

**NAME**

org.freedesktop.hostname1 – The D–Bus interface of systemd–hostnamed

**INTRODUCTION**

**systemd-hostnamed.service**(8) is a system service that can be used to control the hostname and related machine metadata from user programs. This page describes the hostname semantics and the D–Bus interface.

**THE D–BUS API**

The service exposes the following interfaces on the bus:

```
node /org/freedesktop/hostname1 {
  interface org.freedesktop.hostname1 {
    methods:
      SetHostname(in s hostname,
                  in b interactive);
      SetStaticHostname(in s hostname,
                        in b interactive);
      SetPrettyHostname(in s hostname,
                        in b interactive);
      SetIconName(in s icon,
                  in b interactive);
      SetChassis(in s chassis,
                 in b interactive);
      SetDeployment(in s deployment,
                    in b interactive);
      SetLocation(in s location,
                  in b interactive);
      GetProductUUID(in b interactive,
                     out ay uuid);
      Describe(out s json);
    properties:
      readonly s Hostname = '...';
      readonly s StaticHostname = '...';
      readonly s PrettyHostname = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s DefaultHostname = '...';
      readonly s HostnameSource = '...';
      readonly s IconName = '...';
      readonly s Chassis = '...';
      readonly s Deployment = '...';
      readonly s Location = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s KernelName = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s KernelRelease = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s KernelVersion = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s OperatingSystemPrettyName = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s OperatingSystemCPENAME = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s HomeURL = '...';
      @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
      readonly s HardwareVendor = '...';
```

```

    @org.freedesktop.DBus.Property.EmitsChangedSignal("const")
    readonly s HardwareModel = '...';
};
interface org.freedesktop.DBus.Peer { ... };
interface org.freedesktop.DBus.Introspectable { ... };
interface org.freedesktop.DBus.Properties { ... };
};

```

Whenever the hostname or other metadata is changed via the daemon, **PropertyChanged** signals are sent out to subscribed clients. Changing a hostname using this interface is authenticated via [polkit](#)<sup>[1]</sup>.

## SEMANTICS

The *StaticHostname* property exposes the "static" hostname configured in `/etc/hostname`. It is not always in sync with the current hostname as returned by the **gethostname**(3) system call. If no static hostname is configured this property will be the empty string.

When **systemd**(1) or **systemd-hostnamed.service**(8) set the hostname, this static hostname *has the highest priority*.

The *Hostname* property exposes the actual hostname configured in the kernel via **sethostname**(3). It can be different from the static hostname. This property is never empty.

The *PrettyHostname* property exposes the *pretty hostname* which is a free-form UTF-8 hostname for presentation to the user. User interfaces should ensure that the pretty hostname and the static hostname stay in sync. E.g. when the former is "Lennart's Computer" the latter should be "lennarts-computer". If no pretty hostname is set this setting will be the empty string. Applications should then find a suitable fallback, such as the dynamic hostname.

The *DefaultHostname* property exposes the default hostname (configured through **os-release**(5), or a

fallback set at compilation time).

The *HostnameSource* property exposes the origin of the currently configured hostname. One of "static" (set from /etc/hostname), "transient" (a non-permanent hostname from an external source), "default" (the value from os-release or the compiled-in fallback).

The *IconName* property exposes the *icon name* following the XDG icon naming spec. If not set, information such as the chassis type (see below) is used to find a suitable fallback icon name (i.e. "computer-laptop" vs. "computer-desktop" is picked based on the chassis information). If no such data is available, the empty string is returned. In that case an application should fall back to a replacement icon, for example "computer". If this property is set to the empty string, the automatic fallback name selection is enabled again.

The *Chassis* property exposes a *chassis type*, one of the currently defined chassis types: "desktop", "laptop", "server", "tablet", "handset", as well as the special chassis types "vm" and "container" for virtualized systems. Note that in most cases the chassis type will be determined automatically from DMI/SMBIOS/ACPI firmware information. Writing to this setting is hence useful only to override misdetected chassis types, or to configure the chassis type if it could not be auto-detected. Set this property to the empty string to reenable the automatic detection of the chassis type from firmware information.

Note that systemd-hostnamed starts only on request and terminates after a short idle period. This effectively means that **PropertyChanged** messages are not sent out for changes made directly on the files (as in: administrator edits the files with vi). This is the intended behavior: manual configuration changes should require manual reloading.

The transient (dynamic) hostname exposed by the *Hostname* property maps directly to the kernel hostname. This hostname should be assumed to be highly dynamic, and hence should be watched directly, without depending on **PropertyChanged** messages from systemd-hostnamed. To accomplish this, open /proc/sys/kernel/hostname and **poll**(3) for **SIGHUP** which is triggered by the kernel every time the hostname changes. Again: this is special for the transient (dynamic) hostname, and does not apply to the configured (fixed) hostname.

Applications may read the hostname data directly if hostname change notifications are not necessary. Use **gethostname**(3), /etc/hostname (possibly with per-distribution fallbacks), and **machine-info**(3) for that. For more information on these files and syscalls see the respective man pages.

*KernelName*, *KernelRelease*, and *KernelVersion* expose the kernel name (e.g. "Linux"), release (e.g. "5.0.0-11"), and version (i.e. the build number, e.g. "#11") as reported by **uname**(2).

