

## Documentation - Housekeeping #2026

Packaging Request # 2506 (in progress): arch introduced a base metapackage to replace the meta group

### OpenRC documentation needs clarification and expansion

2018-10-07 02:24 AM - zylwax

<b>Status:</b>	open	<b>% Done:</b>	0%
<b>Priority:</b>	discussion		
<b>Assignee:</b>			
<b>Category:</b>			
<b>Description</b>			
<p>While performing a new openrc install from a parabola-openrc-cli-dual-complete-2018.06.11.iso live medium, I discovered multiple areas of conflict between my system's state and the documentation.</p>			
<p>The first issue I encountered was that the base-openrc group brings in systemd as a dependency of mkinitcpio, which itself is a dependency of linux-libre. I found the inclusion of systemd and its dependencies surprising, since a distribution which offers an openrc-based install generally does so as an explicit alternative to systemd.</p>			
<p>Just as expected when chrooting into the pacstrap directory, the system had systemd installed alongside openrc. The systemctl command was available, as was the rc-service command. I did not test any other commands, but I feel it safe to presume that the installation is not broken at the level of pacman itself and thus that they are available.</p>			
<p>Trying to determine how to proceed, I started reading the OpenRC article (<a href="https://wiki.parabola.nu/OpenRC">https://wiki.parabola.nu/OpenRC</a>), in particular the section entitled "Replacing systemd with eudev". The procedure for removing the libsystemd metapackage with pacman -Rdds libsystemd worked without incident, but then I encountered a number of inconsistencies:</p>			
<ol style="list-style-type: none"><li>1. The next step instructed me to install the packages systemd-dummy and libsystemd-dummy, but libsystemd-dummy does not exist in the standard repositories.</li><li>2. The docs claimed that eudev would be installed along systemd-dummy as a dependency, but eudev was already installed on the system. systemd-dummy was not installed on the fresh base-openrc install.</li><li>3. The last paragraph states that "After the above steps Systemd would be uninstalled", suggesting that the system would uninstall systemd, but the user must do this manually by running pacman -Rsn systemd.</li></ol>			
<p>The last paragraph also suggests enabling the [nonsystemd] repo, highlighting this text with a hyperlink. The hyperlink jumps to the Tips and Tricks =&gt; Nonsystemd repository section of the OpenRC article itself (<a href="https://wiki.parabola.nu/OpenRC#Nonsystemd_repository">https://wiki.parabola.nu/OpenRC#Nonsystemd_repository</a>), which then links to the full Nonsystemd article (<a href="https://wiki.parabola.nu/Nonsystemd">https://wiki.parabola.nu/Nonsystemd</a>). This article instructs the user to enable a [nonsystemd] repository located above [libre] in /etc/pacman.conf. After providing this update to the pacman configuration, the user is asked to run pacman -Syu, which in my case brought in a mkinitcpio package without systemd. Now the user must install the your-initfreedom package, which provides a conflict with systemd to prevent its installation and remove it should it be installed. After asking the user whether to remove several systemd-based packages including libsystemd, Pacman ultimately fails to install your-initfreedom, citing a number of dependencies which removing libsystemd would break.</p>			
<p>Based on the above information, here are what I surmise is the correct, current intervention to install Parabola without systemd from a fresh base-openrc pacstrap:</p>			
<ol style="list-style-type: none"><li>1. pacman -Rdds libsystemd</li><li>2. pacman -S systemd-dummy</li><li>3. pacman -Rsn systemd</li></ol>			
<p>These steps successfully added systemd-dummy and removed systemd from my fresh base-openrc install.</p>			
<p>What is clear right now is that the existing situation is inadequate on several levels. It is surprising enough that the existing base-openrc package group contains the real systemd package, and the documentation on how to remove is quite convoluted and partially outdated. I also have no indication yet as to whether these steps will result in me having the most stable and maintainable systemd-less installation.</p>			
<p>Not being in the openrc development team, I do not know what direction Parabola is taking with regards on how it will support openrc-based installations. I welcome those who are part of this team to clarify this situation into a proper procedure, whatever it may be.</p>			
<p>I have some logs of the previous procedures available upon request (generated with script, so quite messy -- I can try to clean them up if so desired). I also have a snapshot of a fresh install which I can probe with any desired commands for troubleshooting purposes.</p>			

**Related issues:**

Related to Packages - Freedom Issue #2095: [libsystemd-dummy] unwanted depend...

