

Using Slice Checksums in C-Sharp

The Slice compilers can optionally generate [checksums](#) of Slice definitions. For `slice2cs`, the `--checksum` option causes the compiler to generate checksums in each C# source file that are added to a member of the `Ice.SliceChecksums` class:

C#

```
namespace Ice {
    public sealed class SliceChecksums {
        public readonly static System.Collections.Generic.Dictionary<string, string> checksums;
    }
}
```

The checksums map is initialized automatically prior to first use; no action is required by the application.

In order to verify a server's checksums, a client could simply compare the dictionaries using the `Equals` function. However, this is not feasible if it is possible that the server might be linked with more Slice definitions than the client. A more general solution is to iterate over the local checksums as demonstrated below:

C#

```
System.Collections.Generic.Dictionary<string,string> serverChecksums = ...
foreach(System.Collections.Generic.KeyValuePair<string, string> e in Ice.SliceChecksums.checksums) {
    string checksum;
    if (!serverChecksums.TryGetValue(e.Key, out checksum)) {
        // No match found for type id!
    } else if (!checksum.Equals(e.Value)) {
        // Checksum mismatch!
    }
}
```

In this example, the client first verifies that the server's dictionary contains an entry for each Slice type ID, and then it proceeds to compare the checksums.

See Also

- [Slice Checksums](#)