Parental controls in GNOME

25 minutes allocated. 20 minutes for talk, 5 minutes for questions.

Hello, this is an introduction to the parental controls feature which has landed in various GNOME projects over the last year. I aim to cover it from a user perspective, and from a technical perspective. Then I'll look at how you can integrate with parental controls in your app, and finish with future plans.

I have been working on Endless OS for a few years, and one of the consistent pieces of feedback we got from users and integrators was that they needed support for parental controls — the ability to limit what non-admin users can do on some parts of the system, particularly relating to access to and installation of content (like websites, or new apps).

With the growth of flathub, a wide variety of apps are now available for users to install, some of which parents might think are not appropriate for their kids.

We implemented a first version of parental controls in Endless OS to try things out, and then started looking at upstreaming it at a hackfest in London in March 2019 (see `https://tecnocode.co.uk/2019/03/20/parental-controls-hackfest/`).

That was a productive hackfest, and there was interest in the feature from a variety of people and distros, and also interest in the related feature of ‘digital wellbeing’.

Digital wellbeing is applying limitations to yourself on a computer, in order to keep yourself on track (not wasting time, not staying on the computer for too long without a break, not working late, etc.)

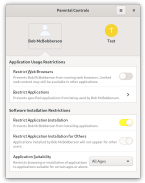

Figure: The Parental Controls app

Currently, parental controls support restricting access to specific applications chosen by the administrator, with ‘web browers’ having their own special switch. The administrator can choose to prevent the user installing any additional applications, or can allow them to only install applications which are suitable for certain age ranges. Restrictions on applications and installation currently only work with flatpak apps. They can be applied separately to each non-admin user, and can only be edited with admin privileges.

Here’s a screenshot of the parental controls app, which is separate from gnome-control-center (but linked to from it) because it’s likely to support statistics and other non-settings-like content in future.

What supports parental controls?

- gnome-shell
- flatpak
- gnome-software
- gnome-control-center
- gnome-initial-setup

There are various integration points for parental controls. The code for parental controls itself – the settings widgets and library – is implemented in the malcontent project.

gnome-shell uses `libmalcontent` to prevent disallowed apps from being launched or shown in search results.

flatpak uses it to prevent age-inappropriate apps from being listed or installed, and to prevent disallowed apps from being launched.

gnome-software uses it for the same.

gnome-control-center uses it to hide disallowed apps from its app settings panel; and also has a link through to the parental controls settings.

gnome-initial-setup has some more in-depth integration which allows parent and child users to be created at initial setup time, and allows the initial restrictions to be set on the child's

Parental controls in GNOME

Initial setup

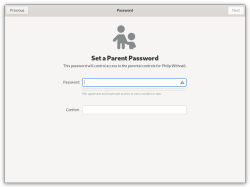

Figure: Setting a parent password in gnome-initial-setup

The flow of setting up parent and child accounts in gnome-initial-setup. If the ‘Set up parental controls for this user’ checkbox isn’t checked, the new flow is bypassed and initial setup proceeds as before.

Parental controls in GNOME

What's the uptake?

What's the uptake?

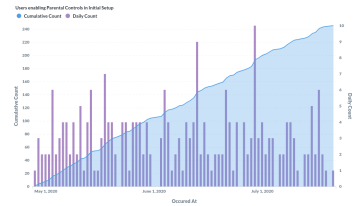

Figure: Uptake of parental controls since EOS 3.8 was released

Parental controls integration in gnome-initial-setup was first released in Endless OS 3.8 (start of May 2020), and we included some anonymous opt-in statistics reporting to measure the uptake.

This only measures users who've installed Endless OS since 3.8 was released, as users who upgraded from a previous OS version won't go through initial setup again. So far, it seems consistently about 2% of new users enable parental controls during initial setup.

How do parental controls work?

How do parental controls work?

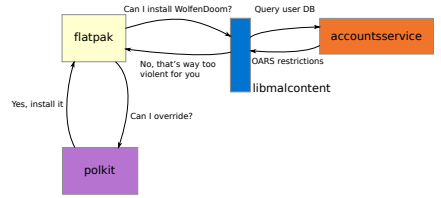

Figure: Parental controls architecture (restricting installation)

This diagram shows an example of how parental controls are implemented in flatpak to potentially disallow the user from installing an app which is not age-appropriate for them.

Say the user wants to install WolfenDoom (a violent game). flatpak passes the OARS (Open Age Ratings Service) metadata for WolfenDoom to libmalcontent (the parental controls library). This metadata contains various ratings to describe the app: whether it contains bloody violence, religious references, sexual content, etc. Each app (not just games) should have this metadata, otherwise it'll be assumed to be maximally violent, sexual, full of gambling, etc.

libmalcontent then queries accountsservice for the user's parental controls settings, which indicate what levels of violence, gambling, etc. the user is allowed to see. It then compares these settings against the app's OARS metadata to work out whether installation is allowed.