**not-a-bug**

Related to Documentation - Bug #1989: recommendation on wiki page to install ...

**open****History****#1 - 2018-10-22 06:24 PM - j2rm**

Hello friend, some weeks ago I tried to install Parabola OpenRC, and the same incidents that you describe here appeared to me. So I could not complete the installation. Could you put in this same thread all the commands you used to get a clean installation of Parabola with OpenRC?

**#2 - 2018-11-16 12:40 PM - zylwax**

Hello j2rm,

You'll have to pardon my lengthy delay in replying back to you as I only just today checked for activity reminders.

I'm happy to report that this install is still going strong without systemd, even after being restored from backups on account of the filesystem going corrupt. Unfortunately, the deployment requires libsystemd to avoid experiencing errors with various programs complaining about missing libraries. I hope that you can live with this. I personally find it alright because the systemd binary itself is not in the final install.

It appears that I no longer have the documentation on this subject due to the filesystem crash, which clobbered my snapshots detailing various branches involving the different setups I tested. As a result, I am recalling what I am pretty sure is the correct procedure from memory.

From a fresh base-openrc install:

1. pacman -S systemd-dummy
2. pacman -Rsn systemd

It's pretty much the same routine as I initially documented without the removal of the libsystemd libraries. Since the libsystemd-dummy package no longer exists, this is the best way to ensure a stable system. Removing the systemd package should also pull out various systemd-\* packages.

This is really shoddy documentation, and I sincerely apologize for my lack of greater care in developing a more cohesive documentation strategy for my setup. Nonetheless, I am really enjoying using openrc on my system to manage services. I highly suggest you tactfully read the openrc documentation and employ its most intelligent suggestions in configuring the rest of your system.

Hopefully I will check my reminders more frequently, but further comments on this thread **will** send me a reminder, so please feel free to try getting my attention again this way. I'll do my best to answer any further questions you have.

**#3 - 2018-12-23 08:27 PM - anon7mous**

[zylwax](#) how do you installed an Desktop-Environment like XFCE? If I want to install [[https://www.parabola.nu/groups/686/openrc-desktop/\[openrc-desktop\]](https://www.parabola.nu/groups/686/openrc-desktop/[openrc-desktop])] in greater detail [[https://www.parabola.nu/packages/extra/686/xorg-server/\[xorg-server\]](https://www.parabola.nu/packages/extra/686/xorg-server/[xorg-server])] it has systemd as dependencies. I'm not able to run a DE.

**#4 - 2018-12-24 04:54 AM - bill-auger**

FWIW - openrc/libsystemd/elogind and friends were in a state of flux for the first year or so - i do think it has stabilized recently and the openrc article was updated a few days after this bug report was opened to reflect the current install procedure - presumably this bug report should have been closed then - are there currently any discrepancies that need to be addressed?

**#5 - 2018-12-24 05:08 AM - bill-auger**

anon7mous -

xorg-server does not depend on systemd - you may be referring to the build dependency listed as "systemd (make)" - that is to say, installing the xorg-server package will not install systemd

installing XFCE should be nothing more than:

```
# pacman -Sy openrc-desktop xfce4
```

if that does not work, it would be a bug that deserves its own bug report

**#6 - 2018-12-24 03:53 PM - zylwax**

bill-auger: I do see that the OpenRC article was updated about two days after my large update by lukeshu. It seems to address the bulk of the discrepancies I experienced in my still-working install. The only thing I feel worth testing at this point is activating the [nonsystemd] repo, which I would prefer to do after making a backup. Once it's complete, I will give it a trial and close this issue if I don't unearth any major problems.

anon7mous: My system does not list systemd as a dependency as xfce4 on my own system, and I have no reason to doubt bill-auger's solution will work for your needs. As requested, please start a new bug report if you would like to pursue this issue further.

