

---

# busctl

Examiner et superviser le bus D-Bus

## OPTIONS

- address=ADDRESS** Se connecte au bus spécifié par l'adresse au lieu d'utiliser celui par défaut
- show-machine** En affichant la liste des paires, affiche une colonne contenant les noms des conteneurs auxquels ils appartiennent.
- unique** En affichant la liste des paires, affiche seulement les noms uniques (sous la forme :number.number)
- acquired** L'opposé de **-unique**. Uniquement les noms connus sont affichés.
- activatable** En affichant la liste des paires, affiche seulement les paires qui ne sont pas encore activés, mais peuvent être démarrés automatiquement si accédés.
- match=MATCH** En affichant les messages échangés, affiche seulement le sous-jeu correspondant au match
- size=** Utilisé avec la commande capture, spécifie la taille de message bus max à capturer. Défaut : 4096
- list** Utilisé avec la commande tree, affiche une liste plate de chemins d'objets au lieu d'une arborescence.
- quiet** Utilisé avec la commande call, supprime l'affichage du payload de réponse. Noter que cela n'empêche pas les erreurs retournées d'être affichés.
- verbose** Avec les commandes call et get-property, affiche dans un format plus verbeux
- expect-reply=BOOL** Utilisé avec la commande call, spécifie si busctl devrait attendre la fin du call de méthode, afficher la donnée de la réponse de la méthode, et retourner un succès ou un échec via de code de sortie du process. Défaut : yes
- auto-start=BOOL** Utilisé avec la commande call, spécifi si le call de méthode devrait implicitement activer le service appelé, qui ne devrait pas être encore en cours d'exécution mais est configuré en auto-start. Défaut : yes
- allow-interactive-authorization=BOOL** Utilisé avec la commande call, spécifie si les services peuvent forcer l'autorisation interactive tout en exécutant l'opération, si la stratégie de sécurité est configurée pour cela. Défaut : yes
- timeout=SECS** Utilisé avec la commande call, spécifie le temps max d'attente pour la fin du call de methode. Par défaut, assume des secondes (peut être spécifié avec ms, us, s, min, h, d, w, month, y). Noter que ce timeout ne s'applique pas si **-expect-reply=no** est utilisé. Défaut : 25s.
- augment-creds=BOOL** Contrôle si les données d'accréditifs reportées par list ou status devraient être augmenté avec les données de /proc. Activé, les données affichées peuvent être inconsistantes, vu que les données dans /proc peuvent être plus récentes que le reste des informations d'accréditifs. Défaut : yes
- user** Dialogue avec le gestionnaire de service de l'utilisateur appelant, au lieu du gestionnaire de service système.
- system** Dialogue avec le gestionnaire de service système.
- H, -host=** Exécute l'opération à distance. Spécifie un nom d'hôte, ou un nom d'utilisateur est un nom d'hôte séparé par un @ pour s'y connecter. Le nom d'hôte peut optionnellement être suffixé par un nom de conteneur, séparé par un :, qui connecte directement au conteneur spécifique sur l'hôte spécifié. Il utilise ssh pour dialoguer avec une instance de gestionnaire de machine. Les noms de conteneur peuvent être énumérés avec machinectl **-H <host>**
- M, -machine=** Exécute l'opération dans un conteneur local. Spécifie un nom de conteneur
- no-pager** Ne sort pas la sortie dans un pager
- no-legend** N'affiche pas la légende (en-tête et pied de page)

## Commandes

**list** Affiche tous les paires sur le bus, par noms de service. Par défaut, affiche tous les noms.

**status [SERVICE]** Affiche les informations de processus et d'accréditifs d'un service, un processus, ou le propriétaire du bus.

**monitor [SERVICE...]** Dump les messages échangés. Si SERVICE est spécifié, affiche les messages vers et depuis ce pair. Sinon affiche tous les messages sur le bus.

