

Stream Helper Functions in C-Sharp

The stream classes provide all of the low-level functions necessary for [encoding and decoding](#) Ice types. However, it would be tedious and error-prone to manually encode complex Ice types such as classes, structs, and dictionaries using these low-level functions. For this reason, the [Slice compiler](#) optionally generates helper functions for streaming complex Ice types.

We will use the following Slice definitions to demonstrate the language mapping:

Slice

```
module M {
    sequence<...> Seq;
    dictionary<...> Dict;
    struct S {
        ...
    };
    enum E { ... };
    class C {
        ...
    };
};
```

The Slice compiler generates the corresponding helper functions shown below:

C#

```

namespace M
{
    public sealed class SeqHelper
    {
        public static int[] read(Ice.InputStream _in);
        public static void write(Ice.OutputStream _out, int[] _v);
    }

    public sealed class DictHelper
    {
        public static Dictionary<...> read(Ice.InputStream _in);
        public static void write(Ice.OutputStream _out, Dictionary<...> _v);
    }

    public sealed class SHelper
    {
        public static S read(Ice.InputStream _in);
        public static void write(Ice.OutputStream _out, S _v);
    }

    public sealed class EHelper
    {
        public static M.E read(Ice.InputStream _in);
        public static void write(Ice.OutputStream _out, M.E _v);
    }

    public sealed class CHelper
    {
        public CHelper(Ice.InputStream _in);
        public void read();
        public static void write(Ice.OutputStream _out, C _v);
        public M.C value
        {
            get;
        }
        // ...
    }

    public sealed class CPrxHelper : Ice.ObjectPrxHelperBase, CPrx
    {
        public static CPrx read(Ice.InputStream _in);
        public static void write(Ice.OutputStream _out, CPrx _v);
    }
}

```

In addition, the Slice compiler generates the following member functions for struct types:

C#

```

public struct S {
    ...
    public void ice_read(Ice.InputStream in);
    public void ice_write(Ice.OutputStream out);
}

```

Be aware that a call to `CHelper.read` does not result in the immediate extraction of an Ice object. The `value` property of the given `CHelper` object is updated when `readPendingObjects` is invoked on the [input stream](#).

See Also

- [slice2cs Command-Line Options](#)
- [The InputStream Interface in C-Sharp](#)