## #7 - 2018-12-24 04:49 PM - zylwax

I just tried out the nonsystemd repo on my system and it appears to be doing its job.

Here was my procedure:

1. Enable the [nonsystemd] repo and run `pacman -Syu`
2. Attempt to install `your-initfreedom`, but `pacman` refused to uninstall the `systemd-nss-*` packages since my setup still had `libsystemd`
3. I notice that that the `pcr/libsystemd-dummy` package now exists again, so I ran `pacman -S libsystemd-dummy` then `pacman -Rsn libsystemd`. Removing the real `libsystemd` package also removes the `systemd-nss-*` packages.
4. I attempt to install `your-initfreedom` again and it works without incident.

With this procedure stated, the only issue I see with the documentation is the lack of any mention for the now-reinstated `libsystemd-dummy` package. Looking over the details for this package along with the `base-openrc` package group, I don't see any indication this package gets pulled in automatically in the course of creating a fresh `base-openrc` install, so I feel it deserves a mention somewhere in the docs.

Lastly, there is a scary warning on [https://wiki.parabola.nu/OpenRC#Nonsystemd\\_repository](https://wiki.parabola.nu/OpenRC#Nonsystemd_repository) that some essential configuration files can be overwritten upon activating and updating your system with the [nonsystemd] repository. While I didn't have any issues with this, it still seems worthwhile to leave the warning in place given how dire a breakage it would represent.

## #8 - 2018-12-25 07:44 AM - bill-auger

indeed, i confirm that it is not possible to install 'your-initfreedom' on an installed openrc system - i would say this is a bug

```
# yes | pacman -S your-initfreedom
resolving dependencies...
looking for conflicting packages...
:: removing systemd-nss-systemd breaks dependency 'nss-systemd' required by libsystemd
:: removing systemd-nss-myhostname breaks dependency 'nss-myhostname' required by libsystemd
:: removing systemd-nss-mymachines breaks dependency 'nss-mymachines' required by libsystemd
:: removing systemd-nss-resolve breaks dependency 'nss-resolve' required by libsystemd
```

the result is the same with ``pacman -S your-initfreedom libsystemd-dummy`` - one must first install 'libsystemd-dummy' and only then will 'your-initfreedom' be satisfiable - im still not clear what is the purpose of 'your-initfreedom'; but it does not seem to be very helpful for the purpose of migrating between inits, as the package name would suggest

the same conflict arises when installing 'base-openrc' with the [nonsystemd] repo enabled

```
# pacstrap /mnt base-openrc
==> Creating install root at /mnt/
==> Installing packages to /mnt/
:: Synchronizing package databases...
nonsystemd is up to date
libre is up to date
core is up to date
extra is up to date
community is up to date
pcr is up to date
pcr-testing is up to date
:: There are 29 members in group base-openrc:
:: Repository nonsystemd
   1) filesystem  2) your-initfreedom
:: Repository libre
   3) licenses  4) linux-libre  5) pacman  6) pacman-mirrorlist  7) your-freedom
:: Repository pcr
   8) base-meta  9) cronie-openrc  10) cryptsetup-openrc  11) dbus-openrc  12) device-mapper-openrc  13)
```

```
Enter a selection (default=all):
resolving dependencies...
looking for conflicting packages...
:: your-initfreedom and systemd-nss-myhostname are in conflict
```

note that is is also not possible to install 'your-initfreedom' on an installed systemd system; so it is impossible to install 'your-initfreedom' on any installed parabola system, without first "jumping through some hoop"

```
# yes | pacman -S your-initfreedom
resolving dependencies...
looking for conflicting packages...
:: your-initfreedom and netctl are in conflict. Remove netctl? [y/N] y
:: your-initfreedom and systemd-nss-myhostname are in conflict. Remove systemd-nss-myhostname? [y/N] y
:: your-initfreedom and systemd-nss-mymachines are in conflict. Remove systemd-nss-mymachines? [y/N] y
:: your-initfreedom and systemd-nss-resolve are in conflict. Remove systemd-nss-resolve? [y/N] y
:: your-initfreedom and systemd-nss-systemd are in conflict. Remove systemd-nss-systemd? [y/N] y
error: failed to prepare transaction (could not satisfy dependencies)
:: removing systemd-nss-systemd breaks dependency 'nss-systemd' required by libsystemd
:: removing systemd-nss-myhostname breaks dependency 'nss-myhostname' required by libsystemd
```

```
:: removing systemd-nss-mymachines breaks dependency 'nss-mymachines' required by libsystemd
:: removing systemd-nss-resolve breaks dependency 'nss-resolve' required by libsystemd
```

