# Chapter 2

Contax T
Sidewalk

# Collections Framework Overview

## Learning Objectives

- *List the four namespaces that constitute the .NET collections framework*
- *Understand the organization and content of the .NET collections framework*
- *List the members of each .NET collection framework namespace*
- *Navigate the .NET framework application programming interface (API) documentation*
- *List the non-generic collection classes and their corresponding generic replacements*

## Introduction

As elementary as the material in this chapter may appear upon initial consideration, it pays huge dividends to know your way around the .NET framework documentation. You will spend countless hours studying the documentation, no matter how many books you read, because that's where you'll find the most up to date information on the classes, structures, and other components of the .NET application programming interface (API).

In this chapter I show you how to decipher exactly what functionality a particular collection class provides by studying its inheritance hierarchy and list of implemented interfaces. I then highlight the major collection components found in the four namespaces of the .NET collections framework.

I also provide you with a helpful listing of the non-generic collection classes and their generic replacements.

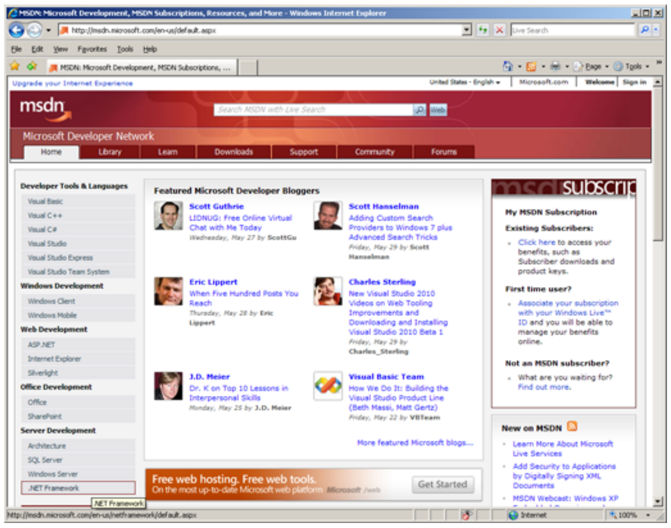
## The Microsoft Developer Network Documentation

The first step towards getting good help is knowing where to find it. The .NET API documentation hosted on the Microsoft Developer Network (MSDN) is the definitive source for the latest information regarding the .NET framework.

There are two ways of gaining access to the .NET framework documentation: 1. The straight forward way, which is to go directly to the MSDN website, follow the links to the docs, and then bookmark the link, or 2. The fastest way, which is to enter the name of the class or component you're looking for into Google. This, of course, assumes you know what you're looking for. If you take the time to explore the .NET framework API, you'll have a good idea of what to look for.

### The MSDN Website – www.msdn.com

The Microsoft Developer Network is the community portal for developers using Microsoft technology. In addition to many other areas of interest, it hosts the .NET framework API documentation. The URL to the site is http://www.msdn.com. Figure 2-1 shows the MSDN homepage.


MSDN: Microsoft Development, MSDN Subscriptions, Resources, and More - Windows Internet Explorer
http://msdn.microsoft.com/en-us/default.aspx
Live Search
File Edit View Favorites Tools Help
MSDN: Microsoft Development, MSDN Subscriptions, ...
Page Tools
Upgrade your Internet Experience
United States - English
Microsoft.com
Welcome
Sign in
msdn
Search MSDN with Live Search
Web
Microsoft Developer Network
Home
Library
Learn
Downloads
Support
Community
Forums
Developer Tools & Languages
Visual Basic
Visual C++
Visual C#
Visual Studio
Visual Studio Express
Visual Studio Team System
Windows Development
Windows Client
Windows Mobile
Web Development
ASP.NET
Internet Explorer
Silverlight
Office Development
Office
SharePoint
Server Development
Architecture
SQL Server
Windows Server
.NET Framework
.NET Framework
msdn subscrip
Free web hosting. Free web tools.
On the most up-to-date Microsoft web platform Microsoft /web
Get Started
http://msdn.microsoft.com/en-us/netframework/default.aspx
Internet
100%

Featured Microsoft Developer Bloggers

Scott Guthrie
LIDNUG: Free Online Virtual Chat with Me Today
Wednesday, May 27 by ScottGu

Scott Hanselman
Adding Custom Search Providers to Windows 7 plus Advanced Search Tricks
Friday, May 29 by Scott Hanselman

Eric Lippert
When Five Hundred Posts You Reach
Thursday, May 28 by Eric Lippert

Charles Sterling
New Visual Studio 2010 Videos on Web Tooling Improvements and Downloading and Installing Visual Studio 2010 Beta 1
Friday, May 29 by Charles_Sterling

J.D. Meier
Dr. K on Top 10 Lessons in Interpersonal Skills
Monday, May 25 by J.D. Meier

Visual Basic Team
How We Do It: Building the Visual Studio Product Line (Beth Massi, Matt Gertz)
Friday, May 22 by VBTeam

More featured Microsoft blogs...

My MSDN Subscription

Existing Subscribers:
- Click here to access your benefits, such as Subscriber downloads and product keys.

