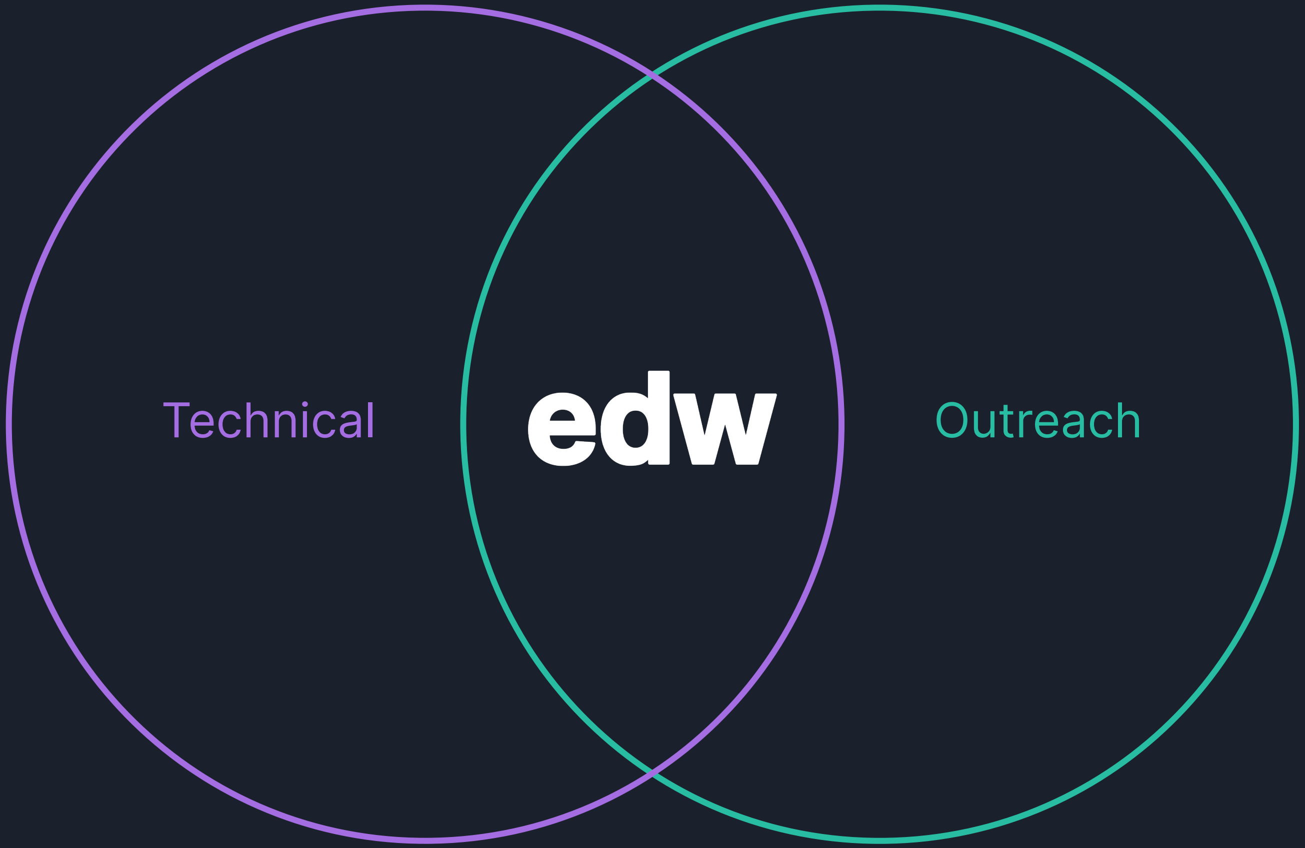
edw

2021



Technical

edw

Outreach

195

New open source apps

AppCenter

for everyone.

```
public class MyApp : Gtk.Application {
    public MyApp () {
        Object (
            application_id: "com.github.myteam.myapp",
            flags: ApplicationFlags.FLAGS_NONE
        );
    }

    protected override void activate () {
        var window = new Gtk.ApplicationWindow (this) {
            default_height = 768,
            default_width = 1024,
            title = "MyApp"
        };
        window.show_all ();
    }

    public static int main (string[] args) {
        return new MyApp ().run (args);
    }
}
```

```
Project('com.github.myteam.myapp', 'vala', 'c')

executable(
    meson.project_name(),
    'src' / 'Application.vala',
    dependencies: [
        dependency('gtk+-3.0')
    ],
    install: true
)

install_data(
    'data' / 'myapp.desktop',
    install_dir: get_option('datadir') / 'applications',
    rename: meson.project_name() + '.desktop'
)

install_data(
    'data' / 'myapp.appdata.xml',
    install_dir: get_option('datadir') / 'metainfo',
    rename: meson.project_name() + '.appdata.xml'
)
```

```yaml
app-id: com.github.myteam.myapp
command: com.github.myteam.myapp

runtime: io.elementary.Platform
runtime-version: 'daily'
sdk: io.elementary.Sdk

finish-args:
  - '--share=ipc'
  - '--socket=fallback-x11'
  - '--socket=wayland'

modules:
  - name: myapp
    buildsystem: meson
    sources:
      - type: dir
        path: .
```

```vala
public class MyApp : Gtk.Application {
    public MyApp () {
        Object (
            application_id: "com.github.myteam.myapp",
            flags: ApplicationFlags.FLAGS_NONE
        );
    }

    protected override void activate () {
