). Compilar esos dos paquetes sin ninguna optimización funciona bien **switches on**. Eso lo arregla. Por lo que yo puedo decir, esto es un bug en mi compilador, no en GNOME. ***Another possibility is one I have on one of my machines (A RedHat box with an AMD5x86 processor and egcs). If I use any optimization when I compile glib or imlib, I get the problem described (imlib uses the gmodule portion of glib to load the PNG code). Try compiling those two packages with no optimization switches on. That might fix it. As far as I can tell, this is a bug in my compiler, not in GNOME.

También he recibido un informe donde se comenta que el problema podría ocurrir por interferencias de versiones de ImageMagick. Si estás desorientado, quizá quieras intentar quitar o actualizarla. ***I have also received a report that the problem could occur from interference from some versions of ImageMagick. If you are at a loss, you might want to try removing or upgrading it.

Chapter 6. Componentes de la Arquitectura

Esta sección trata las diversas piezas de GNOME, y qué hacen. ***This section discusses the various pieces of GNOME, and what they do.

6.1. Librerías

6.1.1. ¿Por qué necesito tantas librerías para GNOME?

GNOME requiere las librerías que requiere por dos grandes razones. Queríamos que GNOME tuviera muchas características útiles, y no queríamos empezar desde cero. Usar librerías comunes para muchas de las características de GNOME le permite a GNOME funcionar mejor y más consistentemente con otras aplicaciones no GNOME. ***GNOME requires the libraries it does for two big reasons. We wanted GNOME to have lots of useful features, and we didn't want to write everything from scratch. By using common libraries for many of GNOME's features it allows GNOME to work better and more consistently with other non-GNOME applications.

6.2. CORBA, ORBit y Baboon

Network Objects es el nombre intermedio de GNOME. Aquí te mostramos como funcionan. Network Objects is GNOME's middle name. This is how they work.

6.2.1. ¿Qué es CORBA?

CORBA (http://www.omg.org/news/begin.htm) es el Common Object Request Broker Architecture (*-* en-castellano-aqui *-*). Del documento del Open Management Group (OMG, el grupo de estándares que controla CORBA ***the standards body which controls CORBA) ¿Qué es CORBA? (http://www.omg.org/about/wicorba.htm) viene con la siguiente descripción:

> limiteX The Common Object Request Broker Architecture (CORBA), is the Object Management Group's answer to the need for inter-operability among the rapidly proliferating number of hardware and software products available today. Simply stated, CORBA allows applications to communicate with one another no matter where they are located or who has designed them(...)
>
> The (ORB) is the middleware that establishes the client-server relationships between objects. Using an ORB, a client can transparently invoke a method on a server object, which can be on the same machine or across a network. The ORB intercepts the call and is responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results. The client does not have to be aware of where the object is located, its programming language, its operating system, or any other system aspects that are not part of an object's interface. In so doing, the ORB provides inter-operability between applications on different machines in heterogeneous distributed environments and seamlessly interconnects multiple object systems.
>
> In fielding typical client/server applications, developers use their own design or a recognized standard to define the protocol to be used between the devices. Protocol definition depends on the implementation language, network transport and a dozen other factors. ORBs simplify this process. With an ORB, the protocol is defined through the application interfaces via a single implementation language-independent specification, the IDL (Ed. Interface Description Language). And ORBs provide flexibility. They let programmers choose the most appropriate operating system, execution environment and even programming language to use for each component of a system under construction. More importantly, they allow the integration of existing components. In an ORB-based solution, developers simply model the legacy component using the same IDL they use for creating new objects, then write "wrapper" code that translates between the standardized bus and the legacy interfaces.

OK, enough of that stuff. Here's Todd's explanation:

> Do you remember RPC? You know, Remote Procedure Calls? Sun used them as the basis

for NFS and NIS. Microsoft used the Distributed Computing Environment's RPC standard as the basis for DCOM.

Well, a remote procedure call is really pretty simple. First, you define the procedure in some standard format. What is there to a procedure definition? Well, you've got the name of the procedure, the arguments which go into it, and the results which come out of it. Then you produce the client side, which tells clients how to feed in arguments and get back results, and then you build the server side, which takes in arguments and gives back results.

You can use this model to build pretty powerful systems. Just take the standard Unix file system interface, convert it to RPC, and bammo, you've got NFS. Note that as long as the inputs and outputs are thrown around in a standard way, the clients calling the procedure and the server servicing the procedure don't have to be written in the same language, running on the same machine, or even running on the same operating system or hardware. This is, truth be told, the killer feature of RPC.

However, as procedural programming (C, Perl) has been superseded by object-oriented programming (Objective C, Java, C++), you really need something more the procedures in RPC. You need something which supports objects: creating objects, accessing object data, accessing object methods, destroying objects. This is where CORBA comes in: think of CORBA as next-generation RPC, just extended to support object-oriented programming. Instead of having under RPC the following:

```
void foo(int bar); void baz(){return(-1);}
```

you instead have under CORBA the following:

```
interface bubba{ void foo(in int bar); void baz() raises (InValid-
Context); }
```

The lesson of all of this? CORBA lets one program language-independent, location-transparent object-oriented interfaces between software components. It' s cool.

6.2.2. What are some CORBA components?

I would like to include here definitions of some of the more strange CORBA terminology, so that people can easily get into the groove of what CORBA is and have a good "beck-of-the-envelope" understanding of the components involved. Whenever I refer to a non-obvious CORBA term, I intend to define it here.

