

Locator Semantics for Servers

A location service must know the endpoints of any [object adapter](#) whose identifier can be used in an indirect proxy. For example, suppose a client uses the following proxy:

```
Object1@PublicAdapter
```

The Ice run time in the client includes the identifier `PublicAdapter` in its [locate request](#) and expects to receive the associated endpoints. The only way the location service can know these endpoints is if it is given them. When you consider that an object adapter's endpoints may not specify fixed ports, and therefore the endpoint addresses may change each time the object adapter is activated, it is clear that the best source of endpoint information is the object adapter itself. As a result, an object adapter that is [properly configured](#) contacts the locator during activation to supply its identifier and current endpoints. More specifically, the object adapter registers itself with an object implementing the `Ice::LocatorRegistry` interface, whose proxy the object adapter obtains from the locator.

A location service may require that all object adapters be pre-registered via some implementation-specific mechanism. ([IceGrid](#) behaves this way by default.) This implies that activation can fail if the object adapter supplies an identifier that is unknown to the location service. In such a situation, the object adapter's `activate` operation raises `NotRegisteredException`.

In a similar manner, an object adapter that participates in a [replica group](#) includes the group's identifier in the locator request that is sent during activation. If the location service requires replica group members to be configured in advance, `activate` raises `NotRegisteredException` if the object adapter's identifier is not one of the group's [registered participants](#).

See Also

- [Object Adapters](#)
- [Locator Semantics for Clients](#)
- [Locator Configuration for a Server](#)
- [IceGrid](#)
- [Terminology](#)
- [Object Adapter Replication](#)