        var window = new Gtk.ApplicationWindow (this) {
            default_height = 768,
            default_width = 1024,
            title = "MyApp"
        };
        window.show_all ();
    }

    public static int main (string[] args) {
        return new MyApp ().run (args);
    }
}
```

```meson
Project('com.github.myteam.myapp', 'vala', 'c')

executable(
    meson.project_name(),
    'src' / 'Application.vala',
    dependencies: [
        dependency('gtk+-3.0')
    ],
    install: true
)

install_data(
    'data' / 'myapp.desktop',
    install_dir: get_option('datadir') / 'applications',
    rename: meson.project_name() + '.desktop'
)

install_data(
    'data' / 'myapp.appdata.xml',
    install_dir: get_option('datadir') / 'metainfo',
    rename: meson.project_name() + '.appdata.xml'
)
```

```yaml
app-id: com.github.myteam.myapp
command: com.github.myteam.myapp

runtime: io.elementary.Platform
runtime-version: 'daily'
sdk: io.elementary.Sdk

finish-args:
  - '--share=ipc'
  - '--socket=fallback-x11'
  - '--socket=wayland'

modules:
  - name: myapp
    buildsystem: meson
    sources:
      - type: dir
        path: .
```

```
[Desktop Entry]
Name=MyApp
GenericName=Hello World App
Comment=Proves that we can use Vala and Gtk
Categories=Utility;Education;
Exec=com.github.myteam.myapp
Icon=com.github.myteam.myapp
Terminal=false
Type=Application
Keywords=Hello World;Example;
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<component type="desktop">
  <id>com.github.myteam.myapp</id>
  <metadata_license>CC0</metadata_license>
  <name>Your App's Name</name>
  <summary>A Catchy Tagline</summary>
  <description>
    <p>A quick summary of your app's main selling points and features.
Just a couple sentences per paragraph is best.</p>
  </description>
</component>
```