- *Object Request Broker* - (or ORB) an ORB is a piece of "middleware", as it's called, which sits between clients and servers and makes easy communication between them possible. The ORB is a conceptual entity, which sometimes takes the form of a shared library and sometimes takes the form of an external program. the ORB is responsible for establishing and destroying sessions between clients and servers and marshaling, unmarshaling, and transporting messages between them during a session.
- *Object Adaptor* - (or OA) an Object Adaptor provides the channel through which an object server (like the gnome panel) communicates with the Object Request Broker (ORBit).
- *GIOP/IIOP* - CORBA takes all of these object-oriented language-isms and handles carrying them between object-oriented software components (marshaling, transporting, and demarshaling). If the two pieces of software are in different locations, then it will use the GIOP protocol to move information between them. IIOP is a specification of GIOP for the Internet protocol suite; theoretically you could run GIOP over something other than IP, in which case it would be called something else. I for one can't figure out why you would want to. 8^) If any version of GIOP other than IIOP ever became important, then I'm sure that ORBit would support it, too. (Think of IIOP as HTML to GIOP's SGML.)

6.2.3. What are CORBA Services?

Additionally, there are really two parts to CORBA. There is the part which I've already described, which tells you how to do objectified RPC, how to marshal arguments, how to write IDL files, how GIOP works, etc. and then there are the "CORBA Services". These are services built on top of the basic CORBA infrastructure which make distributed object programming easier.

Perhaps the most important of these is the naming service. E.g., say that you were a program which for some reason needed spell-checking services. One really neat thing that you can do (if your CORBA implementation is nifty enough) is to call up your ORB and say "Hey! I need spell-check service. Go find it for me." The ORB then uses the naming service to see what objects are registered which provide spell-check service. It can then wake up the calling program and say "You can find the spell- check service at #867-5309." You then call up 867-5309 and, lo and behold, there's a spell check service there. Now, that might be a shared library which got mapped into your address space and you are now doing straight function calls into it, or it might be a community spell-check service in Mongolia offered as a public service by the Mongolian FreeBSD users' group over the internet via IIOP. You don't know and, except for maybe a little lag induced by the round-trip to Mongolia, you really don't care. Cool, non?

The naming service, then, allows objects to register themselves for future use. There are other CORBA services, including a transaction service, a security service, a time service, a "relationship service", which might or might not be what you think it is, and more. My copy of the CORBAservices document from March 1997 lists 14 of them.

You can find a copy of the CORBAservices specification at the OMG web page at http://www.omg.org/corba/csindx.htm (http://www.omg.org/corba/csindx.htm) . If you are interested in implementing any of them, then I'm sure that the ORBit developers would love to talk to you. (See ORBit)

6.2.4. Which CORBA version does GNOME use?

Work is presently underway within the OMG to develop version 3.0, but for now GNOME is using CORBA version 2.2, which is the present standard. I have not heard what GNOME will do when 3.0 comes out.

6.2.5. Where can I find out more about CORBA?

A good jumping-off page is Linas Vepstas's CORBA page at http://linas.org/linux/corba.html (http://linas.org/linux/corba.html) .

Since there seems to be a lot of interest in the difference between CORBA and COM, I'll just mention this one other paper:

http://www.research.att.com/~ymwang/papers/HTML/DCOMnCORBA/S.html (http://www.research.att.com/~ymwang/papers/HTML/DCOMnCORBA/S.html) , DCOM and CORBA Side by Side, Step by Step, and Layer by Layer.

Finally, the TAO project at Washington University in St. Louis has their own ORB, TAO. That group is led by Dr. Douglas Schmidt. You can find a lot of good information on his CORBA page: http://siesta.cs.wustl.edu/~schmidt/corba.html (http://siesta.cs.wustl.edu/~schmidt/corba.html) . Be sure to look around at the ACE and TAO pages while you're there; they are very good.

6.2.6. What role does CORBA play in GNOME?

CORBA enables the component architecture of GNOME. It serves a role similar to that of COM/DCOM under win32.

6.2.7. What CORBA implementation is GNOME using?

In the beginning, the plan was to use ILU (ftp://parcftp.parc.xerox.com/pub/ilu/ilu.html) . ILU had a number of benefits, foremost among them that it supports several languages. While ILU was (and is) awfully neat, Xerox's attitude towards it was unclear, and the licensing terms proved to be fatal: ILU is not free software, and the GNOME team proved unable to get Xerox to modify the license. The GNOME project won't adopt technology which is not free software, and that was the end of that.

The project then settled upon MICO (http://diamant-atm.vsb.cs.uni-frankfurt.de/~mico/) . The main attractions of MICO are that it has an Object Adaptor, it is IIOP-compliant, and it is licensed under the terms of the GPL. However, it does not presently support languages other than C++, and it used just an unbelievably amount of memory.

Still dissatisfied, the GNOMErs began work on our own ORB, called ORBit.

6.2.8. ¿Qué es orbit?

ORBit está diseñado *-* concebido *-* para ser multilingual; ILU es una prueba de que esto es posible. Ahora mismo sólo soporta C, pero en el futuro soportará otros lenguajes. (Realmente, ¡lo decimos en serio! *-* gracias diccionario !!! *-* Ahora mismo es un proyecto muy muy nuevo, que es la única razón por la que está en C solamente). También soporta GIOP/IIOP, el protocolo CORBA de la OMG que permite a diferentes ORBs ser capaces de hablar entre sí. ***ORBit is intended to be multilingual; ILU proves that this is possible. Right now it only supports C, but in the future it will support other languages. (Really, we mean it! It is a very, very new project right now, which is the only reason it is C only.) It also supports GIOP/IIOP, the OMG's CORBA protocol which allows different ORBs to be able to talk to each other.

Finalmente, ORBit está concebido para ser de alto rendimiento. Eso significa poca memoria y alta velocidad. Aquí, ORBit está siguiendo el ejemplo de ***Finally, ORBit is intended to be high performance. That means low-memory and high-speed. Here, ORBit is taking a lot of cues from Flick (http://www.cs.utah.edu/projects/flux/flick/) . Elliot Lee, el instigador del proyecto ORBit, piensa que puede, si violar el CORBA **spec, acercar el coste de una CORBA call ¿darn? al de una llamada a librería estandar, donde una implementación local de un servicio pedido está disponible. Ya veremos. ****the instigator of the ORBit project, thinks that he can, without violating the CORBA spec, get the cost of a CORBA call darn close to that of a standard library call, where a local implementation of the requested service is available. We'll see.

