

D-Bus: aplicaciones que "hablan" entre sí

Traducción al castellano de la [página de introducción al proyecto D-Bus](#) en «*Freedesktop Organization*» cuya última edición de dicho documento fue efectuado el 17 de diciembre de 2018.

¿Qué es D-Bus?

D-Bus es un sistema de bus de mensajes, una forma sencilla para que las aplicaciones se comuniquen entre sí. Además de la comunicación entre procesos, D-Bus ayuda a coordinar el ciclo de vida del proceso; hace que sea sencillo y confiable codificar una aplicación o *daemon* de "instancia única", y lanzar aplicaciones y demonios cuando se necesiten sus servicios.

D-Bus suministra tanto un *daemon* del sistema (para eventos como "nuevo dispositivo de hardware agregado" o "cola de impresora cambiada") y un *daemon* de sesión de inicio de sesión por usuario (para las necesidades generales de IPC entre las aplicaciones de usuario). Además, el bus de mensajes se construye sobre un marco general de paso de mensajes uno a uno, que puede ser utilizado por cualquiera de las dos aplicaciones para comunicarse directamente (sin pasar por el *daemon* del bus de mensajes). Actualmente, las aplicaciones que se comunican están en una computadora, o mediante TCP / IP sin cifrar, adecuado para usar detrás de un muro de fuego con directorios de inicio de NFS compartidos (buscamos ayuda con mejores [transportes remotos](#): el mecanismo de transporte está bien resumido y es extensible).

La [implementación de referencia de la API de bajo nivel dbus](#) y el protocolo D-Bus se han probado en gran medida en el mundo real durante varios años, y ahora están "establecidos en piedra". Los cambios futuros serán compatibles o versionados apropiadamente.

La biblioteca de referencia de **libdbus** de bajo nivel no tiene dependencias requeridas; la única dependencia requerida del *daemon* de bus de referencia es un analizador XML ([expat](#)). Los enlaces de alto nivel específicos para marcos específicos (Qt, GLib, Java, C #, Python, etc.) agregan más dependencias, pero pueden hacer más suposiciones y, por lo tanto, son mucho más fáciles de usar. Los enlaces evolucionan por separado del **libdbus** de bajo nivel, por lo que algunos son más maduros y [ABI estables](#) que otros; revise los documentos para el enlace que planea usar.

También hay algunas reimplementaciones del protocolo D-Bus para lenguajes como C #, Java y Ruby. Estos no utilizan la implementación de referencia de **libdbus**.

Cabe señalar que la implementación de bajo nivel no está diseñada principalmente para que la utilicen los autores de aplicaciones. Más bien, es una base para los autores vinculantes y una

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

referencia para las reimplementaciones. Si puede hacerlo, se recomienda que utilice uno de los enlaces o implementaciones de nivel superior. Una lista de estos se puede encontrar en la [página de enlaces](#).

La [lista de proyectos que usan D-Bus](#) está creciendo y brindan una gran cantidad de ejemplos sobre el uso de varias API para aprender.

D-Bus es muy portátil a cualquier Linux o UNIX, y hay un puerto para Windows en progreso.

Si tiene algún problema con D-Bus o sugerencias para mejorar, los informes de errores y los comentarios son bienvenidos.

¡Súbete al D-Bus hoy!

Freedesktop logotipo

Lista de correo

Toda la "discusión" sobre D-Bus está en: dbus@lists.freedesktop.org

Reportando excepciones y sugiriendo soluciones

Informe los errores (y envíe las solicitudes de combinación) para la especificación D-Bus y el dbus de implementación de referencia a través del proyecto [dbus / dbus en la instancia de Gitlab de freedesktop.org](#).

Para otras implementaciones y enlaces, use el rastreador de errores para la implementación apropiada. Por ejemplo, los errores de GDBus se rastrean en el [proyecto GLib en Gitlab de GNOME](#), y los errores de sd-bus se rastrean en el [proyecto systemd en Github](#).