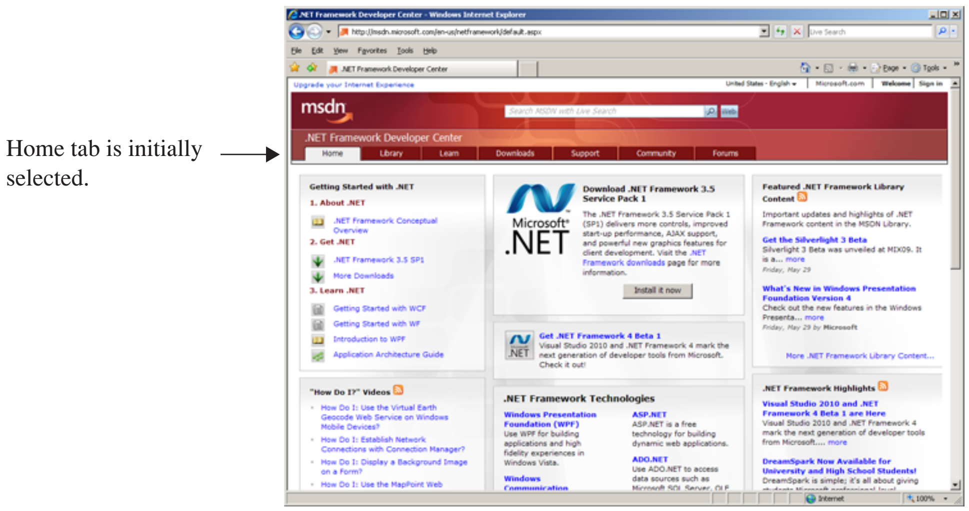
First time user?
- Associate your subscription with your Windows Live™ ID and you will be able to manage your benefits online.

Not an MSDN subscriber?
- What are you waiting for? Find out more.

New on MSDN
- Learn More About Microsoft Live Services
- Add Security to Applications by Digitally Signing XML Documents
- MSDN Webcast: Windows XP

Figure 2-1: Microsoft Developer Network (MSDN) Home Page

On the MSDN homepage locate the .NET Framework link in the lower left corner as figure 2-1 illustrates. Click the link. This takes you to the .NET Framework Developer Center page which is shown in figure 2-2.


Home tab is initially selected.
.NET Framework Developer Center - Windows Internet Explorer
http://msdn.microsoft.com/en-us/netframework/default.aspx
File Edit View Favorites Tools Help
.NET Framework Developer Center
Live Search
Page Tools
Upgrade your Internet Experience
United States - English | Microsoft.com | Welcome | Sign in
msdn
Search MSDN with Live Search
Web
.NET Framework Developer Center
Home
Library
Learn
Downloads
Support
Community
Forums
Install it now
Internet
100%

Getting Started with .NET

1. About .NET

.NET Framework Conceptual Overview

2. Get .NET

.NET Framework 3.5 SP1

More Downloads

3. Learn .NET

Getting Started with WCF

Getting Started with WF

Introduction to WPF

Application Architecture Guide

"How Do I?" Videos

- How Do I: Use the Virtual Earth Geocode Web Service on Windows Mobile Devices?
- How Do I: Establish Network Connections with Connection Manager?
- How Do I: Display a Background Image on a Form?
- How Do I: Use the MapPoint Web

Download .NET Framework 3.5 Service Pack 1

The .NET Framework 3.5 Service Pack 1 (SP1) delivers more controls, improved start-up performance, AJAX support, and powerful new graphics features for client development. Visit the .NET Framework downloads page for more information.

Get .NET Framework 4 Beta 1

Visual Studio 2010 and .NET Framework 4 mark the next generation of developer tools from Microsoft. Check it out!

.NET Framework Technologies

Windows Presentation Foundation (WPF)

Use WPF for building applications and high fidelity experiences in Windows Vista.

Windows Communication

ASP.NET

ASP.NET is a free technology for building dynamic web applications.

ADO.NET

Use ADO.NET to access data sources such as Microsoft SQL Server, OLE

Featured .NET Framework Library Content

Important updates and highlights of .NET Framework content in the MSDN Library.

Get the Silverlight 3 Beta

Silverlight 3 Beta was unveiled at MIX09. It is a... more

Friday, May 29

What's New in Windows Presentation Foundation Version 4

Check out the new features in the Windows Presenta... more

Friday, May 29 by Microsoft

More .NET Framework Library Content...

.NET Framework Highlights

Visual Studio 2010 and .NET Framework 4 Beta 1 are Here

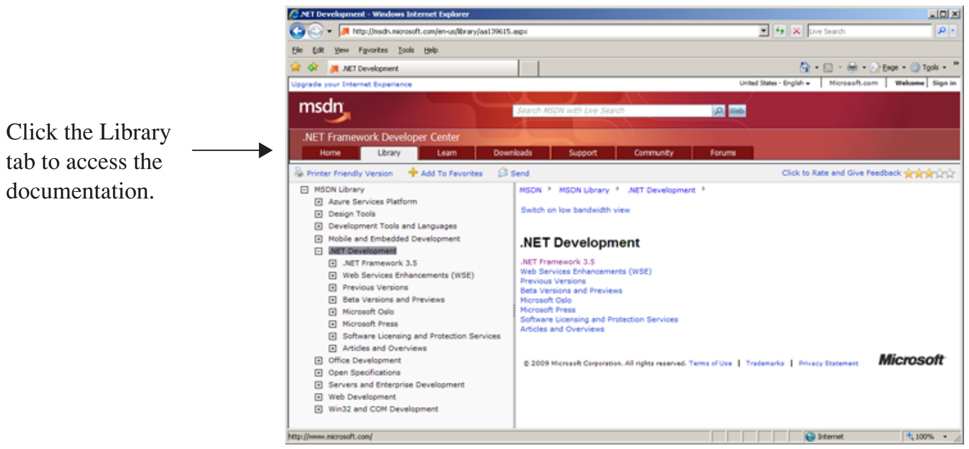
Visual Studio 2010 and .NET Framework 4 mark the next generation of developer tools from Microsoft.... more

DreamSpark Now Available for University and High School Students!