6.2.9. ¿Por qué no ha hecho GNOME mas uso de CORBA?

La razón más importante es que con MICO no se podía trabajar. Consumía demasiada memoria, era sólo C++, etc. ***ad naus.*** Con la reciente versión del GNOME ORBit, podemos empezar a desplegar CORBA en el GNOME. ***The big reason is that MICO was just unworkable. it took too much memory, it was C++ only, etc. ad naus. With the recent release of the GNOME ORBit, we can start deploying CORBA in the GNOME.

6.2.10. ¿Cómo se usará CORBA en GNOME? o ¿qué es BABOON? ***How will CORBA be used in GNOME, or, what is BABOON?

(Esta sección es casi literal de Miguel de Icaza, Jefe de los GNOMErs). ***This section comes pretty much verbatim from Miguel de Icaza, Chief of the GNOMErs.)

CORBA será usado en varios contextos. Un conjunto de interfaces y librerías de rutinas llamadas "Baboon" serán usadas para simplificar e integrar aplicaciones: ***CORBA will be used in various contexts. A set of interfaces and library routines called "Baboon" are used to simplify and integrate applications:

- Componentes: There are pre-packaged libraries/servers que proporcionan algunas funcionalidades que pueden ser usadas por cualquier aplicación como los ***spelling servers, servicios de calendario o otros servicios no gráficos **non-GUI services. Components: There are pre-packaged libraries/servers that do provide some functionality that can be used by any application like spelling servers, calendaring services and other non-GUI services.
- ***Application embedding and in-place activation: Baboon es un conjunto de interfaces para proporcionar **application embedding y in-place activation que están siendo definidos. Los interfaces Baboon e interacciones están modelados depués de los interfaces OLE2 y OpenDoc. ***Application embedding and in-place activation: Baboon is a set of interfaces for providing application embedding and in-place activation which are being defined. The Baboon interfaces and interactions are modeled after the OLE2 and OpenDoc interfaces.
- Aplicaciones GNOME como provedoras de servicio: las aplicaciones GNOME exportarán sus funcionalidades internas a través de CORBA para permitir a cualquiera reusar su funcionalidad. Esto permite el scripting del escritorio GNOME como en cualquier aplicación que use CORBA o lenguaje de script pueda invocar métodos en un servidor GNOME (por ejemplo, que un script en Perl pueda controlar remotamente para invocar una hoja de cálculo para hacer un gráfico o para hacer un informe de ventas). Alguna gente llama a lo anterior "Automation". A diferencia del mundo COM, el uso de CORBA hace esto de forma transparente a la

implementación de un servidor. ***GNOME Applications as service providers: GNOME applications will export their inner workings through CORBA to allow anyone to reuse their functionality. This allows scripting of the GNOME desktop as any CORBA-aware application or scripting language can invoke methods in a GNOME server (for example, a script in Perl can control remotely invoke a spreadsheet to make a graph or to make a sales report). Some people call this "Automation". Unlike the COM world, the use of CORBA makes this transparent to the implementor of a server.

- Controles Rehutilizables: Otro conjunto de interfaces de Baboon se ocupan de los controles rehutilizables. Esto es similar a los JavaBeans de Sun y las Active-X de Microsoft. ***Reusable controls: Another set of Baboon interfaces deal with reusable controls. This is similar to Sun's JavaBeans and Microsoft Active-X.

BABOON significa: Baboon Allows Baboon Objects Over Networks (Baboon permite objetos baboon sobre redes).

6.3. Gráficos en GNOME

6.3.1. ¿Qué es GTK+?

GTK+ es el Gimp Toolkit. Fue escrito originalmente como parte del proyecto Gimp, pero se ha convertido en un excelente ***widget set de propósito general. Más información en ****GTK+ is the Gimp Toolkit. It was originally written as part of the Gimp project, but it has grown into an excellent general purpose widget set. Find out more at http://www.gtk.org (http://www.gtk.org)

6.3.2. ¿Qué papel juega GTK+ en GNOME?

GTK+ es la interfaz que las aplicaciones de GNOME usan para interactuar con el usuario. Es similar al papel de Motif en CDE, Qt en KDE o el-widget-set-que-no-tiene-nombre en Win32. ***GTK+ is the interface which GNOME applications use to interact with the user. It is similar to the role of Motif in CDE, Qt in KDE or the-widget-set-which-has-no-name in Win32.

6.3.3. ¿Qué lenguajes soporta GTK+?

¡Puedes escribir programas GTK+ en toneladas de lenguajes! El humilde autor de estas FAQ conoce los siguientes ***bindings: ****You can write GTK+ programs in tons of languages! Your humble FAQ author knows of the following bindings:

- C (original)
- C++
- ***Objective C
- Python
- Perl 5
- Scheme (Guile)
- TOM
- Gwydion Dylan
- Ada95

Si hay otros, entonces por favor haznos saber cuáles son. (Ver If there are others, then please let us know about them. (See Contactar con el ***Maintainer de las FAQ ****Contact the FAQ Maintainer)

Te animamos a añadir ***to add bindings to other languages should the fancy strike you. ***N.b., que los bindings de Scheme ya tienen una representación formal en el API, y por lo tanto si quieres empezar a trabajar con nuevos bindings, quizá te gustaría empezar por aquí. ****You are encouraged to add bindings to other languages should the fancy strike you. N.b., that the Scheme bindings already have a formalized representation of the API, so if you want to start working on new bindings, then you might want to start from here.

