**Selecting a Collection Class**

Be sure to choose your collection class carefully. Using the wrong type can restrict your use of the collection.

Important

Avoid using the types in the [System.Collections](https://docs.microsoft.com/en-us/dotnet/api/system.collections) namespace. The generic and concurrent versions of the collections are recommended because of their greater type safety and other improvements.

Consider the following questions:

* Do you need a sequential list where the element is typically discarded after its value is retrieved?
  + If yes, consider using the [Queue](https://docs.microsoft.com/en-us/dotnet/api/system.collections.queue) class or the [Queue<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.queue-1) generic class if you need first-in, first-out (FIFO) behavior. Consider using the [Stack](https://docs.microsoft.com/en-us/dotnet/api/system.collections.stack) class or the [Stack<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.stack-1) generic class if you need last-in, first-out (LIFO) behavior. For safe access from multiple threads, use the concurrent versions, [ConcurrentQueue<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.concurrent.concurrentqueue-1) and [ConcurrentStack<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.concurrent.concurrentstack-1). For immutability, consider the immutable versions, [ImmutableQueue<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablequeue-1) and [ImmutableStack<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablestack-1).
  + If not, consider using the other collections.
* Do you need to access the elements in a certain order, such as FIFO, LIFO, or random?
  + The [Queue](https://docs.microsoft.com/en-us/dotnet/api/system.collections.queue) class, as well as the [Queue<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.queue-1), [ConcurrentQueue<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.concurrent.concurrentqueue-1), and [ImmutableQueue<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablequeue-1) generic classes all offer FIFO access. For more information, see [When to Use a Thread-Safe Collection](https://docs.microsoft.com/en-us/dotnet/standard/collections/thread-safe/when-to-use-a-thread-safe-collection).
  + The [Stack](https://docs.microsoft.com/en-us/dotnet/api/system.collections.stack) class, as well as the [Stack<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.stack-1), [ConcurrentStack<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.concurrent.concurrentstack-1), and [ImmutableStack<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablestack-1) generic classes all offer LIFO access. For more information, see [When to Use a Thread-Safe Collection](https://docs.microsoft.com/en-us/dotnet/standard/collections/thread-safe/when-to-use-a-thread-safe-collection).
  + The [LinkedList<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.linkedlist-1) generic class allows sequential access either from the head to the tail, or from the tail to the head.
* Do you need to access each element by index?
  + The [ArrayList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.arraylist) and [StringCollection](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.stringcollection) classes and the [List<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1) generic class offer access to their elements by the zero-based index of the element. For immutability, consider the immutable generic versions, [ImmutableArray<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablearray-1) and [ImmutableList<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablelist-1).
  + The [Hashtable](https://docs.microsoft.com/en-us/dotnet/api/system.collections.hashtable), [SortedList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.sortedlist), [ListDictionary](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.listdictionary), and [StringDictionary](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.stringdictionary) classes, and the [Dictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2) and [SortedDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sorteddictionary-2) generic classes offer access to their elements by the key of the element. Additionally, there are immutable versions of several corresponding types: [ImmutableHashSet<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablehashset-1), [ImmutableDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutabledictionary-2), [ImmutableSortedSet<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablesortedset-1), and [ImmutableSortedDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.immutablesorteddictionary-2).
  + The [NameObjectCollectionBase](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.nameobjectcollectionbase) and [NameValueCollection](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.namevaluecollection) classes, and the [KeyedCollection<TKey,TItem>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.objectmodel.keyedcollection-2) and [SortedList<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sortedlist-2) generic classes offer access to their elements by either the zero-based index or the key of the element.
* Will each element contain one value, a combination of one key and one value, or a combination of one key and multiple values?
  + One value: Use any of the collections based on the [IList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.ilist) interface or the [IList<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.ilist-1) generic interface. For an immutable option, consider the [IImmutableList<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.iimmutablelist-1) generic interface.
  + One key and one value: Use any of the collections based on the [IDictionary](https://docs.microsoft.com/en-us/dotnet/api/system.collections.idictionary) interface or the [IDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.idictionary-2) generic interface. For an immutable option, consider the [IImmutableSet<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.iimmutableset-1) or [IImmutableDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.immutable.iimmutabledictionary-2) generic interfaces.
  + One value with embedded key: Use the [KeyedCollection<TKey,TItem>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.objectmodel.keyedcollection-2) generic class.
  + One key and multiple values: Use the [NameValueCollection](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.namevaluecollection) class.
* Do you need to sort the elements differently from how they were entered?
  + The [Hashtable](https://docs.microsoft.com/en-us/dotnet/api/system.collections.hashtable) class sorts its elements by their hash codes.
  + The [SortedList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.sortedlist) class, and the [SortedList<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sortedlist-2) and [SortedDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sorteddictionary-2) generic classes sort their elements by the key. The sort order is based on the implementation of the [IComparer](https://docs.microsoft.com/en-us/dotnet/api/system.collections.icomparer) interface for the [SortedList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.sortedlist) class and on the implementation of the [IComparer<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.icomparer-1) generic interface for the [SortedList<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sortedlist-2) and [SortedDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sorteddictionary-2) generic classes. Of the two generic types, [SortedDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sorteddictionary-2) offers better performance than [SortedList<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sortedlist-2), while [SortedList<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sortedlist-2) consumes less memory.
  + [ArrayList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.arraylist) provides a [Sort](https://docs.microsoft.com/en-us/dotnet/api/system.collections.arraylist.sort) method that takes an [IComparer](https://docs.microsoft.com/en-us/dotnet/api/system.collections.icomparer) implementation as a parameter. Its generic counterpart, the [List<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1) generic class, provides a [Sort](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1.sort) method that takes an implementation of the [IComparer<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.icomparer-1) generic interface as a parameter.
* Do you need fast searches and retrieval of information?
  + [ListDictionary](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.listdictionary) is faster than [Hashtable](https://docs.microsoft.com/en-us/dotnet/api/system.collections.hashtable) for small collections (10 items or fewer). The [Dictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2) generic class provides faster lookup than the [SortedDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.sorteddictionary-2) generic class. The multi-threaded implementation is [ConcurrentDictionary<TKey,TValue>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.concurrent.concurrentdictionary-2). [ConcurrentBag<T>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.concurrent.concurrentbag-1) provides fast multi-threaded insertion for unordered data. For more information about both multi-threaded types, see [When to Use a Thread-Safe Collection](https://docs.microsoft.com/en-us/dotnet/standard/collections/thread-safe/when-to-use-a-thread-safe-collection).
* Do you need collections that accept only strings?
  + [StringCollection](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.stringcollection) (based on [IList](https://docs.microsoft.com/en-us/dotnet/api/system.collections.ilist)) and [StringDictionary](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized.stringdictionary) (based on [IDictionary](https://docs.microsoft.com/en-us/dotnet/api/system.collections.idictionary)) are in the [System.Collections.Specialized](https://docs.microsoft.com/en-us/dotnet/api/system.collections.specialized) namespace.
  + In addition, you can use any of the generic collection classes in the [System.Collections.Generic](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic) namespace as strongly typed string collections by specifying the [String](https://docs.microsoft.com/en-us/dotnet/api/system.string) class for their generic type arguments. For example, you can declare a variable to be of type [List<String>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1) or [Dictionary<String,String>](https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.dictionary-2).