DreamSpark is simple; it's all about giving

Figure 2-2: .NET Developer Center Home Tab

Referring to figure 2-2 — the Home tab is initially selected when you arrive at the .NET Framework Developer Center page. Click the Library tab to access the documentation as is shown in figure 2-3.


Click the Library tab to access the documentation.
.NET Development - Windows Internet Explorer
http://msdn.microsoft.com/en-us/library/aa139615.aspx
File Edit View Favorites Tools Help
.NET Development
Live Search
Page Tools
Upgrade your Internet Experience
United States - English | Microsoft.com | Welcome | Sign in
msdn
Search MSDN with Live Search
Web
.NET Framework Developer Center
Home
Library
Learn
Downloads
Support
Community
Forums
Printer Friendly Version
Add To Favorites
Send
Click to Rate and Give Feedback
http://www.microsoft.com/
Internet
100%

MSDN Library
- Azure Services Platform
- Design Tools
- Development Tools and Languages
- Mobile and Embedded Development
- .NET Development
  - .NET Framework 3.5
  - Web Services Enhancements (WSE)
  - Previous Versions
  - Beta Versions and Previews
  - Microsoft Oslo
  - Microsoft Press
  - Software Licensing and Protection Services
  - Articles and Overviews
- Office Development
- Open Specifications
- Servers and Enterprise Development
- Web Development
- Win32 and COM Development

MSDN › MSDN Library › .NET Development ›

Switch on low bandwidth view

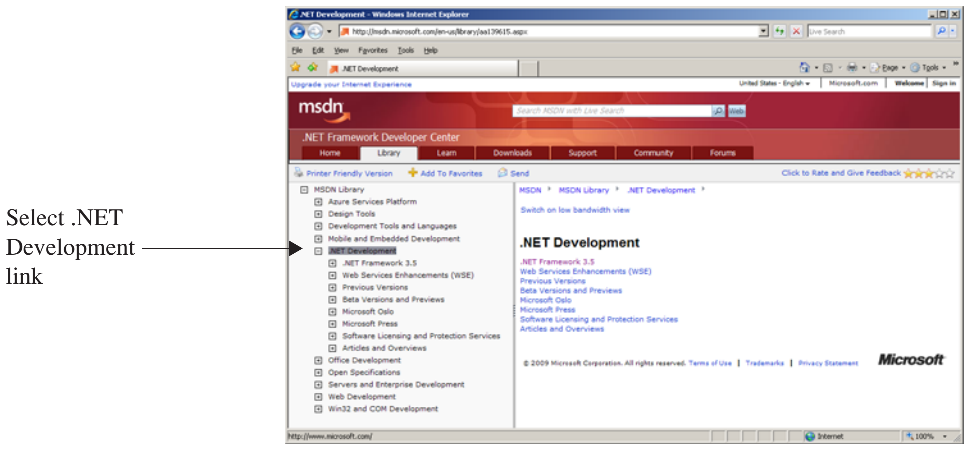
.NET Development

.NET Framework 3.5
Web Services Enhancements (WSE)
Previous Versions
Beta Versions and Previews
Microsoft Oslo
Microsoft Press
Software Licensing and Protection Services
Articles and Overviews

© 2009 Microsoft Corporation. All rights reserved. Terms of Use | Trademarks | Privacy Statement Microsoft

Figure 2-3: .NET Developer Center Library Tab

Next, locate the .NET Development link in the left frame as figure 2-4 illustrates.


Select .NET Development link
.NET Development - Windows Internet Explorer
http://msdn.microsoft.com/en-us/library/aa139615.aspx
File Edit View Favorites Tools Help
.NET Development
Live Search
Page Tools
Upgrade your Internet Experience
United States - English | Microsoft.com | Welcome | Sign in
msdn
Search MSDN with Live Search
Web
.NET Framework Developer Center
Home
Library
Learn
Downloads
Support
Community
Forums
Printer Friendly Version
Add To Favorites
Send
Click to Rate and Give Feedback
http://www.microsoft.com/
Internet
100%

MSDN Library
- Azure Services Platform
- Design Tools
- Development Tools and Languages
- Mobile and Embedded Development
- .NET Development
  - .NET Framework 3.5
  - Web Services Enhancements (WSE)
  - Previous Versions
  - Beta Versions and Previews
  - Microsoft Oslo
  - Microsoft Press
  - Software Licensing and Protection Services
  - Articles and Overviews
- Office Development
- Open Specifications
- Servers and Enterprise Development
- Web Development
- Win32 and COM Development

MSDN › MSDN Library › .NET Development ›

Switch on low bandwidth view

.NET Development

.NET Framework 3.5
Web Services Enhancements (WSE)
Previous Versions
Beta Versions and Previews
Microsoft Oslo
Microsoft Press
Software Licensing and Protection Services
Articles and Overviews

© 2009 Microsoft Corporation. All rights reserved. Terms of Use | Trademarks | Privacy Statement Microsoft

Figure 2-4: .NET Developer Link Selected

Underneath the .NET Development link, locate and expand the .NET Framework 3.5 link, then locate the .NET Framework Class Library link as is shown in figure 2-5.

