NAME

xdg-dbus-proxy − D−Bus proxy

SYNOPSIS

xdg−dbus−proxy [*OPTION*...] [*ADDRESS PATH* [*OPTION*...]...]

DESCRIPTION

xdg−dbus−proxy is a filtering proxy for D−Bus connections. Its arguments are one or more *ADDRESS−PATH* pairs specifying the buses to proxy, with options that specify what filtering to apply.

Basic Operation

The proxy listens to the unix domain socket at *PATH*, and for each client that connects to the socket, it opens up a new connection to the specified D−Bus *ADDRESS* (typically the session bus) and forwards data between the two. During the authentication phase all data is forwarded as received, and additionally for the first 1 byte zero we also send the proxy credentials to the bus.

Once the connection is authenticated there are two modes, filtered and unfiltered. In the unfiltered mode all messages are sent on as they are received. In the filtering mode policy is applied to determine which messages to allow, and which to drop.

Filtering is applied only to outgoing signals and method calls and incoming broadcast signals. All replies (errors or method returns) are allowed once for an outstanding method call, and never otherwise.

If a client ever receives a message from another peer on the bus, the senders unique name is made visible, so the client can track caller lifetimes via NameOwnerChanged signals. If a client calls a method on or receives a broadcast signal from a name (even if filtered to some subset of paths or interfaces), that names basic policy is considered to be (at least) TALK, from then on.

Policy

The policy for the filtering consists of a mapping from well−known names to a policy that is either SEE, TALK or OWN. The default initial policy is that the the user is only allowed to TALK to the bus itself (org.freedesktop.DBus, or no destination specified), and TALK to its own unique ID. All other clients are invisible.

Here is a description of the policy levels (each level implies the ones before it):

SEE
: The name/ID is visible in the ListNames reply

  The name/ID is visible in the ListActivatableNames reply

  You can call GetNameOwner on the name

  You can call NameHasOwner on the name

  You see NameOwnerChanged signals on the name

  You see NameOwnerChanged signals on the ID when the client disconnects

  You can call the GetXXX methods on the name/ID to get e.g. the peer pid

  You get AccessDenied rather than NameHasNoOwner when sending messages to the name/ID

TALK
: You can send any method calls and signals to the name/ID

  You will receive broadcast signals from the name/ID (if you have a match rule for them)

  You can call StartServiceByName on the name

OWN

You are allowed to call RequestName/ReleaseName/ListQueuedOwners on the name

Policy is specified with the **−−see**, **−−talk** and **−−own** options. The well−known names in these options can have a '.*' suffix. A name of "org.foo.*" matches "org.foo", org.foo.bar", and "org.foo.bar.gazonk", but not "org.foobar".

Polices are specified for well−known names, but they also affect the owner of that name, so that the policy for a unique ID is the union of the polices for all the names it owns. For technical reasons, the policy for a unique name is "sticky", in that the highest policy granted by a once−owned name is kept, even when the client releases that name. This is impossible to avoid in a race−free way in a proxy. But it is rarely a problem in practice, as few clients release names and stay on the bus.

In addition to the basic SEE/TALK/OWN policy, it is possible to specify more complicated rules about what method calls can be made on and what broadcast signals can be received from well−known names. A rule can restrict the allowed calls/signals to a specific object path or a subtree of object paths, and it can restrict the allowed interface down to an individual method or signal name.

Rules are specified with the **−−call** and **−−broadcast** options. The *RULE* in these options determines what interfaces, methods and object paths are allowed. It must be of the form [*METHOD*][*@PATH*], where *METHOD* can be either '*' or a D−Bus interface, possible with a '.*' suffix, or a fully−qualified method name, and *PATH* is a D−Bus object path, possible with a '/*' suffix.

OPTIONS

When options are used multiple times, the last option wins, unless otherwise specified.

General options:

−−help
Print help and exit

−−version
Print version

−−fd=FD
Write to FD when the proxies are ready, and stop when it is closed.

−−args=FD
Parse nul−separated arguments from the given file descriptor. This option can be used multiple times to parse options from multiple sources.

Proxy Options:

These options can only be used after an ADDRESS and apply to the proxy for that address.

−−filter
Enable filtering

−−log
Turn on logging

−−sloppy−names
Make all unique names visible.

−−see=*NAME*
Set the SEE policy for the given name.

−−talk=*NAME*
Set the TALK policy for the given name.

−−own=*NAME*
Set the OWN policy for the given name.

−−call=*NAME=RULE*
Set a rule for calls on the given name.

−−broadcast=*NAME=RULE*

Set a rule for broadcast signals from the given name.

EXAMPLES

$ xdg−dbus−proxy −−fd=26 unix:path=/run/usr/1000/bus
/run/usr/1000/.dbus−proxy/session−bus−proxy −−filter −−own=org.gnome.ghex.*
−−talk=ca.desrt.dconf −−call=org.freedesktop.portal.*=*
−−broadcast=org.freedesktop.portal.*=@/org/freedesktop/portal/*