Lo ideal es incluir cobertura de suite de prueba con su parche; o si informa de un error, es bueno agregar una prueba que falle incluso si no tiene un "parche" para ello.

Los "parches" para mejorar la cobertura de la prueba son muy bienvenidos, aunque D-Bus ya se encuentra entre los códigos base mejor cubiertos que existen.

Documentación

Algunas cosas del subdirectorío doc / son pre-compiladas y navegables aquí. Si eres nuevo en D-Bus, el tutorial es probablemente el mejor lugar para comenzar (aunque está muy incompleto, lo básico está cubierto).

Información genérica del protocolo D-Bus:

- [D-Bus specification](#).
- [D-Bus Overview](#) de la documentación de **txdbus**.
- [Una introducción básica](#) por Jeroen Vermeulen.
- [Una introducción al D-Bus](#) desde la documentación de la librería **Qt**.
- [Preguntas y respuestas frecuentes \(FAQ\)](#).
- Panorama gráfico D-Bus [png](#) [svg](#).
- [Tutorial sobre D-Bus \(incompleto, incomplete, tiene detalles y reimplementaciones\)](#).
- [Archivo de configuración DTD](#).
- Si estás confundido acerca de algunos de los conceptos en D-Bus, mira [algunas analogías](#).
- Algunas herramientas para trabajar con D-Bus: tenga en cuenta que la especificación de D-Bus está incompleta, especialmente en su descripción del *daemon* del bus de mensajes. La especificación para el protocolo en sí es razonablemente completa, aunque no siempre

clara o precisa. ¡Tus soluciones son bienvenidas! Mientras tanto, es posible que deba complementar su lectura de la especificación con una lectura del código fuente de la implementación de referencia.

[SVG: Diagrama D-Bus]

Documentos específicos para la implementación de referencia, **dbus**:

- [API reference manual for the reference implementation \(libdbus\)](#).
- [@todo items from reference implementation manual](#) and high-level [TODO](#) (enlace histórico [TODO](#)).
- [Browse reference implementation source](#).
- [dbus-daemon\(1\)](#) (incluye ficheros de ejemplos de documentación).
- [dbus-send\(1\) dbus-monitor\(1\) dbus-launch\(1\) dbus-uuidgen\(1\)](#).
- [Consejos para sacar el mejor partido sobre D-Bus \("hacking"\)](#). [Lista de autores](#).
- [Noticias](#). [Registro de cambios](#). [Fichero LEEME](#).
- [Plan de pruebas](#).
- [Tarball](#) (fichero comprimido) con la mayoría de los documentos anteriores. Tenga en cuenta que **libdbus** es una biblioteca de bajo nivel, destinada a ser el servidor para un enlace de idioma y con la complejidad adicional necesaria para implementar el **dbus-daemon**. Se ahorrará mucho dolor si utiliza un envoltorio de nivel superior o una reimplementación. La documentación de estos suele estar vinculada desde la [página de enlaces](#).

Artículos en toda la web, incluyendo algunos tutoriales:

- ["Connect desktop apps using D-BUS" \(IBM developerWorks\)](#) by Ross Burton (July 2004).
- ["Get on D-BUS" \(Red Hat Magazine\)](#) by John Palmieri (January 2005).
- ["Get on the D-BUS" \(Linux Journal\)](#) by Robert Love (January 2005).
- ["The DBus missing tutorial - DBus Activation"](#) by Raphaël Slinckx (2005).
- [D-Bus Low-Level API Tutorial](#) by Matthew Johnson (Nov 2005).
- [Introduction To D-BUS](#) by Aaron Seigo & KDE community (2007).
- ["An Introduction to the D-Bus Language Binding for ooRexx"](#) by Rony G. Flatscher (December 2011, Intro to D-Bus concepts followed by intro to the ooRexx bindings).