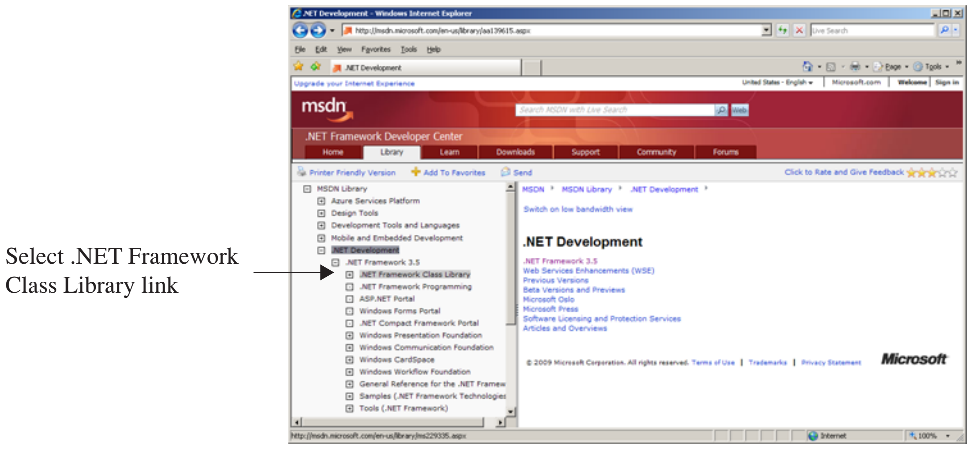
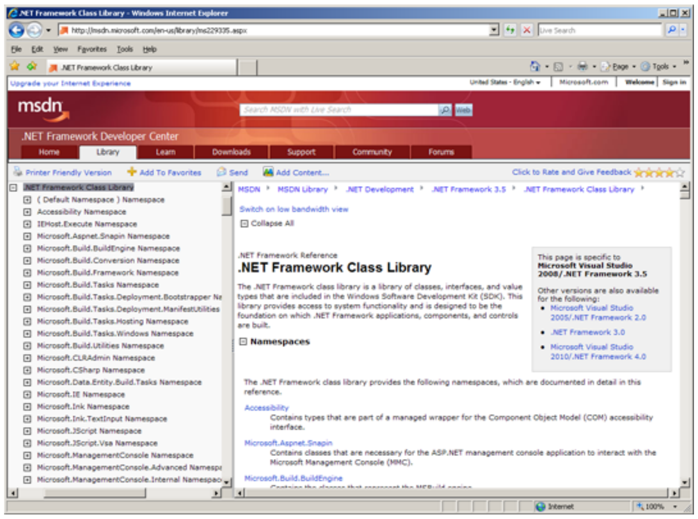

Select .NET Framework Class Library link
.NET Development - Windows Internet Explorer
msdn
.NET Framework Developer Center
Home
Library
Learn
Downloads
Support
Community
Forums
MSDN Library
Azure Services Platform
Design Tools
Development Tools and Languages
Mobile and Embedded Development
.NET Development
.NET Framework 3.5
.NET Framework Class Library
.NET Framework Programming
ASP.NET Portal
Windows Forms Portal
.NET Compact Framework Portal
Windows Presentation Foundation
Windows Communication Foundation
Windows CardSpace
Windows Workflow Foundation
.NET Development
.NET Framework 3.5
Web Services Enhancements (WSE)
Previous Versions
Beta Versions and Previews
Microsoft Oslo
Microsoft Press
Software Licensing and Protection Services
Articles and Overviews
Microsoft

Figure 2-5: .NET Framework Class Library Link

Click the .NET Framework Class Library link to expand its contents as shown in figure 2-6. A description of the .NET Framework Class Library appears in the right hand frame. Bookmark the site for future reference.


.NET Framework Class Library - Windows Internet Explorer
msdn
.NET Framework Developer Center
.NET Framework Class Library
{ Default Namespace } Namespace
Accessibility Namespace
IEHost.Execute Namespace
Microsoft.Aspnet.Snapin Namespace
Microsoft.Build.BuildEngine Namespace
Microsoft.Build.Conversion Namespace
Microsoft.Build.Framework Namespace
Microsoft.Build.Tasks Namespace
.NET Framework Reference
.NET Framework Class Library
Namespaces

The .NET Framework class library is a library of classes, interfaces, and value types that are included in the Windows Software Development Kit (SDK). This library provides access to system functionality and is designed to be the foundation on which .NET Framework applications, components, and controls are built.

This page is specific to Microsoft Visual Studio 2008/.NET Framework 3.5

Other versions are also available for the following:
- Microsoft Visual Studio 2005/.NET Framework 2.0
- .NET Framework 3.0
- Microsoft Visual Studio 2010/.NET Framework 4.0

The .NET Framework class library provides the following namespaces, which are documented in detail in this reference.

Accessibility
Contains types that are part of a managed wrapper for the Component Object Model (COM) accessibility interface.

Microsoft.Aspnet.Snapin
Contains classes that are necessary for the ASP.NET management console application to interact with the Microsoft Management Console (MMC).

Figure 2-6: .NET Framework Class Library Home

## Using Google to Quickly Locate Documentation

A faster way to access the .NET framework documentation is to search for a particular namespace or component in Google or your favorite web browser. This works best if you have some idea of what you're looking for. Figures 2-7 through 2-9 show how Google can be used to search for and quickly locate the System.Collections namespace.

## Where to Go From Here

Now that you know where to find .NET framework documentation, I recommend taking the time to explore the .NET framework class library docs to get a feel for what's there. Explore the *System*, *System.Collections*, and *System.Collections.Generic* namespaces. Get a good feel for the classes, structures, and other components located in each namespace. Study their methods and properties. At first this seems like a daunting task, but if you devote a little time each day to studying a small piece of the documentation, you'll quickly learn your way around.