as i mentioned, parabola's init ecosystem has been in a state of flux since the beginning with multiple devs working on it in different directions - it has been a long time coming; but we seem to be homing in on a standard procedure/configuration; so this is a good time to get this nailed down and remove as much confusion as possible once and for all - (and that is not to mention the confusion with the udev and consolekit/elogind options) - i did some experiments today to try detangling the general install procedure which raised some very obvious questions - i will post them separately below

#### #9 - 2018-12-25 07:46 AM - bill-auger

- Priority changed from bug to discussion

- Tracker changed from Bug to Housekeeping

#### #10 - 2018-12-25 08:11 AM - bill-auger

installing 'base-openrc' pulls in several systemd packages - is this not precisely what openrc support intends to avoid? - this, as many may be aware, is a common point of contention from users seeing these "systemd" packages - it is one thing to explain that a package like 'libsystemd-dummy' is not any part of systemd, or that 'libsystemd' and 'elogind' are parts of systemd that some programs require, but are not themselves the 'systemd' init system; but the packages that 'base-openrc' pulls in are the 'systemd' init system - so whats the deal with that?

```
# pacstrap /mnt base-openrc --print | grep systemd | grep -v '/nonsystemd/'
.../systemd-libsystemd-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-nss-systemd-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-nss-myhostname-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-nss-mymachines-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-nss-resolve-239.370-1.par1-x86_64.pkg.tar.xz
.../libsystemd-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-common-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-239.370-1.par1-x86_64.pkg.tar.xz
```

the result is the same after enabling the [nonsystemd] repo, and also with:

```
# pacstrap /mnt base-openrc your-initfreedom
```

however, if 'libsystemd-dummy' is given explicitly to pacstrap, the results are different - and also if 'your-initfreedom' is given as well, so 'your-initfreedom' is not playing any obvious role here either)

```
# pacstrap /mnt base-openrc libsystemd-dummy --print | grep systemd | grep -v '/nonsystemd/'
or
# pacstrap /mnt base-openrc libsystemd-dummy your-initfreedom --print | grep systemd | grep -v '/nonsystemd/'
.../systemd-libsystemd-239.370-1.par1-x86_64.pkg.tar.xz
.../libsystemd-dummy-1:1-1-any.pkg.tar.xz
.../systemd-common-239.370-1.par1-x86_64.pkg.tar.xz
.../systemd-239.370-1.par1-x86_64.pkg.tar.xz
```

clearly, it is difficult to explain to concerned users that an openrc system will not have systemd as it's init; and that the 'systemd' package that 'base-openrc' requires is not systemd - to add to the confusion, there are these packages:

- libre/notsystemd
- libre/notsystemd-common
- libre/notsystemd-udev
- pcr/systemd-dummy

from the package names alone, these seem to be (semantically) what would be expected to be installed (rather than 'systemd', 'systemd-common', etc) if they exist simply to satisfy frivolous, zealous, or "erroneous" dependencies

#### #11 - 2018-12-25 08:18 AM - bill-auger

regarding, 'base-openrc' and providers of 'openrc-pid1', the openrc wiki page reads:

```
"You have a few choices you can make, or accept the defaults of what's in base-openrc."
```

that is quite confusing because in fact, the user is not presented with any choices of 'openrc-pid1' - the only options are the packages in the 'base-openrc' group, which includes 'openrc-init' - so i assume that is to say that there are alternative options for 'openrc-pid1', but if one installs the 'base-openrc' group (the most common situation), then one would need to replace the standard 'openrc-init' with some other provider of 'openrc-pid1' afterward, or manually pick and choose all the other 28 packages in the 'base-openrc' group other than 'openrc-init'; as it is not possible to install 'base-openrc' with any other 'openrc-pid1' provider - it then goes on to say:

```
"After installing an openrc-pid1 provider, OpenRC should boot by default instead of systemd."
```