Habiendo escrito lo anterior, me tengo que preguntar si no podríamos de alguna manera usar CORBA para cortar este nudo gordiano. Si *-* eso de nudo gordiano lo busque en el dicnario de la RAEL *-* los desarrolladores de ORBit can get the cost of a local

CORBA call down to that of a normal shared-library call, entonces ¿por qué no CORBificar GTK y trabajar para añadir ***bindings de nuevos lenguajes a ORBit? Sé, que la sabiduría popular dice que CORBA es demasiado pesado para cosas de mucho volumen como las llamadas a X. Oye, no me culpes por lanzar ideas locas: ¡la razón por la que mantengo las FAQ es que no puedo programar! ****Having written that, I have to wonder if we can't somehow use CORBA to cut this Gordian knot. If the ORBit developers can get the cost of a local CORBA call down to that of a normal shared-library call, then why not CORBify GTK and work on adding bindings for new languages to ORBit? I know, the conventional wisdom is that CORBA is too heavyweight for high-volume stuff like X calls. Hey, don't blame me for throwing out crazy ideas: the reason I maintain the FAQ is that I can't program!

6.3.4. ¿Qué es imlib?

Junto con GTK, GNOME usa imlib, una ***image library para X que soporta múltiples formatos de imagen de forma transparente para el programador. Imlib puede incluso usar programas convertidores externos como ImageMagick o pmbplus para convertir formatos de imagen desconocidos a conocidos, por lo que si existe un convertidor para un tipo de imagen, ¡puedes usar casi con toda seguridad ese tipo de imagen en tu aplicación! Imlib también soporta múltiples profundidades de bit, desde de 24 bits hasta 1 bit en monocromo, otra vez de forma transparente al programador, y realiza localización inteligente de mapa de colores y ***dithering automático para conseguir el mejor rendimiento de tu pantalla. Cualquiera que halla intentado alguna vez hacer esto con sus apps X, o quien halla tenido que soportar el destello de la pantalla cundo sus apps tienen que ***allocate a private color map, seguro que os gustará esta característica. *****Along with GTK, GNOME uses imlib, an image library for X which supports multiple image formats transparently to the programmer. Imlib can even use external converter programs like ImageMagick or pmbplus to convert unknown image formats to known formats, so if a converter exists for an image type, you can almost certainly use that image type in your application! Imlib also supports multiple bit-depths, from 24-bit down to 1-bit monochrome, again transparently to the programmer, and does intelligent color map allocation and automatic dithering to get the best possible performance out of your display. Anyone who has ever tried to do this in their X apps, or who has had to endure their screen flashing when apps have to allocate a private color map, you will especially like this feature.

Puedes encontrar más información sobre imlib en http://www.labs.redhat.com/imlib (http://www.labs.redhat.com/imlib) .

6.3.5. ¿Qué es eso de los themes? ***What's the deal with themes?

Los themes te permiten cambiar el aspecto y sensación de las aplicaciones GTK+ sin recompilarlas. Con los themes, eres capaz de cambiar el theme de todos los apps GTK+ en un ***display dado a la vez. Themes let you change the look and feel of GTK+ applications without recompiling them. With themes, you are able to change the theme of all GTK+ apps on a given display at the same time.

En realidad existen dos partes en un theme en GTK+, el motor y la configuración. El motor es una librería compartida especialmente construida con instrucciones sobre la manera en que los themes deben dibujar los widgets (los controles de una ventana, como los botones, barras de desplazamiento y menús). La configuración incluye información sobre fuentes, colores, que motor usar, y algunas veces imágenes gráficas que incorporar en el theme. Juntos, los dos constituyen el Theme, y le dan a GTK+ su flexible aparaciencia. ****There are actually two parts of a theme in GTK+, the engine and the configuration. The engine is a specially constructed shared library with instructions on the themes way to draw the widgets (the controls of a window, like the buttons, scrollbars and menus). The configuration includes information like fonts, colors, which engine to use, and sometimes graphic images to incorporate into the theme. Together, the two make the Theme, and give GTK+ its flexible appearance.

Para ver algunos ejemplos de cómo es un app GTK+ tematizado, échale un vistazo a: ***To see some examples of what themed GTK+ apps look like, take a look at: http://gtk.themes.org (http://gtk.themes.org) .

6.3.6. ¿Cómo hago para que funcionen los themes?

***How do I get themes working?

Primero, necesitas una versión reciente de GTK+, yo recomiendo actualmente la versión 1.2.1, pero la 1.2.0 funciona bien también. Después consigue los themes que quieras. Una buena selección de themes está disponible en el módulo de motores gtk. Incluso hay más disponibles en ***First, you need a recent version of GTK+, I recommend version 1.2.1 at the moment, but 1.2.0 works fine as well. You then get the themes you want. A good selection of themes is available in the gtk-engines module. Even more are available at http://gtk.themes.org (http://gtk.themes.org) .

Muchos de los módulos incluyen un motor. Necesitas compilar el motor, e instalarlo. Con los motores gtk, esto se realiza de igual manera que con cualquier otro módulo GNOME. Los themes de la website pueden tener distintas instrucciones, consulta su documentación. Algunos de los themes de la website son meras configuraciones alternativas de otro motor, muchos de ellos son para el motor pixmap en los motores gtk. ****Many of the themes include an engine. You need to compile the engine, and install it. In gtk-engines, this is done the same way as any GNOME module. The themes from the website might have different instructions, check their documentation. Some of the themes on the website are merely alternate configurations for another engine, many of them are for the pixmap engine in gtk-engines.

Una vez que el thema está instalado, puedes elegirlo a través del Centro de Control GNOME, bajo Escritorio-> Selector de Temas. El selector te permitirá elegir entre todos los temas instalados ***on the fly, previsualizar la apariencia del tema, y en la versión 1.04 o posterior incluso permitirte la elección de fuente. *****Once the theme is installed, you can select them through the GNOME Control Center, under Desktop->Theme Selector. The selector will allow you to choose between all the themes installed on the fly, preview what the the theme will look like, and in version 1.04 or later even let you select the font.

* *Puedes ver el estilo de los apps GTK temificados por el autor de las FAQ en: ***You can see what the FAQ author's themefied GTK apps look like at: http://www.mindspring.com/~tlewis/screen.jpg (http://www.mindspring.com/~tlewis/screen.jpg) . Comentado debido a un enlace muerto. ****Commented out due to dead link.*

6.3.7. ¿Qué es OpenGL?