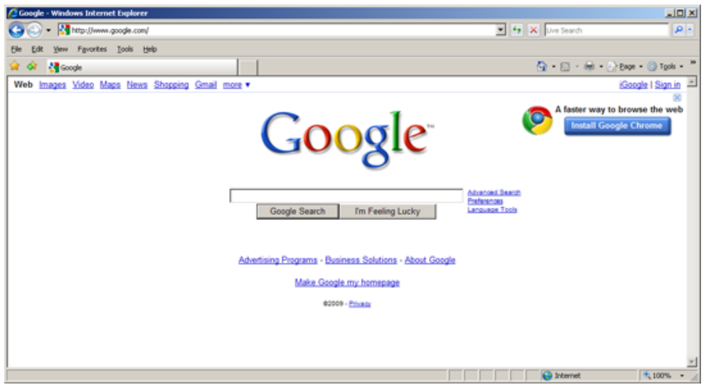
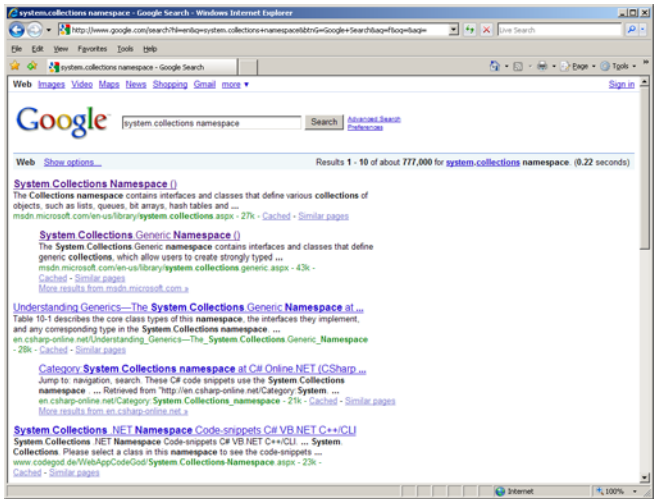


Figure 2-7: Google Home Page


system.collections namespace - Google Search - Windows Internet Explorer
Web Images Video Maps News Shopping Gmail more
Sign in
system.collections namespace
Search
Advanced Search
Preferences
Web Show options...

Results 1 - 10 of about 777,000 for system.collections namespace. (0.22 seconds)

System.Collections Namespace ()
The Collections namespace contains interfaces and classes that define various collections of objects, such as lists, queues, bit arrays, hash tables and ...
msdn.microsoft.com/en-us/library/system.collections.aspx - 27k - Cached - Similar pages

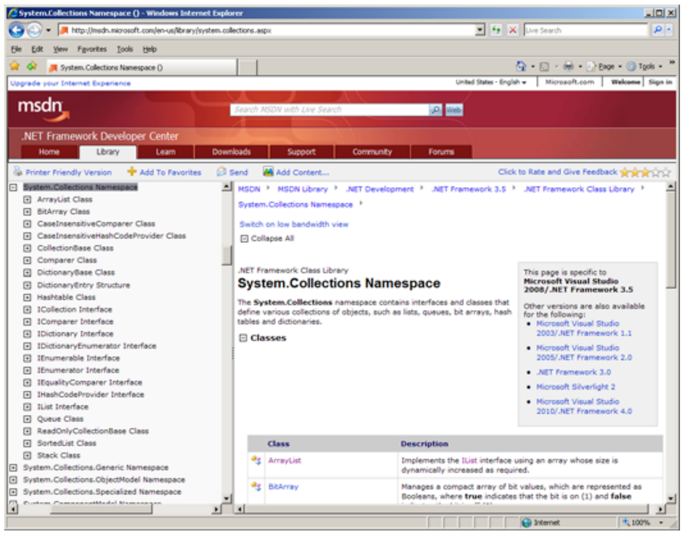
System.Collections.Generic Namespace ()
The System.Collections.Generic namespace contains interfaces and classes that define generic collections, which allow users to create strongly typed ...
msdn.microsoft.com/en-us/library/system.collections.generic.aspx - 43k - Cached - Similar pages
More results from msdn.microsoft.com »

Understanding Generics—The System.Collections.Generic Namespace at ...
Table 10-1 describes the core class types of this namespace, the interfaces they implement, and any corresponding type in the System.Collections namespace. ...
en.csharp-online.net/Understanding_Generics—The_System.Collections.Generic_Namespace - 28k - Cached - Similar pages

Category:System.Collections namespace at C# Online.NET (CSharp ...
Jump to: navigation, search. These C# code snippets use the System.Collections namespace . ... Retrieved from "http://en.csharp-online.net/Category:System. ...
en.csharp-online.net/Category:System.Collections_namespace - 21k - Cached - Similar pages
More results from en.csharp-online.net »

System.Collections .NET Namespace Code-snippets C# VB.NET C++/CLI
System.Collections .NET Namespace Code-snippets C# VB.NET C++/CLI. ... System. Collections. Please select a class in this namespace to see the code-snippets ....
www.codegod.de/WebAppCodeGod/System.Collections-Namespace.aspx - 23k - Cached - Similar pages

