Objective-C Mapping for Local Interfaces

The Ice run time defines APIs using Slice. These APIs are provided as part of the Ice run time library and cannot be invoked remotely. (Doing so would not make any sense.) Therefore, the Slice interfaces for the Ice run time are defined as local interfaces. The Objective-C mapping for local interfaces differs from the default mapping in two ways:

- Local interfaces do not adopt an `<`*`interface-name`*`>Prx` protocol. (Doing so would be misleading because proxies imply that the target object can be remote.) Instead, the protocol for local interfaces has the same name as the interface. For example, the `Ice::Communicator` interfaces is defined as:

  Slice

  ```
  ["objc:prefix:ICE"]
  module Ice {
      local interface Communicator {
          // ...
      }
  };
  ```

  Because `Communicator` is a local interface, objects of type `ICECommunicator` are passed as `id<ICECommunicator>` (*not* `ICECommunicator*` or `id<ICEComunicatorPrx>`).

- Types that come in mutable and immutable variants (strings, sequences, and dictionaries) are always passed as the immutable variant. For example, the `getName` operation on the `ObjectAdapter` interface is defined as:

  Slice

  ```
  ["objc:prefix:ICE"]
  module Ice {
      local interface ObjectAdapter {
          string getName();
      };
  };
  ```

  Because `ObjectAdapter` is a local interface, the `getName` operation maps to:

  Objective-C

  ```
  -(NSString *) getName;
  ```

  Note that the returned string is of type `NSString` instead of `NSMutableString` (as would be the case for an operation on a non-local interface).
  For local interfaces, parameters are passed as the immutable version because their values are not meant to be modified by application code. In addition, passing the immutable version avoids an unnecessary data copy.

See Also

- Modules
- Local Types
- Objective-C Mapping for Operations