How do parental controls work?

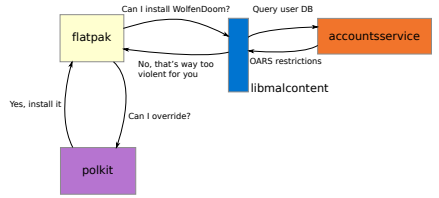

Figure: Parental controls architecture (restricting installation)

In this example, the user is not allowed to install WolfenDoom as it's too violent. flatpak then queries polkit to see if the admin wants to override malcontent and allow installation anyway. This will pop up a polkit authorisation dialogue. If the user enters the admin password, installation is allowed, otherwise it's denied.

Note that this is all implemented within the same user process space. It's not real security: a determined user will always be able to find a way around it (for example, by downloading and running a statically linked version of WolfenDoom). It should prevent the average child from doing things they're not supposed to, though.

How do parental controls work?

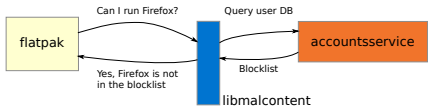

Figure: Parental controls architecture (restricting installed apps)

While restricting installation of new apps is implemented using OARS metadata, restricting access to apps which are already installed uses a blocklist. This allows the administrator to have age-inappropriate apps installed system-wide, but restricted for some users. Like all user data for parental controls, that blocklist is also stored in accountsservice.

How can I integrate my app with parental controls?

Add OARS data to your appstream:
`https://hughsie.github.io/oars/`

If you do nothing else, you should spend 5 minutes doing this: add OARS metadata to your application please! Even if your application contains no violence, gambling, advertisements or online chat, its metadata needs to say as much, otherwise we can't know.

There's an easy online questionnaire which generates the right metadata for you, and you then put it into your appdata file.

Parental controls in GNOME

How can I integrate my app with parental controls?

How can I integrate my app with parental controls?

```
<content_rating type="oars-1.1">
  <content_attribute id="violence-cartoon">moderate<
  <content_attribute id="violence-fantasy">moderate<
  <content_attribute id="social-chat">intense</cont
  <content_attribute id="social-info">intense</cont
  <content_attribute id="social-audio">none</conten
  <content_attribute id="social-location">none</con
  <content_attribute id="social-contacts">none</con
  <content_attribute id="money-purchasing">intense<
  <content_attribute id="money-gambling">none</cont
</content_rating>
```

Here's an example of some OARS metadata. Note that the attributes are not all about violence — there are attributes relating to privacy and advertising as well, for example. There's a specification on the OARS website.

If any OARS data is provided, then missing attributes are assumed to have value `none`. If no OARS data is provided, they're all assumed to have value `intense`.

How can I integrate my app with parental controls?

If your application has more complex parental controls needs – for example, it can display various kinds of content dynamically and you need to restrict some of it – you might want to use `libmalcontent` within your app to check the user's parental controls settings and internally filter what they are allowed to see or do.

There's some example code for that in the malcontent repository, but please also get in touch with me and I can help out.

How can I integrate my app with parental controls?

Use libmalcontent to implement internal filtering

Example code: `https://gitlab.freedesktop.org/pwithnall/malcontent/-/tree/master/malcontent-client`

How can I enable parental controls in my distro?

Currently, as far as I know, Endless OS is the only distribution with parental controls enabled. Enabling it in your distribution should be a matter of enabling the optional `libmalcontent` dependency in the projects listed earlier, and ensuring that all the apps you ship have correct OARS metadata.

If you allow app installation from sources other than flatpak, you may need to hook up appstream metadata for your package manager to allow gnome-software to read the OARS metadata for apps, and hence know what kinds of content they contain.

What about the future?

- Digital wellbeing
- Screen time
- Time limited sessions

At the hackfest in March 2019, various other parts of the feature were discussed, and digital wellbeing was enthusiastically received by people. I'd like to work on that next.

Screen time, in particular, is a nice and self-contained feature which should be relatively easy to implement. It would limit the amount of time you can spend on the computer without a break. If anyone is keen to work on it, I am happy to mentor and provide support.

Parental controls in GNOME

Screen time mockups

Screen time mockups

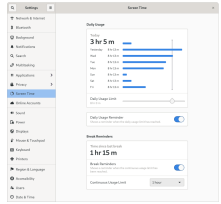

Figure: Screen time mockup (Allan Day)

Here's a mockup for how screen time functionality could look, thanks to Allan Day.