... which is probably not worth mentioning; because 'base-openrc' installs 'openrc-init' and it can not be removed (only replaced with some other 'openrc-pid1' provider) - so some 'openrc-pid1' provider will be installed at that point in the tutorial

the next section describes an analogous phantom "choice" of udev provider - 'base-openrc' installs eudev, and it is not possible to install 'base-openrc' with any other 'udev' provider

my expertise in this area is not very deep, but i will repeat my suggestion from a while back, which is based primarily on semantics; but would make this more sensible and robust if it is possible - that is: for a package with a name such as 'your-initfreedom' should be installed on every parabola system, and it should be the only parabola package that depends=( ) on **some** init system (currently one of: systemd, openrc), and for each init system to, in turn, depends=( ) some applicable provider of these options for pid1 and udev - for example: such that installing 'your-initfreedom' would force the user to choose (and/or allow easy switching to and from) any one init system at a time to satisfy 'your-initfreedom', and then (if applicable) force the user to choose one of each pid1 and udev alternative to satisfy the previously chosen init system, which finally would fully satisfy the 'base-openrc' group, making it possible to install the 'base-openrc' group with any of the applicable pid1 and udev alternatives - in that way it should be possible to eliminate the 'base-openrc' group and use the dependency mechanism to its full advantage instead in combination with the presence/absence of the nonsystemd repo - that should be enough to robustly determine which package should be installed, or at least to present the user with options that are intuitive

e.g.

```
There are two providers of 'your-initfreedom':
(1)systemd-meta , (2)openrc-meta
Choose an option: _
```

... and if the [nonsystemd] was not enabled (the default case), 'systemd-meta' would be the only available provider of 'your-initfreedom'

again, im not sure how feasible that would be (it may require ALPM hooks or some debian-style "what do you want to do?" dialog); but it is more intuitive and would go a long way toward removing confusion and making init transitions simpler and more robust

semantically speaking, such alternatives (that is: the ones that are not in the arch base) should be in the [notsystemd] repo (along with anything else outside the base distro that exists only for the purpose of init sanity); because they are not relevant for the standard parabola base system which is intended to be as close to arch as possible

#### #12 - 2018-12-25 08:22 AM - bill-auger

it is actually possible to install 'openrc-init' on an installed systemd system - would this not lead to some unexpected or very broken behavior?

if indeed the wiki is correct in that:

```
"After installing an openrc-pidl provider, OpenRC should boot by default instead of systemd."
```

... then we are in a situation where both openrc and systemd appear to be fully installed at the same time - even if this situation is sane, it is very confusing and it would be hard to imagine any value in that configuration

```
# pacman -S openrc-init
resolving dependencies...
looking for conflicting packages...
:: openrc-init and systemd-sysvcompat are in conflict (init). Remove systemd-sysvcompat? [y/N] y
```

```
Packages (3) openrc-0.38.2-2  systemd-sysvcompat-239.300-1.par1 [removal]  openrc-init-0.38.2-2
```

i suggest that it would be highly desirable to make openrc and systemd mutually exclusive options (as well as shepherd and any other inits that come along) by using the dependency mechanism to the fullest extent possible, and avoiding multiple 'base-\*' package groups as the preferred install/migration method (such as the idea in comment # 11)

