Using Slice Checksums in Python

The Slice compilers can optionally generate checksums of Slice definitions. For `slice2py`, the `--checksum` option causes the compiler to generate code that adds checksums to the dictionary `Ice.sliceChecksums`. The checksums are installed automatically when the Python code is first imported; no action is required by the application.

In order to verify a server's checksums, a client could simply compare the dictionaries using the comparison operator. However, this is not feasible if it is possible that the server might return a superset of the client's checksums. A more general solution is to iterate over the local checksums as demonstrated below:

Python

```
serverChecksums = ...
for i in Ice.sliceChecksums:
    if not serverChecksums.has_key(i):
        # No match found for type id!
    elif Ice.sliceChecksums[i] != serverChecksums[i]:
        # 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