OpenGL es un API de renderización de 2D y 3D desarrollado por Sillicon Graphics que se ha convertido en un estandar de facto de la industria. ***is a 2D and 3D rendering API developed by Silicon Graphics that has become a de facto industry standard. Basically, it is a tool to take a viewpoint of 3D space, and produce a 2D image of it suitable for drawing on a screen. To quote SGI's FAQ:

> OpenGL is a rendering only, vendor neutral API providing 2D and 3D graphics functions, including modeling, transformations, color, lighting, smooth shading, as well as advanced features like texture mapping, NURBS, fog, alpha blending and motion blur. OpenGL works in both immediate and retained (display list) graphics modes.

GNOME usa una implementación del API OpenGL llamado Mesa. Mesa no es OpenGL per se, ya que es un proyecto Free Software que no ha realizado el proceso de licencia de marca registrada OpenGL, pero es una fiel representación del API OpenGL. Mesa no es OpenGL(tm) de identica forma que GNU/Linux y FreeBSD no son Unix(tm)- uses an implementation of the OpenGL API called Mesa. Mesa is not OpenGL per se, since it is a Free Software project that has not gone through the OpenGL trademark licensing process, but it is a faithful representation of the OpenGL API. Mesa is not OpenGL(tm) in precisely the same way that GNU/Linux and FreeBSD are not Unix(tm).

6.3.8. ¿Y qué me dices del soporte Drag and Drop - Arrastrar y Soltar? What about Drag and Drop support?

GNOME usa actualmente el protocolo Xde de Drag and Drop. XDE está siendo remplazado con una implementación del Protocolo Drag-And-Drop de las X Windows para interoperar con otros toolkits. Más información en http://www.cco.caltech.edu/~jafl/xdnd/ (http://www.cco.caltech.edu/~jafl/xdnd/) .

6.4. Manejo de Sesiones

6.4.1. ¿Qué es el manejo de sesiones?

El manejo de sesiones es una extensión del sistema X Window que permite a las aplicaciones guardar y restablecer su estado. Esto permite al usuario que su escritorio quede como lo dejó, y no que se encuentren con un escritorio limpio cada vez que reentra en la GNOMosphere. ***Session Management is an extension to the X Window system which allows applications to save and restore their state. This allows users to have their desktop look like they left it, rather than having a clean desktop every time they re-enter the GNOMosphere.

6.4.2. ¿Cómo uso el manejo de sesiones en mi aplicación GNOME?

Dr. Mike de RedHat ha escrito un tutorial sobre este tema: http://www.gnome.org/devel/start/sm.shtml (http://www.gnome.org/devel/start/sm.shtml) .

6.5. DocBook

6.5.1. ¿Qué es DocBook?

DocBook es un DTD SGML (un conjunto de etiquetas -tags- para un archivo de texto) que es muy útil para escribir documentación. La mayoría de la documentación de GNOME está escrita en DocBook. Para más información, mira en las páginas de Oasis's DocBook en ***is a SGML DTD (a set of tags for a text file) that is very useful for writing documentation. Most of the GNOME documentation is written in DocBook.

To find out more, look at Oasis's DocBook pages at http://www.oasis-open.org/docbook (http://www.oasis-open.org/docbook) . También hay un buen tutorial para DocBook en ***There is also a good tutorial for DocBook at http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro.html (http://nis-www.lanl.gov/~rosalia/mydocs/docbook-intro.html) .

La versión de DocBook, y herramientas relacionadas, como las usadas por GNOME se pueden encontrar en ***The version of DocBook, and related tools, as used by GNOME can be found at ftp://ftp.cygnus.com/pub/home/rosalia/docware (ftp://ftp.cygnus.com/pub/home/rosalia/docware) . Tanto los tarballs como los RPMs binarios están disponibles allí. ***Tarballs and binary RPMs are both available there.

6.5.2. ¿Qué es SGML?

Por rellenar. ***To be filled in.

6.6. Internacionalización y Adapatación Local (I18N & L10N) ***Internationalization & Localization (I18N & L10N)

6.6.1. ¿Que son la Internacionalización(I18N) y la Adaptación Local(L10N)? ***What are Internationalization(I18N) and Localization(L10N)?

De la página info del GNU gettext: ***From the GNU gettext info page:

Por *internacionalización* , nos referimos al la operación por la cual un programa, o un conjunto de programas metidos en un paquete, es consciente y capaz de soportar multiples lenguajes. Este es un proceso de generalización, por la que los programas son liberados del uso único de cadenas inglesas u otros hábitos específicos del inglés, y conectados a formas genéricas de hacer lo mismo. Los desarrolladores de programas usan varias técnicas para internacionalizar sus programas, algunos de ellos han sido estandarizados. GNU gettext ofrece uno de estos estandars. ******refers to the operation by which a program, or a set of programs turned into a package, is made aware and able to support multiple languages. This is a generalization process, by which the programs are untied from using only English strings or other English specific habits, and connected to generic ways of doing the same, instead. Program developers may use various techniques to internationalize their programs, some of them have been standardized. GNU gettext offers one of these standards.

Por *adaptación local (localization)* , nos referimos a la operación por la cual, en un conjunto de programas ya internacionalizados, se le suministra al programa toda la información necesaria para que pueda valerse para manejar su objetivo de la manera adecuada para el lenguaje nativo y sus hábitos culturales. Este es un proceso de particularización, por el cual los métodos genéricos ya implementados en un programa internacionalizado se usan de manera específica. El entorno de programación pone varias funciones a disposición de los programadores que ayunda a esta **configuración runtime. La descripción formal de un conjunto específico de hábitos culturales para un país, junto con todas las traducciones asociadas orientadas al mismo lenguaje nativo, se conoce como el locale para este lenguaje o país. Los usuarios consiguen la adaptación local de los programas configurando los valores correctos para variables de entorno especiales, más que ejecutando esos programas, identificando que locale se devería usar. *** las primeras frases son algo dudosas *** ***one means the operation by which, in a set of programs already internationalized, one gives the program all needed information so that it can bend itself to handle its input and output in a fashion which is correct for some native language and cultural habits. This is a particularisation process, by which generic methods already implemented in an internationalized program are used in specific ways. The programming environment puts several functions to the programmers disposal which allow this runtime configuration. The formal description of specific set of cultural habits for some country, together with all associated translations targeted to the same native language, is called the locale for this language or country. Users achieve localization of programs by setting proper values to special environment variables, prior to executing those programs, identifying which locale should be used.

