

**NAME**

oddjobd.conf

**DESCRIPTION**

The **/etc/oddjobd.conf** configuration file specifies which services the *oddjobd* server provides over the D-Bus, and authorization rules which are enforced in addition to those enforced by the system message bus.

The configuration file is an XML document. The top-level element type is `<oddjobconfig>`, which contains one or more `<service>` elements. Each `<service>` describes a service which will be provided on the system-wide message bus.

Each `<object>` describes an object path which will be recognized by the specified service. The object path may include wildcards, in which case any call to an object with a path name which matches the specified path will be accepted. An object contains one or more `<interface>` elements, each of which describes a group of methods described in `<method>` elements.

Each `<method>` element must specify the method name as a value for its *name* attribute and may include a `<helper>` element which the name of an executable to run as its *exec* attribute and the number of arguments which will be passed to the helper as its *argument* attribute. A `<helper>` may also include attributes indicating whether or not the invoking user's name should be prepended to that argument list, and whether that argument list should be passed in to the helper via stdin (the default) or on its command line.

Each `<oddjobconfig>`, `<service>`, `<object>`, `<interface>`, or `<method>` element may also include authorization elements `<allow>` and `<deny>`. Each `<allow>` or `<deny>` rule specifies some combination of a user name and/or a UID range which the invoking user must match for the rule to apply. A rule can also specify the caller's SELinux context, user, role, or execution domain, and be applied or not based on whether or not policy is being enforced. All `<deny>` rules for the method are checked first, followed by all of its `<allow>` rules. If no matches are found, the `<deny>` rules for the containing `<interface>` element are checked, followed by its `<allow>` rules, and so on. If all ACLs are searched and no matches turn up, access is denied.

The *oddjobd* server will automatically supply information used by the D-Bus introspection mechanism on behalf of your objects, but only if the client which is requesting the information is allowed to invoke the *Introspect* method of the *org.freedesktop.DBus.Introspectable* interface provided by the object.

The configuration file may also indicate that the contents of other files should be read by the configuration parser, using an `<include>` element.

**EXAMPLES**

Here is an example file:

```
<?xml version="1.0"?>
<oddjobconfig/>
```

Another:

```
<?xml version="1.0"?>
<oddjobconfig>
  <allow user="wally"/>
  <service name="com.redhat.oddjob">
    <allow user="polly"/>
    <object name="/com/redhat/oddjob">
      <allow user="holly"/>
      <interface name="com.redhat.oddjob">
        <allow user="bob"/>
        <method name="pwd">
```

```

    <helper exec="/bin/pwd" argument_count="0" prepend_user_name="no"/>
    <allow user="jimmy"/>
    <allow user="billy"/>
    <allow min_uid="0" max_uid="1000"/>
  </method>
  <method name="reboot">
    <helper exec="/sbin/reboot" argument_count="0"/>
  </method>
</interface>
<interface name="org.freedesktop.DBus.Introspectable">
  <allow min_uid="0" max_uid="0"/>
</interface>
</object>
</service>
<include ignore_missing="yes">/etc/oddjobd-local.conf</include>
<include ignore_missing="yes">/etc/oddjobd.conf.d/*.conf</include>
</oddjobconfig>

```

And another:

```

<?xml version="1.0"?>
<oddjobconfig>
  <service name="com.example.management">
    <object name="/com/example/power">
      <interface name="com.example.shutdown">
        <method name="reboot">
          <allow user="root"/>
          <helper exec="/sbin/reboot" argument_count="0"/>
        </method>
      </interface>
      <interface name="org.freedesktop.DBus.Introspectable">
        <allow min_uid="0" max_uid="0"/>
      </interface>
    </object>
    <object name="/com/example/power">
      <interface name="com.example.shutdown">
        <method name="poweroff">
          <allow user="root"/>
          <helper exec="/sbin/poweroff" argument_count="0"/>
        </method>
      </interface>
      <interface name="org.freedesktop.DBus.Introspectable">
        <allow min_uid="0" max_uid="0"/>
      </interface>
    </object>
  </service>
</oddjobconfig>

```

## SEE ALSO

**oddjob\_request(1) oddjob.conf(5) oddjobd(8)**