freedesktop.org

CSS Styles

Check & Radio Buttons

Popovers

Action States

FlowBox

Spin Buttons

Lists

Scales

Keyboard Navigation

Localization

Stack

Grid

Animations

Resizable Panes

Text Entries

Progress Bars

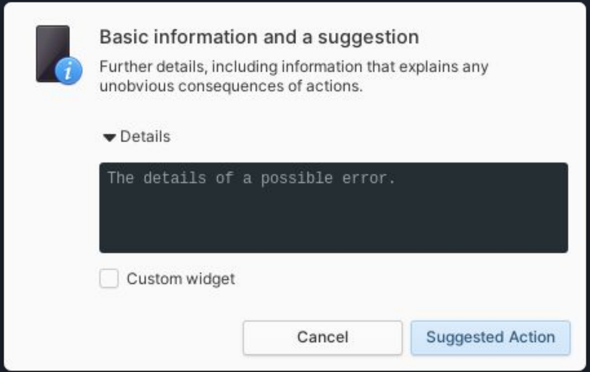
Menus

Switches

Inspector

Buttons

Granite + Handy



Basic information and a suggestion

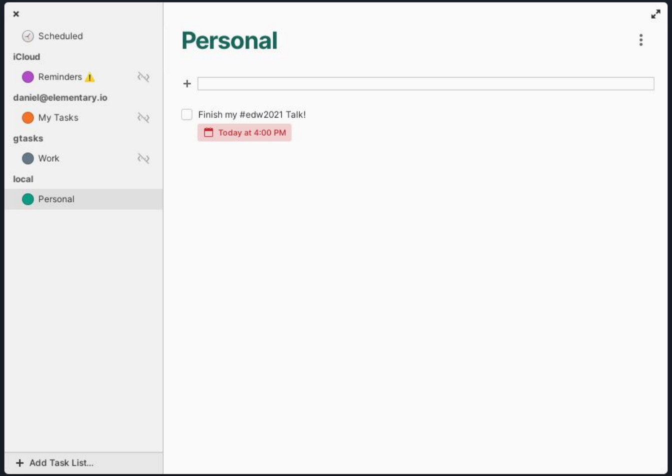
Further details, including information that explains any unobvious consequences of actions.

The details of a possible error.

☐ Custom widget

```
var dialog = new Granite.MessageDialog (
    "Basic information and a suggestion",
    "Further details, including information that explains any unobvious consequences of actions.",
    new ThemedIcon ("phone"),
    Gtk.ButtonsType.CANCEL
) {
    badge_icon = new ThemedIcon ("dialog-information"),
    transient_for = (Gtk.Window) get_toplevel ()
};

dialog.show_error_details ("The details of a possible error.");

dialog.custom_bin.add (new Gtk.CheckButton.with_label ("Custom widget"));

var suggested_button = dialog.add_button ("Suggested Action", Gtk.ResponseType.ACCEPT);
suggested_button.get_style_context ().add_class (Gtk.STYLE_CLASS_SUGGESTED_ACTION);

dialog.show_all ();

dialog.response.connect ((response_id) => {
   if (response_id == Gtk.ResponseType.ACCEPT) {
       toast.send_notification ();
   }

   dialog.destroy ();
});
```

```
deck = new Hdy.Deck () {
    can_swipe_back = true,
    can_swipe_forward = true
};
deck.add (category_view);
deck.add (setting_view);
```

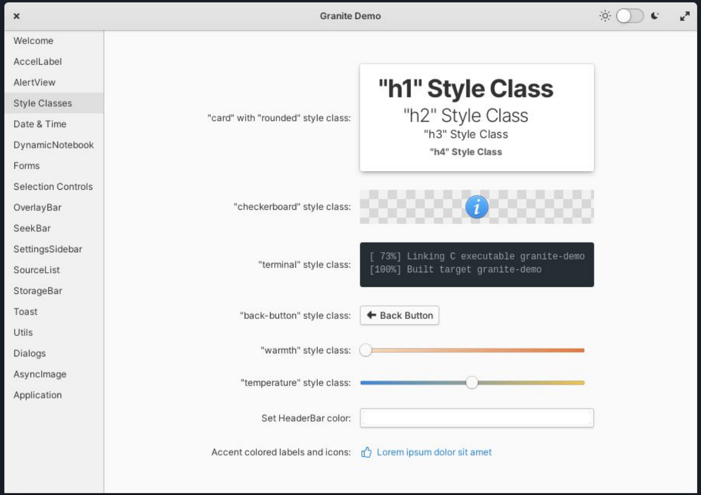


Personal

□ Finish my #edw2021 Talk!

Today at 4:00 PM

```
public class Tasks.MainWindow : Hdy.ApplicationWindow {
    construct {
        Hdy.init ();

        var sidebar_header = new Hdy.HeaderBar () {
            decoration_layout = "close:",
            show_close_button = true
        };
        sidebar_header.get_style_context ().add_class (Gtk.STYLE_CLASS_FLAT);

        var list_header = new Hdy.HeaderBar () {
            decoration_layout = ":maximize",
            show_close_button = true
        };
        list_header.get_style_context ().add_class (Gtk.STYLE_CLASS_FLAT);

        var paned = new Gtk.Paned (Gtk.Orientation.HORIZONTAL);
        paned.pack1 (sidebar_header, false, false);
        paned.pack2 (listview_header, true, false);

        add (paned);
    }
}
```



"card" with "rounded" style class:

"h1" Style Class

"h2" Style Class

"h3" Style Class

"h4" Style Class

"checkerboard" style class:

"terminal" style class:

```
[ 73%] Linking C executable granite-demo
[100%] Built target granite-demo
```

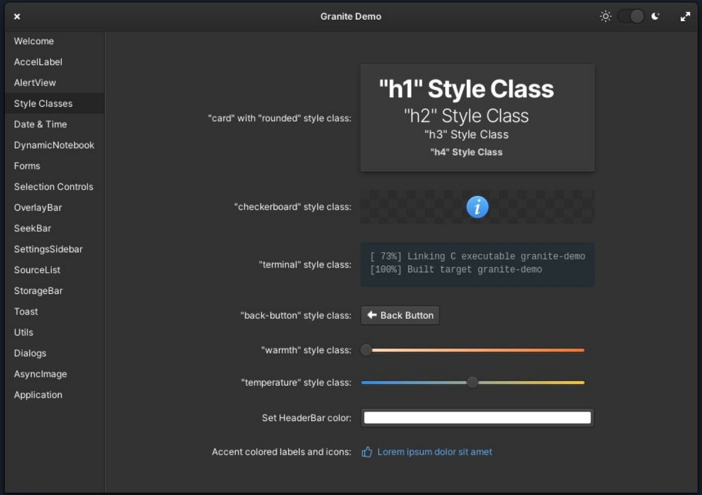
"back-button" style class:

"warmth" style class:

"temperature" style class:

Set HeaderBar color:

Accent colored labels and icons: Lorem ipsum dolor sit amet


Granite Demo

"card" with "rounded" style class:

"h1" Style Class

"h2" Style Class

"h3" Style Class

"h4" Style Class

"checkerboard" style class:

"terminal" style class:

[73%] Linking C executable granite-demo
[100%] Built target granite-demo

"back-button" style class:

"warmth" style class:

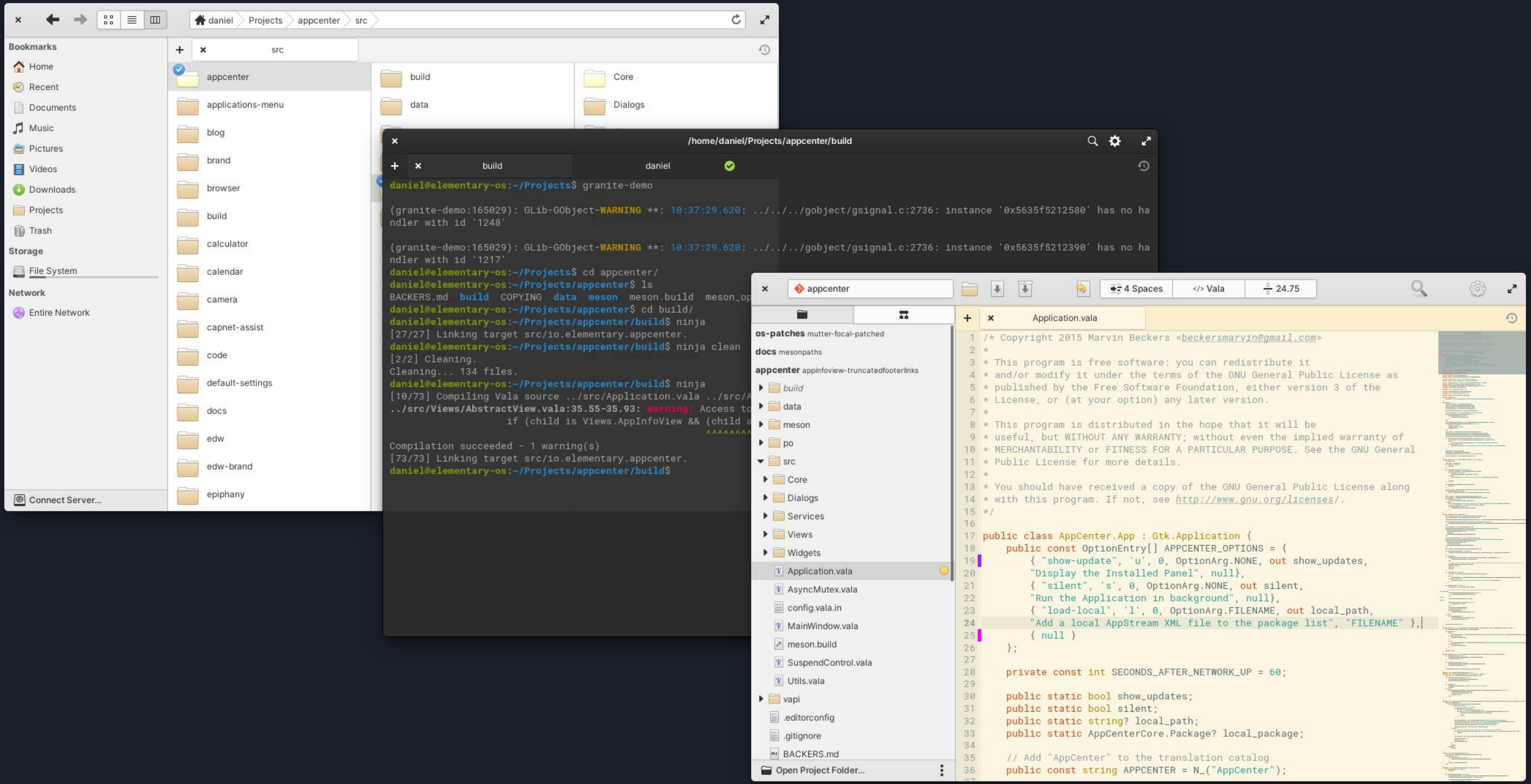
"temperature" style class:

Set HeaderBar color:

Accent colored labels and icons: Lorem ipsum dolor sit amet

>400

Open source SVG icons



```
daniel@elementary-os:~/Projects$ granite-demo

(granite-demo:165029): GLib-GObject-WARNING **: 10:37:29.620: ../../../gobject/gsignal.c:2736: instance '0x5635f5212580' has no ha
ndler with id '1248'

(granite-demo:165029): GLib-GObject-WARNING **: 10:37:29.620: ../../../gobject/gsignal.c:2736: instance '0x5635f5212390' has no ha
ndler with id '1217'
daniel@elementary-os:~/Projects$ cd appcenter/
daniel@elementary-os:~/Projects/appcenter$ ls
BACKERS.md  build  COPYING  data  meson  meson.build  meson_op
daniel@elementary-os:~/Projects/appcenter$ cd build/
daniel@elementary-os:~/Projects/appcenter/build$ ninja
[27/27] Linking target src/io.elementary.appcenter.
daniel@elementary-os:~/Projects/appcenter/build$ ninja clean
[2/2] Cleaning.
Cleaning... 134 files.
daniel@elementary-os:~/Projects/appcenter/build$ ninja
[10/73] Compiling Vala source ../src/Application.vala ../src/A
../src/Views/AbstractView.vala:35.55-35.93: warning: Access to
                    if (child is Views.AppInfoView && (child a
                                                         ^^^^^^^^

Compilation succeeded - 1 warning(s)
[73/73] Linking target src/io.elementary.appcenter.
daniel@elementary-os:~/Projects/appcenter/build$
```

```vala
/* Copyright 2015 Marvin Beckers <beckersmarvin@gmail.com>
*
* This program is free software: you can redistribute it
* and/or modify it under the terms of the GNU General Public License as
* published by the Free Software Foundation, either version 3 of the
* License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be
* useful, but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
* Public License for more details.
*
* You should have received a copy of the GNU General Public License along
* with this program. If not, see http://www.gnu.org/licenses/.
*/

public class AppCenter.App : Gtk.Application {
    public const OptionEntry[] APPCENTER_OPTIONS = {
        { "show-update", 'u', 0, OptionArg.NONE, out show_updates,
        "Display the Installed Panel", null},
        { "silent", 's', 0, OptionArg.NONE, out silent,
        "Run the Application in background", null},
        { "load-local", 'l', 0, OptionArg.FILENAME, out local_path,
        "Add a local AppStream XML file to the package list", "FILENAME" },
        { null }
    };

    private const int SECONDS_AFTER_NETWORK_UP = 60;

    public static bool show_updates;
    public static bool silent;
    public static string? local_path;
    public static AppCenterCore.Package? local_package;

    // Add "AppCenter" to the translation catalog
    public const string APPCENTER = N_("AppCenter");
```

stripe

developer.elementary.io

edw