Figure 2-8: Search Results for System.Collections Namespace


System.Collections Namespace () - Windows Internet Explorer
http://msdn.microsoft.com/en-us/library/system.collections.aspx
Upgrade your Internet Experience
United States - English
Microsoft.com
Welcome
Sign in
msdn
Search MSDN with Live Search
.NET Framework Developer Center
Home
Library
Learn
Downloads
Support
Community
Forums
Printer Friendly Version
Add To Favorites
Send
Add Content...
Click to Rate and Give Feedback

System.Collections Namespace
ArrayList Class
BitArray Class
CaseInsensitiveComparer Class
CaseInsensitiveHashCodeProvider Class
CollectionBase Class
Comparer Class
DictionaryBase Class
DictionaryEntry Structure
Hashtable Class
ICollection Interface
IComparer Interface
IDictionary Interface
IDictionaryEnumerator Interface
IEnumerable Interface
IEnumerator Interface
IEqualityComparer Interface
IHashCodeProvider Interface
IList Interface
Queue Class
ReadOnlyCollectionBase Class
SortedList Class
Stack Class
System.Collections.Generic Namespace
System.Collections.ObjectModel Namespace
System.Collections.Specialized Namespace

MSDN › MSDN Library › .NET Development › .NET Framework 3.5 › .NET Framework Class Library › System.Collections Namespace ›

Switch on low bandwidth view

Collapse All

.NET Framework Class Library

System.Collections Namespace

The **System.Collections** namespace contains interfaces and classes that define various collections of objects, such as lists, queues, bit arrays, hash tables and dictionaries.

This page is specific to **Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:
- Microsoft Visual Studio 2003/.NET Framework 1.1
- Microsoft Visual Studio 2005/.NET Framework 2.0
- .NET Framework 3.0
- Microsoft Silverlight 2
- Microsoft Visual Studio 2010/.NET Framework 4.0

Classes

| Class | Description |
|---|---|
| ArrayList | Implements the IList interface using an array whose size is dynamically increased as required. |
| BitArray | Manages a compact array of bit values, which are represented as Booleans, where **true** indicates that the bit is on (1) and **false** |

Figure 2-9: System.Collections Namespace Documentation

### Quick Review

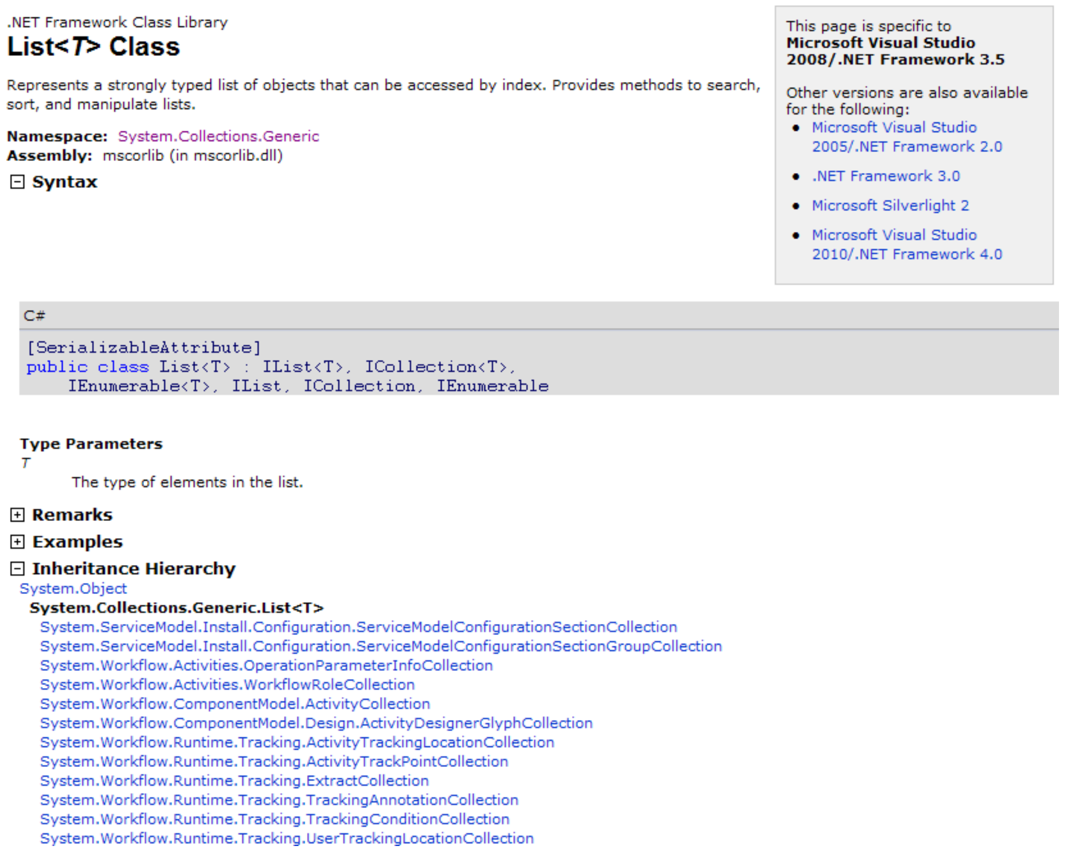
The Microsoft Developer Network (MSDN) contains the latest .NET framework API documentation. It's a good idea to explore the site and bookmark the .NET Framework Class Library page for future reference.

Use Google or your favorite web browser to quickly locate MSDN API documentation pertaining to individual .NET framework namespaces or components.

## Navigating an Inheritance Hierarchy

When you come across a particular class or structure in the .NET framework, chances are the functionality of that component is the result of inheriting from a base class (which itself may inherit from another base class), the implementation of multiple interfaces, or a combination of both. To better understand what, exactly, a component does, you must be able to trace its inheritance hierarchy and/or track down and study its implemented interfaces.

