We can manage a collection of key-value pairs

The key must be unique.

The key must not be changed.

The key must not be null.

Dictionary values are accessed by key.

Lists are marginally faster at iteration, Dictionaries are marginally faster at lookup.

var states = new Dictionary<string, string>()

Do:

Use dictionary to manage a generic collection by key.

Retrieve by key if you know it exists.

Use foreach to iterate through

Avoid:

Using a dictionary if there is no clear or unique key.

Using a dictionary if you don’t plan to lookup by key.

Retrieving a value without containsKey or TryGetValue.

Iterating through the elements. Do key or value instead.

You can use collection initialisers with Dictionaries. Syntax:

var states = new Dictionary<string, string>()

{

{key, value}

{key, value}

}

CollectionAssert can test equality of collections in unit tests.

To access a value in a dictionary we use the key with syntax:

var value = states[“CA”];

KeyNotFoundException – if the key doesn’t exist in the collection. To avoid this issue use the **ContainsKey()** method. Alternatively you can use the **TryGetValue()** which returns a bool for success and an out param for the value. This can be more efficient as it only performs one lookup.

We can iterate through Dictionaries; the key or value. Use a foreach.

System.Collections.Generic

* Dictionary<TKey, TValue>
* SortedList<TKey, TValue>
* SortedDictionary<TKey, TValue>

Use a dictionary whenever you plan to access a strongly typed collection of elements by key and sorting is not a requirement.

SortedList is a collection that is sorted on the key. Use if you need the keys in a sorted order. It uses less memory than a SortedDictionary and is quicker to populate if the data being entered is already sorted.

Use a SortedDictionary if you want sorting by key and the elements are unsorted.

System.Collections.ObjectModel

* ReadOnlyDictionary
* KeyedCollection

System.Collections.Specialized

* OrderedDictionary. Can be accessed by key or index.

System.Collections.Concurrent has thread safe dictionary classes.

FAQ

1. When is it appropriate to use a generic Dictionary?

When you want to store and access a collection by unique key.

1. What are the primary differences between a generic list and generic dictionary?

A list accesses elements by index, Dictionary by key.

1. What are the limitations of a dictionary key?

They must be unique, cannot be null and must not be changed.

1. What is the difference between foreach and for when iterating through a dictionary?

Foreach allows iteration by key, value or element. To use a for loop the developer needs to know the keys.