

Request Contexts



Methods on a proxy are overloaded with a trailing parameter representing a *request context*. The Slice definition of this parameter is as follows:

Slice

```
module Ice
{
    dictionary<string, string> Context;
}
```

As you can see, a context is a dictionary that maps strings to strings or, conceptually, a context is a collection of name-value pairs. The contents of this dictionary (if any) are implicitly marshaled with every request to the server, that is, if the client populates a context with a number of name-value pairs and uses that context for an invocation, the name-value pairs that are sent by the client are available to the server.

On the server side, the operation implementation can access the received `Context` via the `ctx` member of the `Ice::Current` parameter and extract the name-value pairs that were sent by the client.

Context names beginning with an underscore are reserved for use by Ice.

Topics

- [Explicit Request Contexts](#)
- [Per-Proxy Request Contexts](#)
- [Implicit Request Contexts](#)
- [Design Considerations for Request Contexts](#)

See Also

- [The Current Object](#)