The best place to find this type of information is in the .NET framework documentation for the particular component you're interested in getting to know better. Consider for a moment the List<T> class. If you navigate to the List<T> documentation page on the MSDN website you'll find inheritance and implementation information in two sections: **Syntax** and **Inheritance Hierarchy** as is shown in figure 2-10.


.NET Framework Class Library

**List<*T*> Class**

Represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists.

**Namespace:** System.Collections.Generic
**Assembly:** mscorlib (in mscorlib.dll)

⊟ **Syntax**

This page is specific to
**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:
- Microsoft Visual Studio 2005/.NET Framework 2.0
- .NET Framework 3.0
- Microsoft Silverlight 2
- Microsoft Visual Studio 2010/.NET Framework 4.0

C#
```
[SerializableAttribute]
public class List<T> : IList<T>, ICollection<T>,
    IEnumerable<T>, IList, ICollection, IEnumerable
```

**Type Parameters**
*T*
The type of elements in the list.

⊞ **Remarks**
⊞ **Examples**
⊟ **Inheritance Hierarchy**
System.Object
**System.Collections.Generic.List<T>**
System.ServiceModel.Install.Configuration.ServiceModelConfigurationSectionCollection
System.ServiceModel.Install.Configuration.ServiceModelConfigurationSectionGroupCollection
System.Workflow.Activities.OperationParameterInfoCollection
System.Workflow.Activities.WorkflowRoleCollection
System.Workflow.ComponentModel.ActivityCollection
System.Workflow.ComponentModel.Design.ActivityDesignerGlyphCollection
System.Workflow.Runtime.Tracking.ActivityTrackingLocationCollection
System.Workflow.Runtime.Tracking.ActivityTrackPointCollection
System.Workflow.Runtime.Tracking.ExtractCollection
System.Workflow.Runtime.Tracking.TrackingAnnotationCollection
System.Workflow.Runtime.Tracking.TrackingConditionCollection
System.Workflow.Runtime.Tracking.UserTrackingLocationCollection

Figure 2-10: List<T> MSDN Documentation Page - Syntax and Inheritance Hierarchy Sections Expanded

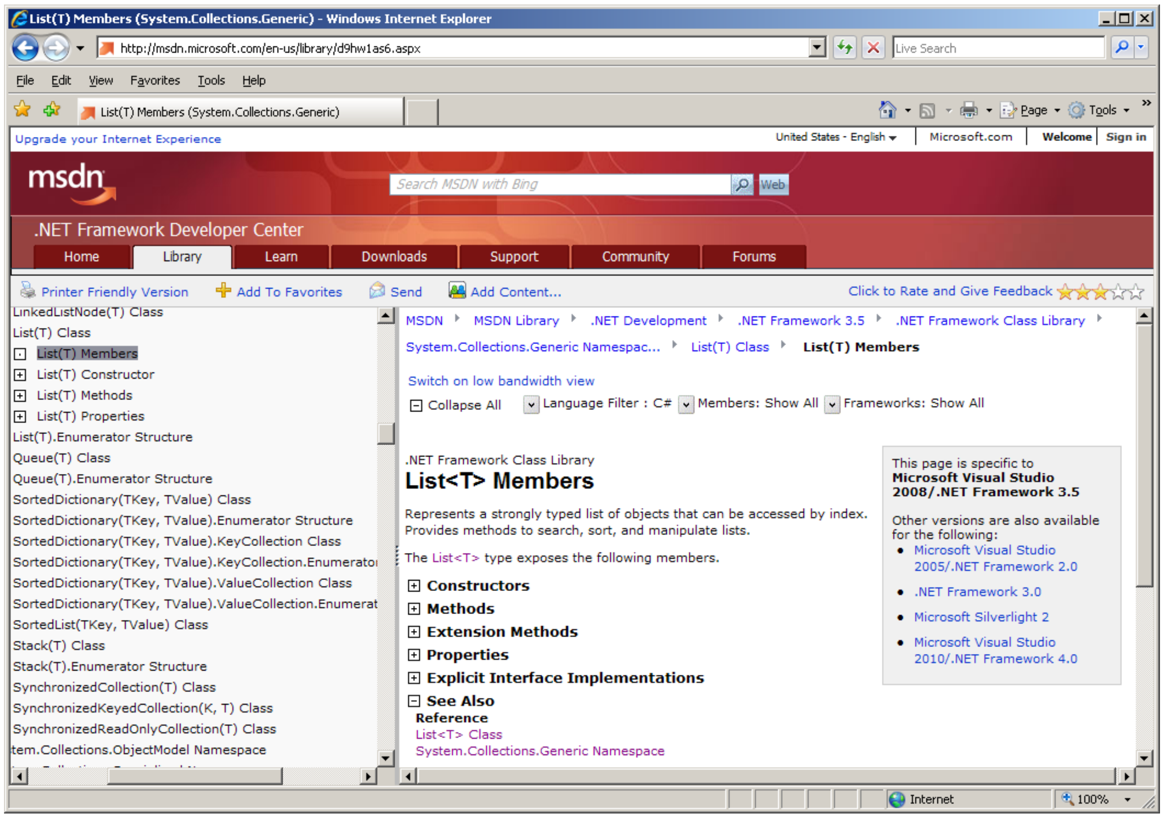
Referring to figure 2-10 — the **Syntax** section gives you the class declaration. The class declaration is important because it lists the class it extends, if any, and all its implemented interfaces. In the .NET framework an interface begins with the letter 'I'. So, by reading the List<T> class declaration you can tell at a glance it does not inherit from a class other than System.Object, and it implements a long list of interfaces.

