Subject: DBus package for U++
Posted by Oblivion on Sun, 30 Aug 2026 14:13:40 GMT
View Forum Message <> Reply to Message

Hi,

I have made my DBus package for U++ public. Basically, it is a fresh implementation of DBus protocol from scratch.
It will be available on UppHub, shortly (awaiting merge).

Here's a summary from its README:

DBus is a native D-Bus Inter-Process Communication (IPC) library for the U++ framework. This repository contains the core DBus components required to build both clients and servers entirely using native U++ code.

Features

- Native Protocol Implementation: Speaks the D-Bus binary protocol directly using standard U++ Socket and String classes, avoiding external dependencies like libdbus or glib.
- Event Loop Integration: Integrates easily with U++'s SocketWaitEvent for asynchronous messaging. It sleeps when idle and wakes on network traffic, ensuring it won't burn CPU cycles or freeze your GUI.
- Full Client & Server Support: Query properties, call remote methods, claim well-known bus names, listen for incoming requests, and send structured replies seamlessly.
- Built-in Authentication: Automatically handles standard D-Bus EXTERNAL handshakes to connect cleanly to both local session buses and system buses.

Documentation & Tutorials

Design Docs: Pseudoblocking, Async Mechanics, and API Usage
Tutorials:
Method Calling
Complex Variant Marshaling
Global Multiplexing and Async Event Loops
Sending Desktop Notifications
Building a Server Service

Examples
The DBus nest ships with ready-to-run interactive examples demonstrating both client-side and server-side setups, from basic blocking calls to background daemons.

- Method A basic example demonstrating D-Bus method calls.
- Notify A minimal client application demonstrating how to trigger a native desktop notification using a simple blocking method call.
- Monitor An asynchronous listener that subscribes to system-wide D-Bus broadcast signals and logs them to the console in real-time.

- Marshall Demonstrates how to marshall complex, poymorphic D-Bus data easily.
- Properties Demonstrates how to query and parse complex nested structures using the standard org.freedesktop.DBus.Properties interface.
- Server A headless background daemon that claims a well-known bus name and routes remote method requests asynchronously.

Subject: Re: DBus package for U++
Posted by Oblivion on Thu, 03 Sep 2026 19:03:09 GMT
View Forum Message <> Reply to Message

Hi,

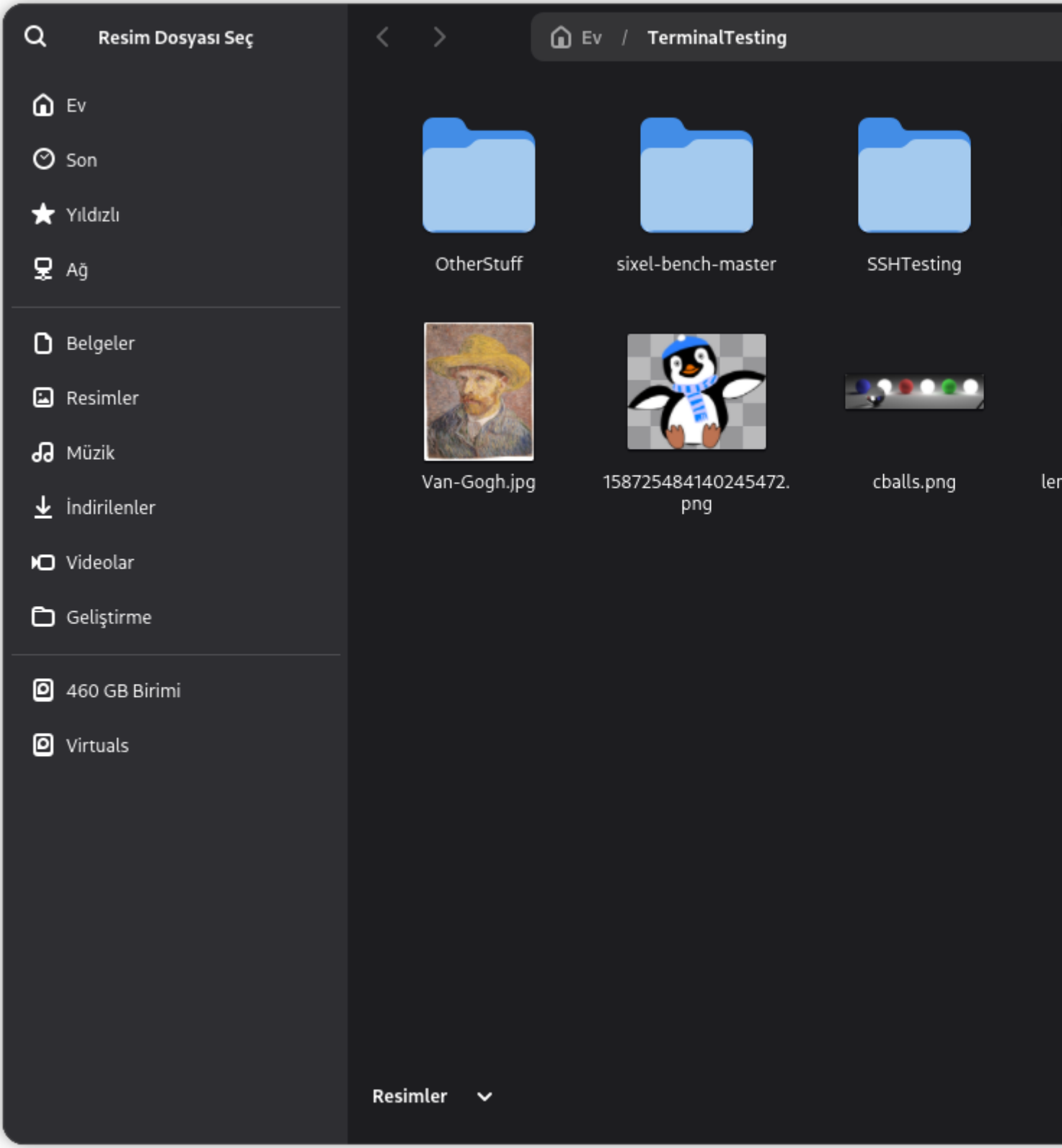
I have added a native file chooser example to DBus package. It demonstrates:

- Multi file selection
- File type filtering
- Localization

Best regards,
Oblivion

File Attachments

1) dbus-filechooser-gnome.png, downloaded 163 times


Resim Dosyası Seç
Ev
Son
Yıldızlı
Ağ
Belgeler
Resimler
Müzik
İndirilenler
Videolar
Geliştirme
460 GB Birimi
Virtuals
Ev / TerminalTesting
OtherStuff
sixel-bench-master
SSHTesting
Van-Gogh.jpg
1587254841402454 72.png
cballs.png
Resimler

Subject: Re: DBus package for U++
Posted by Oblivion on Fri, 11 Sep 2026 07:38:25 GMT
View Forum Message <> Reply to Message

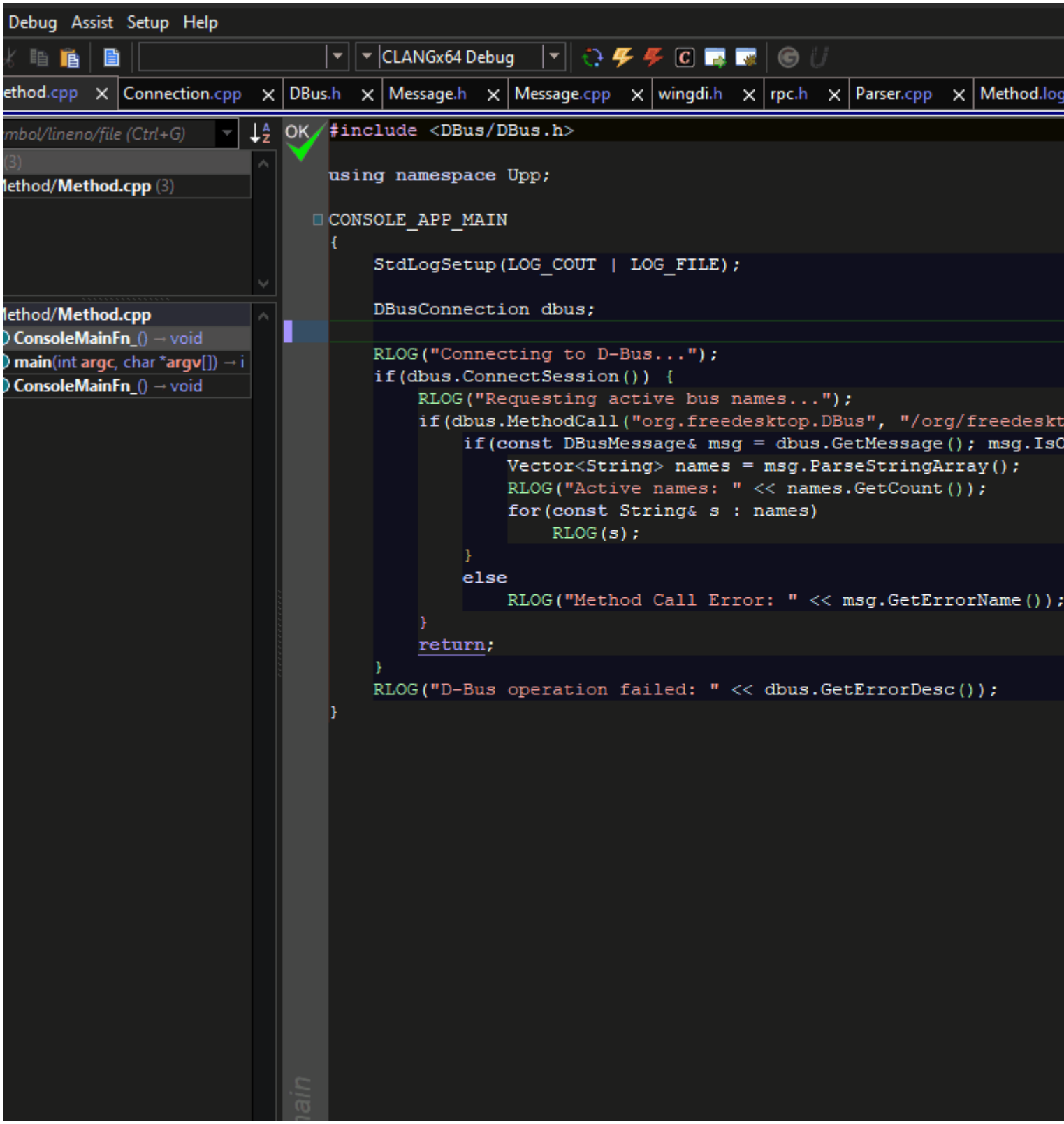
Hi,

With the latest merge, DBus package for U++ now offically has native Windows support, meaning you can compile and use DBus package in a cross-platform fashion, provided that you install official D-bus daemon for Windows. Now we can move to `DBusLib`, which will contain think wrappers of DBus package for various stuff (Inspection, mpris, notification, file selection, network query, etc.)

Best regards,
Oblivion

File Attachments

1) upp-dbus-windows.gif, downloaded 103 times



Subject: Re: DBus package for U++
Posted by Oblivion on Sat, 12 Sep 2026 16:28:26 GMT

Hi,

I finally added DBusLib to UppHub DBus nest. It contains (for now):

- `DBusNotification`
- `DBusInspector`
- `DBusMediaPlayer`

What's missing (for now):

- `DBusFileSel`
- `DBusKeyring`

All of these classes can work both in blocking and non-blocking mode and they are simply thin wrappers around the core DBus package, meaning they are very cheap and easy to use. API docs are also included. And examples section now contain introspection, MPRIS reference examples.

Cheers.

Subject: Re: DBus package for U++
Posted by Oblivion on Wed, 16 Sep 2026 16:40:58 GMT

Hi,

DBusFileSelector is added to DBusLib package. It is now possible to do this with native desktop file choosers:

```
#include <DBusLib/DBusLib.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
   StdLogSetup(LOG_FILE | LOG_COUT);

   DBusConnection dbus;
   if(!dbus.ConnectSession()) {
      RLOG("D-Bus connection failed. " << dbus.GetErrorDesc());
      Exit(1);
   }

   const char *filetypes = "Images\t*.png *.jpg *.jpeg\nText\t*.txt *.doc *.odt";
```

```
    RLOG("Selected file(s): " << SelectOpenFiles(dbus, filetypes));
    RLOG("Selected Folder:  " << SelectDirectory(dbus));
//    RLOG("Selected file content: " << SelectLoadFile(dbus, filetypes));
}
```

These one-liner functions all reflect Upp's own FileSel style. and allow opening, loading, saving files in a single line with native file selectors of available desktop (GNOME, KDE, etc.).

Best regards,
Oblivion

Subject: Re: DBus package for U++
Posted by Oblivion on Mon, 21 Sep 2026 05:43:03 GMT
View Forum Message <> Reply to Message

Hi,

The latest entry in the `DBusLib` is `DbusScreenshot`.
It allows taking screenshots even on wayland, both in interactive mode and non-interactive mode (the latter requires polkit).

The reference example:

```
#include <DBusLib/DBusLib.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
    StdLogSetup(LOG_FILE|LOG_COUT);

    DBusConnection dbus;
    if(!dbus.ConnectSession()) {
        RLOG("Failed to connect to D-Bus session bus: " << dbus.GetErrorDesc());
        Exit(1);
    }

    // Take a screenshot in "interactive mode"
    RLOG("Screenshot path: " << TakeScreenshot(dbus));

}
```

With this addition we have now

- DBusInspector

- DBusMediaPlayer
- DBusNetworkManager
- DBusNotification
- DBusFileSelector
- DBusScreenshot

in the companion library for DBus core. They all have API docs included.

The list will eventually grow.

Best regards,
Oblivion