**capture [SERVICE...]** Similaire à monitor mais écrit la sortie au format pcap.

**tree [SERVICE...]** Affiche une arborescence d'objets d'un ou plusieurs services. Si SERVICE est spécifié, affiche l'arborescence d'objets des services spécifiés, sinon affiche toutes les arborescences sur le bus.

**introspect SERVICE OBJECT [INTERFACE]** Affiche les interfaces, méthodes, propriétés et signaux de l'objet spécifié (identifié par son chemin) sur le service spécifié. Si l'argument interface est passé, la sortie est limitée aux membres de l'interface spécifiée.

**call SERVICE OBJECT INTERFACE METHOD [SIGNATURE [ARGUMENT...]]** Invoque une méthode et affiche la réponse. Prend un nom de service, un chemin d'objet, nom d'interface et nom de méthode. Si les paramètres doivent être passés à l'appel de la méthode, une chaîne signature est requise, suivie par les arguments, individuellement formatés comme chaîne. Pour les détails du format utilisé, voir plus bas. Pour supprimer la sortie de la donnée retournée, utiliser l'option `-quiet`.

**get-property SERVICE OBJECT INTERFACE PROPERTY...** Récupère la valeur courante d'une ou plusieurs propriétés d'objet. Plusieurs propriétés peuvent être spécifiées à la fois.

**set-property SERVICE OBJECT INTERFACE PROPERTY SIGNATURE ARGUMENT...** Définit la valeur courante d'une propriété d'objet.

## Format de paramètres

Les commandes `call` et `set-property` prennent une chaîne signature suivie par une liste de paramètres formatés comme chaîne (voir la spécification D-Bus pour les chaînes de signature). Pour les types simple, chaque paramètre suivant la signature devrait simplement être la valeur du paramètre formaté comme chaîne. Les valeurs booléennes peuvent être formatées en "true", "yes", "on", ou "1". Pour les tableaux, un argument numérique pour le nombre d'entrées doit être spécifié. Pour les variantes, la signature du contenu doit être spécifiée, suivie par le contenu. Pour les dictionnaires et structures, le contenu doit être spécifié directement.

```

formater une simple chaîne "jawoll"
s jawoll
formater un tableau de chaîne avec 2 entrées :
as 3 hello world foobar
formater un tableau dictionnaire qui map les chaînes en variantes, consistant de 3 entrées. La chaîne "One" est assignée à la chaîne "Eins".
"Two" à l'entier 32-bits 2, Yes est assigné comme booléen positif :
a{sv} 3 One s Eins Two u 2 Yes b true

```

## Exemples

Les 2 commandes suivantes écrivent d'abord une propriété puis la lisent. La propriété est trouvée dans l'objet `/org/freedesktop/systemd1` du service `org.freedesktop.systemd1`. Le nom de la propriété est "LogLevel" sur l'interface "org.freedesktop.systemd1.Manager".

```

busctl set-property org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager LogLevel s debug
busctl get-property org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager LogLevel s "debug"

```

Les 2 commandes suivantes lisent une propriété qui contient un tableau de chaînes, et affichent d'abord au format simple, puis au format verbeux

```

busctl get-property org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager Environment
as 2 "LANG=en_US.UTF-8" "PATH=/usr/local/sbin :/usr/local/bin :/usr/sbin :/usr/bin"
busctl get-property -verbose org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager Environment
ARRAY "s" {
STRING "LANG=en_US.UTF-8";
STRING "PATH=/usr/local/sbin :/usr/local/bin :/usr/sbin :/usr/bin";
};

```

La commande suivante invoque une méthode `StartUnit` sur l'interface `org.freedesktop.systemd1.Manager` de l'objet `org/freedesktop/systemd1` du service `org.freedesktop.systemd1`, et lui passe 2 chaînes  `cups.service` et `replace`.

```

busctl call org.freedesktop.systemd1 /org/freedesktop/systemd1 org.freedesktop.systemd1.Manager StartUnit ss "cups.service" "replace"

```

---

o `"/org/freedesktop/systemd1/job/42684"`