6.6.2. ¿Que API de I8N y L10N usa GNOME?

GNOME usa el estándar de internacionalización de Uniforum. Este es el "estándar" inventado por Sun Microsystems y adoptado por el proyecto GNU en su paquete "gettext". ***N.b., this is different from the competing X/Open standard.

6.6.3. ¿Cómo programo usando I18N L10N, o dónde puedo encontrar más información sobre estos temas? ***How do I program using I18N and L10N, or Where do I find out more info on these topics?

Échale un vistazo al manual online del paquete GNU gettext en ****Take a look at the GNU gettext package's online manual at http://gnu.org/manual/gettext/html_chapter/gettext_toc.html (http://gnu.org/manual/gettext/html_chapter/gettext_toc.html) .

De manera adicional, puedes encontrar informativo el FAQ sobre I18N: ***Additionally, you may find the I18N FAQ informative: http://www.vlsivie.tuwien.ac.at/mike/i18n.html (http://www.vlsivie.tuwien.ac.at/mike/i18n.html) .

6.7. Guile

6.7.1. ¿Qué es Guile?

Guile es la implementación realizada por el proyecto GNU de Scheme, un dialecto Common Lisp. Más info en ***Guile is the GNU project's implementation of Scheme, a Common Lisp dialect. Find out more at http://gnu.org/software/guile/guile.html (http://gnu.org/software/guile/guile.html) . Hay también una excelente información en http://www.red-bean.com/guile (http://www.red-bean.com/guile)

Los documentos de Guile son bastante buenos, pero el estándar Scheme (el maravillosamente, oscuramente llamado R5RS) se puede encontrar en: ***The Guile docs are pretty good, but the Scheme standard (the wonderfully, obscurely named R5RS) can be found at: http://www-swiss.ai.mit.edu/~jaffer/r5rs_toc.html (http://www-swiss.ai.mit.edu/~jaffer/r5rs_toc.html) .

6.7.2. ¿Qué papel juega Guile en GNOME?

Guile es el lenguaje de scripting o "pegamento" para el proyecto GNOME. Del anuncion original: ****Guile is the scripting or "glue" language for the GNOME project. From the original announcement:

> Planeamos usar las ***bindings de GTK/Scheme para escribir pequeñas utilidades y aplicaciones. Cuando estas ***bindings estén más maduras, debería ser posible escribir aplicaciones completas en Scheme. ***We plan to use GTK/Scheme bindings for coding small utilities and applications. When these bindings are more mature, it should be possible to write complete applications in Scheme.

Chapter 7. Asuntos de desarrolladores ***Developer Issues

7.1. Qué tiene que hacer un gestor de ventanas para ser compatible-GNOME ***What does a window manager have to do to be GNOME-compliant

Hay tres grandes cosas:

- Debe ser completamente compatible con ICCCM, especialmente con un apropiado soporte para el Manejo de Sesiones X. ***It must be fully ICCCM compliant, especially with proper X Session Management support.
- Debe soportar ***hints MWM ***It must support MWM hints.
- Debe soportar hints GNOME ***It must support GNOME hints.

Para obtener una información más detallada, visita la página Compatibilidad de Gestores de Ventana para GNOME en ****For more detailed information, see the GNOME Window Manager Compliance page at http://www.gnome.org/devel/gnomewm/book1.html (http://www.gnome.org/devel/gnomewm/book1.html)

7.2. ¿Qué es Autoconf? ¿Qué tiene que ver con GNOME?

Autoconf es una herramienta para producir ***shell scripts que automáticamente configuran el código fuente para adaptarse al sistema que estés usando. Autoconf es ****invaluable, ya que hace el desarrollo de esos ***configure scripts drásticamente fácil. GNOME lo usa junto a la herramienta relacionada, Automake, que permite a Autoconf producir Makefiles personalizados para tu sistema. Autoconf usa m4 para operar. M4 es un poderoso (pero arcano) macro lenguaje usado por muchas utilidades GNU. La mayoría de las macros usadas para generar estos ficheros se encuentran o en `/usr/share/aclocal` o en el subdirectorio `macros` del ***source tree.

Si estás usando los tarballs, Autoconf y Automake ya has sido ejecutadas por ti, por lo que no las necesitas en tu sistema. Si usas el CVS, no llegarás muy lejos sin tener versiones recientes de estas herramientas. Puedes encontrar más información sobre Autoconf en http://gnu.org/software/autoconf/autoconf.html (http://gnu.org/software/autoconf/autoconf.html) , y Automake en http://gnu.org/software/automake/automake.html (http://gnu.org/software/automake/automake.html) .

Ten en cuenta, que `configure` , `config.h` , `Makefile` , e incluso `Makefile.in` son todos archivos construidos.***are all constructed files. Si necesitas cambiarlos, no deberías parchearlos directamente, si no mejor modificar los ficheros usados para crearlos: ***If you need to change them, you shouldn't patch them directly, but rather modify the files used to create them: `configure.in` , `config.h.in` , y `Makefile.am` .

7.3. ¿Qué es Libtool? ¿Qué tiene que ver con GNOME?

Libtool es una importante utilidad de portabilidad. Porporciona una interfaz común para el mecanismo de ***building, debugging y mantenimiento de librerías estáticas y dinamicas en la mayoría de sistemas en uso. Funciona en tres fases, el comando ***Libtool is an important portability utility. It gives a common interface to the mechanics of building, debugging and maintaining static and dynamic libraries on most computer systems in use. It works in three phases, the **libtoolize** añade soporte libtool al source tree añadiendo el script **ltconfig** . Este script, ***in turn, genera un script específico del tipo de sistema en el source tree llamado ****generates a system-specific script called **libtool** . El script **libtool** es la parte que en realidad realiza todo el trabajo. ***is the part that actually gets all the work done.

