

## NAME

Tie::SubstrHash - Fixed-table-size, fixed-key-length hashing

## SYNOPSIS

```
require Tie::SubstrHash;
```

```
tie %myhash, 'Tie::SubstrHash', $key_len, $value_len, $table_size;
```

## DESCRIPTION

The **Tie::SubstrHash** package provides a hash-table-like interface to an array of determinate size, with constant key size and record size.

Upon tying a new hash to this package, the developer must specify the size of the keys that will be used, the size of the value fields that the keys will index, and the size of the overall table (in terms of key-value pairs, not size in hard memory). *These values will not change for the duration of the tied hash.* The newly-allocated hash table may now have data stored and retrieved. Efforts to store more than `$table_size` elements will result in a fatal error, as will efforts to store a value not exactly `$value_len` characters in length, or reference through a key not exactly `$key_len` characters in length. While these constraints may seem excessive, the result is a hash table using much less internal memory than an equivalent freely-allocated hash table.

## CAVEATS

Because the current implementation uses the table and key sizes for the hashing algorithm, there is no means by which to dynamically change the value of any of the initialization parameters.

The hash does not support `exists()`.