

Custom Administrative Facets

An application can add and remove administrative facets using the `Communicator` operations shown below:

Slice

```
module Ice {
  local interface Communicator {
    // ...
    void addAdminFacet(Object servant, string facet);
    Object removeAdminFacet(string facet);
  };
};
```

The `addAdminFacet` operation installs a new facet with the given name, or raises `AlreadyRegisteredException` if a facet already exists with the same name. The `removeAdminFacet` operation removes (and returns) the facet with the given name, or raises `NotRegisteredException` if no matching facet is found.

The mechanism for [filtering administrative facets](#) also applies to application-defined facets. If you call `addAdminFacet` while a filter is in effect, and the name of your custom facet does not match the filter, the Ice run time will not expose your facet but instead keeps a reference to it so that a subsequent call to `removeAdminFacet` is possible.

We provide an example of using these communicator operations in our discussion of the [Process](#) facet.

See Also

- [Filtering Administrative Facets](#)
- [The Process Facet](#)