Como con Autoconf y Automake, si usas el CVS, necesitas tener Libtool instalada en tu sistema. Si usas los tarballs, **libtoolize** ya ha sido ejecutada, y **ltconfig** ***is self-contained, por lo que no necesitas tener Libtool instalada. Puedes encontrar más información sobre libtool en http://gnu.org/software/libtool/libtool.html (http://gnu.org/software/libtool/libtool.html)

Chapter 8. Convirtiéndose en un GNOMEr

Información general sobre como introducirse más en el proyecto GNOME. ***General information on getting more involved with the GNOME Project.

8.1. ¿Cómo me uno al movimiento GNOME?

Si quieres introducirte seriamente en GNOME, realmente tienes que unirte a la lista de correo gnome-list (Ver Listas de correo , anterior). Comprueba la lista de listas de correo para ver que otras listas te interesan. También, asegúrate de que lees el Manifesto GNOME en http://www.gnome.org/about/manifesto.shtml (http://www.gnome.org/about/manifesto.shtml) y los artículos sobre la Filosofía del Proyecto GNU en http://gnu.org/philosophy (http://gnu.org/philosophy) para asegurarte de que sabes en qué clase de proyecto te estás metiendo. No tienes que estar de acuerdo con toda la filosofía para contribuir con GNOME, pero al menos respetar que algunos de nosotros estemos entregados a las ideas mostradas en estos documentos.

Debido a que GNOME es un proyecto de software, es obvio que siempre podemos necesitar a más programadores, pero no tienes que ser un programador para hechar una mano en GNOME. Siempre podemos necesitar a más gente que envie claros informes de bugs. Siempre hay más documentación que escribir. Si hablas más de un idioma, siempre podemos necesitar traductores, tanto con la documentación como con las **labels, botones y entradas de menú en las aplicaciones. La gente dispuesta a hacer paquetes binarios para sus sistemas serán acogidos con gusto, particularmente si nadie más esta realizando todavía los binarios para ese sistema. Animamos con entusiasmo a la gente dispuesta y capaz de replicar nuestros servidores WWW, FTP y/o CVS a que lo hagan. **** realmente ¿como traduzco el verbo USE?****** ****Since GNOME is a software project, we obviously can always use more programmers, but you don’t have to be a programmer to help out with GNOME. We also can always use more people sending in clear bug reports. There is always more documentation to be written. If you speak more than one language, we can always use translators, both of the documentation and of the labels, buttons and menu entries within the applications. People willing to make binary packages for their system are quite welcome, particularly if nobody else is making binaries yet for that system. People willing and able to mirror our WWW, FTP and/or CVS servers are enthusiastically encouraged to do so.

8.2. ¿Cómo consigo las ultimísimas versiones CVS de GNOME? ***How do I get the bleeding edge, CVS versions of GNOME?

Si desarrollas para GNOME, o si quieres simplemente mantenerte más al día que con la última versión, deberías tener en cuenta usar nuestro sistema CVS. Hay un servidor principal CVS, para el que necesitas una cuenta, y una red de servidores réplicas de este, a los que cualquiera puede acceder. Esta red de mirrors se llama en conjunto CVS Anónimo (Anonymous CVS), y puedes acceder a ellos mediante anonscvs.gnome.org (***they are set up with round-robin DNS to balance out the load). Si pretendes colaborar con código fuente para el proyecto GNOME, al final querrás una cuenta en el servidor CVS principal, pero primero acostumbrate al sistema de CVS Anónimo, para familiarizarte con el uso que hace GNOME del CVS. *****If you are developing for GNOME, or if you just want to keep more up-to-date than the latest release, you should consider using our CVS system. There is the main CVS server, which you need an account for, and a network of mirrors of the server, which anyone can read from. This network of mirrors is collectively called Anonymous CVS, and they all can be accessed at anoncvs.gnome.org (they are set up with round-robin DNS to balance out the load). If you are looking to contribute source code to the GNOME project, you will eventually want an account on the main CVS server, but get used to the Anonymous CVS system first, so you are familiar with how GNOME makes use of CVS.

Las direcciones sobre como acceder al sistema CVS Anónimo están en ***Directions on how to access the Anonymous CVS system are at http://www.gnome.org/devel/whatiscvs.shtml (http://www.gnome.org/devel/whatiscvs.shtml) . Hay también una muy útil guia en http://www.cinternet.net/~soren/gnome/cvs.shtml (http://www.cinternet.net/~soren/gnome/cvs.shtml) . Los servidores CVS anómimos deberían actualizarse solos *****The anonymous CVS servers should be updating themselves

Para usar la versión CVS, necesitas tener GNU tools instalado en tu sistema. Necesistas por lo menos GNU Autoconf, Automake, Libtool y m4. Si tu sistema no las incluye, puedes conseguirlas del sitio web GNU en http://gnu.org (http://gnu.org) .

8.3. Quiero el ultimísimo GNOME, pero no puedo usar CVS. ¿Cómo lo consigo? ***I want bleeding edge GNOME, but I can't use CVS. How do I get it?

Hay algunas ocasiones en las que CVS no es factible. Quizá quieres asegurarte de que GNOME funciona usando simplemente la herramientas nativas de tu sistema en vez de las GNU tools. Quizá estés al otro lado de un cortafuegos que bloquea el puerto CVS. Sea cual sea tu razón, Jim Pick Software ofrece ***snapshots diarios de la versión CVS, en formato tarball, disponibles via FTP en There are some occasions where CVS just isn't feasible. Perhaps you want to make sure GNOME works using just the native tools for your system instead of the GNU tools. Perhaps you are sitting on the other side of a firewall that blocks the CVS port. Whatever your reason, Jim Pick Software offers daily snapshots of the CVS version, in tarball form, available via FTP at ftp://ftp.jimpick.com/pub/gnome/snap (ftp://ftp.jimpick.com/pub/gnome/snap) . La conversión a formato tarball evita la necesidad de las GNU tools también. ****The conversion to tarball form bypasses *** bonito verbo **** the need for GNU tools as well.