#### #13 - 2018-12-25 08:26 AM - bill-auger

a related question to the previous: 'openrc-init' both "provides" and "conflicts" with 'openrc-pidl' and 'init'; which is a contradiction semantically - that literally implies that this package is in conflict with itself - presumably the intention is to make it mutually exclusive with the other pid1 providers; but this way of accomplishing that is very confusing - wouldnt it be more clear to make 'openrc-pidl' conflict explicitly with the other two 'openrc-pidl' providers; and to make 'init' conflict explicitly with the other four 'init' providers? - (the latter is much in line with my suggestion in comment # 11 above)

```
# pacman -S openrc-pidl
:: There are 3 providers available for openrc-pidl:
:: Repository pcr
  1) openrc-init  2) openrc-sysvinit  3) runit-scripts

# pacman -S init --ignore systemd-sysvcompat
:: systemd-sysvcompat is in IgnorePkg/IgnoreGroup. Install anyway? [Y/n] n
:: There are 4 providers available for init:
:: Repository pcr
  1) openrc-init  2) runit-replaceinit  3) runit-scripts  4) sysvinit
```

note that: according to pacman, the providers of 'init' are not the same set as those mentioned on the wiki ('openrc-init', 'openrc-sysvinit', 'runit-scripts')

#### #14 - 2019-01-09 04:38 PM - bill-auger

the latest openrc ISOs also have this problem - they have the [nonsystemd] repo enabled; but it is not possible to install 'your-initfreedom' (a dependency of 'base-openrc') with the [nonsystemd] repo enabled

```
your-initfreedom and systemd-nss-myhostname are in conflict
```

after disabling the [nonsystemd] repo, the install succeeds; but then is not possible to install 'base-devel'

systemd-libudev and eudev-libudev are in conflict

this is the case whether [nonsystemd] is re-enabled or not

**#15 - 2019-01-13 01:08 AM - bill-auger**

- Related to Freedom Issue #2095: [libsystemd-dummy] unwanted dependency added

**#16 - 2019-01-31 02:41 AM - Megver83**

tl;dr but I've a few things to say about all of this mix of confusions, package conflicts, etc.

In the past (2017 and early 2018) when only a few ppl were in charge of OpenRC and related stuff (like elogind, etc.) nothing of this happened. Everything began when from one day to another everybody wanted to collaborate with OpenRC and other inits (because after the launching of the first OpenRC ISOs then the interest for a full-supported non-Systemd Parabola system increased), which for me it's OK, the problem was that instead of working coordinately everyone worked separately.

Now, what I think is the best solution, is to assign the most qualified packagers to be in charge of all the init stuff (like an "Init packagers team") and these ppl should communicate with each other and coordinate efforts, not split them like now. I think that the best way to choose these devs is that they offer themselves (after all, that's the way Parabola devs work), and if someone else who doesn't "officially" belongs to this groups would like to eventually collaborate with OpenRC/runit/sysvinit/etc. just talk to one of these devs so we can avoid these kind of problems.

If you agree with me, I offer myself to belong to this team. I think that other devs interested in this may be: [bill-auger](#) and [lukeshu](#)

After assigning the packagers, I could write a wiki page documenting all of this, like who's in charge of OpenRC init scripts, openrc pkg itself, runit, runit scripts, sysvinit, systemd, elogind pkgs, nonsystemd pkgs, why package X is named like this, what does it do, why is it important, etc. (systemd already belongs to lukeshu afaik). I know this may sound like too much work, but as I said before, it is to avoid problems like this in the future.

**#17 - 2019-01-31 04:22 AM - bill-auger**

this all sounds good but it misses the main point of this discussion, which is that the current situation is too confusing and possibly no one person actually knows what it all means - perhaps it was a lack of communication, but regardless of how it came to be this way, the very purpose of this ticket is hopefully to get that communication flowing now and get everything well-documented

whether i or anyone else chooses to help with maintaining these packages or not, the very first thing that would need to happen before anyone new could begin, is that the questions raised in this ticket would need to be explained in some detail - i have been taking some notes of things i have asked and read on #parabola IRC over the past year or so, but i do not have enough to make complete sense from

right now, there are too few people who be able to begin explaining to anyone where to begin - i would be satisfied just to understand the arrangement well enough to write it down on the wiki

**#18 - 2019-01-31 04:35 AM - bill-auger**

i have not proposed anything like this before, but it would be very helpful if we could try arranging for as many parabola devs as possible to meet for a real-time mumble chat or jitsi-meet or even just to be on IRC at the same time; at least if only to answer some of the questions in this ticket