Descargas

Implementación de referencias (dbus, incorporating dbus-daemon and libdbus)

dbus es la implementación de referencia de D-Bus. Las versiones publicadas pueden descargarse del [directorio de versiones en dbus.freedesktop.org](http://dbus.freedesktop.org) y están disponibles en todas las principales distribuciones de Linux. En caso de duda, utilice los paquetes de su distribución.

(Nota: traducción hecha el sábado 22 de junio de 2019)

La rama estable actual es [dbus 1.12.x](#). Esta es la versión recomendada para la mayoría de los propósitos.

La rama heredada actual es [dbus 1.10.x](#). Esto todavía es compatible, pero solo para soluciones de seguridad: solo use esta versión cuando actualice versiones anteriores estables o cuando prepare actualizaciones de seguridad para distribuciones congeladas / estables.

Las ramas más antiguas, como 1.8.x y 1.6.x, han llegado al final de su vida útil y es poco probable que tengan más lanzamientos, pero los distribuidores que aún brindan soporte de seguridad para una versión más antigua están invitados a compartir parches con portada posterior a través de las sucursales más antiguas en el mismo repositorio git. Por favor, póngase en contacto con los mantenedores para obtener más información.

La rama de desarrollo actual es [dbus 1.13.x](#), lo que llevará a una rama estable de 1.14.x en el futuro. La rama de desarrollo no viene con soporte de seguridad, y si requiere arreglos de seguridad a menudo se lanzarán como parte de una versión de desarrollo ordinaria que también contiene cambios potencialmente desestabilizadores. No incluya una rama de desarrollo de dbus en su distribución de sistema operativo a menos que pueda comprometerse a rastrear esa rama y la rama estable que la sigue, por ejemplo, continuar rastreando dbus 1.13.x y luego cambiar a 1.14.x cuando esté disponible.

Las ramas de desarrollo sustituidas, como 1.11.xy 1.9.x, no están totalmente soportadas y no reciben ninguna solución de seguridad.

Windows port

El puerto de Windows de los proyectos **windbus** y **dbus4win** se fusionó en la rama de desarrollo de **dbus freedesktop** hace varios años y se lanzó como parte de **dbus**. Gracias a todos los contribuyentes pasados y actuales a ese puerto.

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

El puerto de Windows se sabe que funciona en Windows XP, Windows Vista y Windows 7, los compiladores SDK compatibles son: MSVC 2010, mingw-w32 / w64 (gcc) y cygwin (gcc). " POR HACER: estos necesitan una actualización. "

Todos los interesados en tener **dbus** estable en Windows están invitados a probar la implementación, a informar errores y / o a presentar parches.

Encuadernaciones e implementaciones independientes

Los enlaces y las implementaciones independientes están vinculados a la [página de enlaces](#).

Un enlace envuelve a libdbus (y por lo tanto obtiene automáticamente, por ejemplo, nuevos mecanismos de autenticación y otras adiciones a libdbus), mientras que una reimplementación codifica el protocolo desde cero (y por lo tanto evita una dependencia en la biblioteca de C de libdbus, pero debe mantenerse sincronizado con las nuevas características).

Código fuente

Las últimas versiones de la Especificación D-Bus y el dbus de implementación de referencia están disponibles en el [proyecto dbus / dbus en la instancia de Gitlab de freedesktop.org](#).

Para otras implementaciones y enlaces, use el repositorio de código fuente para la implementación apropiada. Por ejemplo, GDBus es parte del [proyecto GLib en Gitlab de GNOME](#), y sd-bus es parte del [proyecto systemd en Github](#).

Enlaces: [DBusRemote](#) [DBusReleaseArchive](#) [IntroductionToDBus](#) [Software Specifications/config-spec](#) [Specifications/mpris-spec](#) [jhbuild](#).

Enlaces adicionales

En idioma inglés

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

- [dnsmasq](#) es un software que utiliza D-Bus, ver su [página de parámetros disponibles para ello](#).