8.4. ¿Cómo consigo una cuenta que me permita contribuir con la versión CVS de GNOME? ****How do I get an account to let me contribute to the CVS version of GNOME?

Si planeas contribuir con código al proyecto, entonces necesitarás una cuenta en el servidor CVS. Miguel de Icaza (****If you plan on contributing code to the project, then you will need an account on the CVS server. Miguel de Icaza (miguel@nuclecu.unam.mx (mailto:miguel@nuclecu.unam.mx)) determina quien consigue una cuenta en el servidor CVS; si quieres una, entonces envíale un mensaje detallando que código planeas escribir, junto con un password encriptado. Puedes conseguirlo haciendo lo siguiente: determines who gets accounts on the CVS server; if you would like one, then send him a message detailing what code you plan on writing, along with with a crypted password. You can get that by doing the following:

```
perl -e 'print crypt ("YourPassword", "salt");'
```

Donde la cadena "salt" preferiblemente debería ser una cadena aleatoria de dos caracteres. *** dudosa traduccion *** ****Where the "salt" string preferably should be a random 2-char string.

8.5. ¿Por qué faltan directorios algunas veces cuando actualizo un módulo desde el CVS? ****Why are directories sometimes missing when I update a module from CVS?

El CVS posee una excelente característica que tiene un molesto capricho. El CVS permite a un módulo incluir otro módulo, una característica que GNOME usa para organizar mejor las cosas y reducir la duplicidad de código. Desafortunadamente, el comando ****CVS has an excellent feature with an annoying quirk about it. CVS allows a module to include other modules, a feature which GNOME uses to better organize things and to reduce duplication of code. Unfortunately, the **cvs update** no se preocupa de comprobar y ver si se han incluido algún nuevo módulo aparte del principal. Si crees que falta alguna pieza, usa ***command doesn't bother to check to see if any new modules are included off of the main module. If you think there is a missing piece, use **cvs get** en vez del anterior, ya que comprobará la existencia de nuevos módulos incluidos. No te preocupes, siempre y cuando el source tree esté ahí, no intentará hacer un nuevo y total chequeo, si no que actuará al igual que ***instead, it will check for new included modules. Don't worry, as long as the source tree is there, it won't try to do a brand new checkout, it will act the same as **cvs update** así pues descargando lo que sea nuevo o esté actualizado. ****as far as only downloading what is new or updated.

Chapter 9. Sobre las FAQ ***FAQ Issues

9.1. ¿Dónde está la copia oficial de estas FAQ?

La copia oficial y formal de estas FAQ se encuentra en foramto DocBook (SGML) en el servidor CVS en ***The formal and official copy of this FAQ is kept in DocBook (SGML) format on the CVS server at `cvs.gnome.org:/cvs/gnome` , el archivo en sí se encuentra en el módulo ***the file itself is located in the `gnome-libs` en el fichero ***module in the file `devel-docs/gnome-faq.sgml` . Todas las demás versiones son copias, portes, reformas, etc. ***All other versions are copies, ports, reformatings, etc.

9.2. ¿Cómo añado una pregunta y/o respuesta a estas FAQ? ****How do I add a question and/or an answer to this FAQ?

La manera más fácil es enviar un correo al ****maintainer de las FAQ, en The easiest way is to email the FAQ maintainer, at gnomefaq@gnu.org (mailto:gnomefaq@gnu.org) . Si la sugerencia es apropiada, editaremos tu información, y la incorporaremos en el lugar apropiado. ****If the suggestion is appropriate, we will edit your information, and incorporate it in at the appropriate place.

Si tienes acceso al CVS, puedes modificar las FAQ por ti mismo, pero preferiría que limitases las modificaciones directas a rápidos cambios de errores evidentes, más que a cambios importantes o incorporaciones. ****If you have CVS access, you can modify the FAQ on your own, but I would prefer if you limit direct modification to quick fixes of glaring errors, rather than major changes or additions.

Chapter 10. Créditos

10.1. Copyright y Disclaimer

Este artículo se proporciona como es sin ninguna garantía expresa o implícita. Aunque se ha realizado todo el esfuerzo posible para asegurar la exactitud de la información contenida en este artículo, el autor/mantenidor/contribuidores no asume(n)n responsabilidad alguna por los errores u omisiones, o por daños resultantes del uso de la información aquí contenida. ***This article is provided as is without any express or implied warranties. While every effort has been taken to ensure the accuracy of the information contained in this article, the author/maintainer/contributors assume(s) no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

10.2. ****Trademarks and Naming Credits

Aquí están algunas de las trademarks menos obvias usadas en este documento. ***Here are some of the less obvious trademarks used in this document.

*Linux es una trademark de Linus Torvalds. ****is a trademark of Linus Torvalds.*

Debian es una trademark de Software in the Public Interest

*Copyleft y Free Software no son trademarks de nadie, pero son importantes términos en la comunidad Free Software, y para la Free Software Foundation. ****are not trademarks of anyone, but they are important terms in the Free Software community, and for the Free Software Foundation.*

Open Source es una service mark de Software in the Public Interest o de la Open Source Initiative o Bruce Perens, nadie está totalmente seguro. is a service mark of either Software in the Public Interest or the Open Source Initiative or Bruce Perens, nobody's quite sure.
Unix , y The X Window System son trademarks del Open Group.
OpenGL es una trademark de Silicon Graphics, Inc.
Todas las demás trademarks son propiedad de sus respectivos dueños. ****All other trademarks are the property of their respective owners.

Si hay alguna trademark que necesita ser listada formalmente en las anteriores, háganolo saber por todos los medios para que pueda ser añadida. ****If there is a trademark here that needs to be listed formally above, by all means let us know so it can be added.