**#19 - 2019-02-13 03:52 AM - bill-auger**

as if this was not confusing enough, arch has just renamed 'libsystemd' to 'systemd-libs', in order to be named consistently with others such as: gcc-libs and boost-libs

this is now in staging and presumably will be hitting the main-line soon - 'systemd-libs' provides, conflicts with, and replaces 'libsystemd'

**#20 - 2019-02-17 03:47 PM - Megver83**

bill-auger wrote:

i have not proposed anything like this before, but it would be very helpful if we could try arranging for as many parabola devs as possible to meet for a real-time mumble chat or jitsi-meet or even just to be on IRC at the same time; at least if only to answer some of the questions in this ticket

agree, maybe you could write about in the mailing list to specify the Mumble server where we can meet, the date, etc.

**#21 - 2019-03-02 08:14 PM - bill-auger**

question the next:

'your-initfreedom' declares itself to be in the base-openrc group, yet it is in the [nonsystemd] repo which is not enabled by default on any ISO or installed system

**#22 - 2019-03-02 08:19 PM - bill-auger**

according to the description of 'systemd-dummy', it is only useful for systems without systemd; but the only way to accomplish that AFAIK is to enable the [nonsystemd] repo - yet the 'systemd-dummy' package is in [pcr] - this seems like a design smell - yes? - semantically it should be in the [nonsystemd] repo and perhaps even required by 'your-initfreedom'

```
$ pacman -Ss systemd-dummy
```

```
pcr/systemd-dummy 1:1-1
```

```
An empty package that provides 'systemd' to satisfy packages that erroneously depend on it
```

also, "erronously" is correctly spelled "erroneously"; but is erroneously spelled "erronously" in that package description :)

### #23 - 2019-03-02 08:21 PM - bill-auger

- Subject changed from *OpenRC installation documentation appears to need a major rewrite to OpenRC documentation needs clarification and expansion*

### #24 - 2019-03-25 01:56 AM - bill-auger

what would be the best way for scripts to detect the initsystem

it was suggested on the mailing list to determine the appropriate service commands by deteing the presence of some associated files

```
if [ -e /sys/fs/cgroup/systemd ] ; then systemctl start SOME_SERVICE.service ;
elif [ -e /run/openrc/softlevel ] ; then rc-service SOME_SERVICE start ;
elif [ -e /etc/rc.d/SOME_SERVICE ] ; then rc.d start SOME_SERVICE ;
```

perhaps asking pacman is more elegant?

```
readonly INIT=$(pacman -Qq init)
case "$INIT" in
  'systemd-sysvcompat' ) systemctl start SOME_SERVICE.service ;;
  'openrc-init' | \
  'runit-replaceinit' | \
  'runit-scripts' ) rc-service SOME_SERVICE start ;;
  'sysvinit' ) rc.d start SOME_SERVICE ;;
esac
```

### #25 - 2019-03-25 10:28 AM - josealberto4444

I'd go for the first, as it is distro-agnostic. Even though I agree with the second one being more elegant, I think the first one use more basic tools, and I think that is important.

### #26 - 2019-04-16 11:32 PM - bill-auger

- Related to *Bug #1989: recommendation on wiki page to install polkit-elogind fails on fresh openrc installation added*

### #27 - 2019-04-16 11:37 PM - bill-auger

'libsystemd-dummy' has been replaced/renamed with 'systemd-libs-dummy'

the bare minimum pacstrap install for an openrc system is:

```
# pacstrap /mnt base-openrc systemd-libs-dummy
```

or, for an openrc desktop system:

```
# pacstrap /mnt base-openrc systemd-libs-dummy systemd-dummy polkit-elogind openrc-desktop
```

### #28 - 2019-08-31 02:15 PM - BetaRays

systemd-libs-dummy installs systemd-libsystemd but conflicts with systemd-libs. What is the difference between the systemd-libsystemd and systemd-libs and why is the dummy package needed?

### #29 - 2019-11-15 04:01 PM - bill-auger

- Parent task set to #2506