*OperatingSystemPrettyName*, *OperatingSystemCPENAME*, and *HomeURL* expose the *PRETTY\_NAME=*, *CPE\_NAME=* and *HOME\_URL=* fields from **os-release**(5). The purpose of those properties is to allow remote clients to access this information over D-Bus. Local clients can access the information directly.

## Methods

**SetHostname()** sets the transient (dynamic) hostname, which is used if no static hostname is set. This value must be an internet-style hostname, 7-bit lowercase ASCII, no special chars/spaces. An empty string will unset the transient hostname.

**SetStaticHostname()** sets the static hostname which is exposed by the *StaticHostname* property. When called with an empty argument, the static configuration in /etc/hostname is removed. Since the static hostname has the highest priority, calling this function usually affects also the *Hostname* property and the effective hostname configured in the kernel.

**SetPrettyHostname()** sets the pretty hostname which is exposed by the *PrettyHostname* property.

**SetIconName()**, **SetChassis()**, **SetDeployment()**, and **SetLocation()** set the properties *IconName* (the name of the icon representing for the machine), *Chassis* (the machine form factor), *Deployment* (the system deployment environment), and *Location* (physical system location), respectively.

*PrettyHostname*, *IconName*, *Chassis*, *Deployment*, and *Location* are stored in /etc/machine-info. See **machine-info**(5) for the semantics of those settings.

**GetProductUUID()** returns the "product UUID" as exposed by the kernel based on DMI information in

/sys/class/dmi/id/product\_uuid. Reading the file directly requires root privileges, and this method allows access to unprivileged clients through the polkit framework.

**Describe()** returns a JSON representation of all properties in one.

### Security

The *interactive* boolean parameters can be used to control whether polkit should interactively ask the user for authentication credentials if required.

The polkit action for **SetHostname()** is org.freedesktop.hostname1.set-hostname. For **SetStaticHostname()** and **SetPrettyHostname()** it is org.freedesktop.hostname1.set-static-hostname. For **SetIconName()**, **SetChassis()**, **SetDeployment()** and **SetLocation()** it is org.freedesktop.hostname1.set-machine-info.

## RECOMMENDATIONS

Here are three examples that show how the pretty hostname and the icon name should be used:

- When registering DNS-SD services: use the pretty hostname in the service name, and pass the icon name in the TXT data, if there is an icon name. Browsing clients can then show the server icon on each service. This is especially useful for WebDAV applications or UPnP media sharing.
- Set the bluetooth name to the pretty hostname.
- When your file browser has a "Computer" icon, replace the name with the pretty hostname if set, and the icon with the icon name, if it is set.

To properly handle name lookups with changing local hostnames without having to edit /etc/hosts, we recommend using systemd-hostnamed in combination with **nss-myhostname(3)**.

Here are some recommendations to follow when generating a static (internet) hostname from a pretty name:

- Generate a single DNS label only, not an FQDN. That means no dots allowed. Strip them, or replace them with "-".
- It's probably safer to not use any non-ASCII chars, even if DNS allows this in some way these days. In fact, restrict your charset to "a-zA-Z0-9" and "-". Strip other chars, or try to replace them in some smart way with chars from this set, for example "ä" → "ae", and use "-" as the replacement for all punctuation characters and whitespace.
- Try to avoid creating repeated "-", as well as "-" as the first or last char.
- Limit the hostname to 63 chars, which is the length of a DNS label.
- If after stripping special chars the empty string is the result, you can pass this as-is to systemd-hostnamed in which case it will automatically use a suitable fallback.
- Uppercase characters should be replaced with their lowercase equivalents.

Note that while systemd-hostnamed applies some checks to the hostname you pass they are much looser than the recommendations above. For example, systemd-hostnamed will also accept "\_" in the hostname, but we recommend not using this to avoid clashes with DNS-SD service types. Also systemd-hostnamed allows longer hostnames, but because of the DNS label limitations, we recommend not making use of this.

Here are a couple of example conversions:

- "Lennart's PC" → "lennarts-pc"
- "Müllers Computer" → "muellers-computer"
- "Voran!" → "voran"
- "Es war einmal ein Männlein" → "es-war-einmal-ein-maennlein"
- "Jawoll. Ist doch wahr!" → "jawoll-ist-doch-wahr"
- "" → "localhost"
- "...zack!!! zack!..." → "zack-zack"

ORG.FREEDESKTOP.HOSTNAME1(5)    org.freedesktop.hostname1    ORG.FREEDESKTOP.HOSTNAME1(5)

Of course, an already valid internet hostname label you enter and pass through this conversion should stay unmodified, so that users have direct control of it, if they want — by simply ignoring the fact that the pretty hostname is pretty and just edit it as if it was the normal internet name.

## VERSIONING

These D-Bus interfaces follow [the usual interface versioning guidelines](#)<sup>[2]</sup>.

## EXAMPLES

**Example 1. Introspect org.freedesktop.hostname1 on the bus**

```
$ gdbus introspect --system \  
  --dest org.freedesktop.hostname1 \  
  --object-path /org/freedesktop/hostname1
```

## SEE ALSO

David Zeuthen's original Fedora [Feature page about xdg-hostname](#)<sup>[3]</sup>

## NOTES

1. polkit  
<https://www.freedesktop.org/software/polkit/docs/latest/>
2. the usual interface versioning guidelines  
<http://0pointer.de/blog/projects/versioning-dbus.html>
3. Feature page about xdg-hostname  
<https://fedoraproject.org/wiki/Features/BetterHostname>