The **Inheritance Hierarchy** section highlights the class you're currently visiting in black text. Above it in blue text will be a link to its immediate base class. In the case of the List<T> class, its immediate base class is System.Object. (Note: All classes and structures in the .NET framework ultimately and implicitly inherit from System.Object.) Below the class shown in black will be any subclasses derived from that class. As you can see from looking at figure 2-10, the List<T> class serves as the base class for many different specialized collection classes.

## Extension Methods

The .NET framework 3.0 introduced *extension methods*. An extension method is a special type of static method that enables a programmer to add a method to an existing class or structure without having to recompile the code associated with that component. An extension method, although static, is called as if it were an instance method.

The .NET API documentation lists the extension methods available to .NET framework classes and structures in the **Extension Methods** section. To see the list of extension methods for a particular class or structure, navigate to the component's Members page. Figure 2-11 shows the Members page for the List<T> generic collections class.


List(T) Members (System.Collections.Generic) - Windows Internet Explorer
http://msdn.microsoft.com/en-us/library/d9hw1as6.aspx
Live Search
File Edit View Favorites Tools Help
List(T) Members (System.Collections.Generic)
Page Tools
Upgrade your Internet Experience
United States - English
Microsoft.com
Welcome
Sign in
msdn
Search MSDN with Bing
Web
.NET Framework Developer Center
Home
Library
Learn
Downloads
Support
Community
Forums
Printer Friendly Version
Add To Favorites
Send
Add Content...
Click to Rate and Give Feedback

LinkedListNode(T) Class
List(T) Class
List(T) Members
List(T) Constructor
List(T) Methods
List(T) Properties
List(T).Enumerator Structure
Queue(T) Class
Queue(T).Enumerator Structure
SortedDictionary(TKey, TValue) Class
SortedDictionary(TKey, TValue).Enumerator Structure
SortedDictionary(TKey, TValue).KeyCollection Class
SortedDictionary(TKey, TValue).KeyCollection.Enumerato
SortedDictionary(TKey, TValue).ValueCollection Class
SortedDictionary(TKey, TValue).ValueCollection.Enumerat
SortedList(TKey, TValue) Class
Stack(T) Class
Stack(T).Enumerator Structure
SynchronizedCollection(T) Class
SynchronizedKeyedCollection(K, T) Class
SynchronizedReadOnlyCollection(T) Class
tem.Collections.ObjectModel Namespace

MSDN ▸ MSDN Library ▸ .NET Development ▸ .NET Framework 3.5 ▸ .NET Framework Class Library ▸ System.Collections.Generic Namespac... ▸ List(T) Class ▸ **List(T) Members**

Switch on low bandwidth view
Collapse All  Language Filter : C#  Members: Show All  Frameworks: Show All

.NET Framework Class Library
**List<T> Members**

Represents a strongly typed list of objects that can be accessed by index. Provides methods to search, sort, and manipulate lists.

The List<T> type exposes the following members.

**Constructors**
**Methods**
**Extension Methods**
**Properties**
**Explicit Interface Implementations**
**See Also**
**Reference**
List<T> Class
System.Collections.Generic Namespace

This page is specific to
**Microsoft Visual Studio 2008/.NET Framework 3.5**

Other versions are also available for the following:
- Microsoft Visual Studio 2005/.NET Framework 2.0
- .NET Framework 3.0
- Microsoft Silverlight 2
- Microsoft Visual Studio 2010/.NET Framework 4.0

Figure 2-11: Members Page for the List<T> Class

Referring to figure 2-11 — the Members page includes listings for all the constructors, methods, extension methods, properties, and explicit interface implementations. I have collapsed all the sections here to fit the page on the screen.

Constructor methods are defined by the class in question. The sections titled **Methods**, **Properties**, and **Explicit Interface Implementations** list methods and properties either inherited, defined by the class, or implemented by the class to fulfill a contract specified by an interface. The **Extension Methods** section, on the other hand, list methods that have been added to the class above and beyond those proscribed by inheritance. Most of the methods listed in the List<T>'s **Extension Methods** section are defined by the Enumerable class which is located in the System.Linq namespace.

### Quick Review

To get a feel for the functionality a particular class or structure provides, explore the component's inheritance hierarchy, along with its declaration as given in the **Syntax** section of the component's main documentation page. As of .NET framework version 3.0, the **Extension Methods** section lists methods that have been added to the component in addition to those defined by inheritance or interface implementation.

## The .NET Collections Framework Namespaces

To the uninitiated, the .NET collections API presents a bewildering assortment of interfaces, classes, and structures spread over four namespaces. In this section, I provide an overview of some of the things you'll find in each namespace. Afterward, I present a different type of organization that I believe you'll find more helpful.

One thing I will not do in this section is discuss every interface, class, or structure found in these namespaces. If I did that, you would fall asleep quick and kill yourself as your head slammed against the desk on its way down! Instead, I will only highlight the most important aspects of each namespace with an eye towards saving you time and frustration.

One maddening aspect of the .NET collections framework is in the way Microsoft chose to name their collection classes. For example, collection classes that contain the word List do not necessarily implement the IList or IList<T> interfaces. This means you can't substitute a LinkedList for an ArrayList without breaking your code.

In concert with this section you should explore the collections API and see for yourself what lies within each namespace.

### System.Collections

The System.Collections namespace contains non-generic versions of collection interfaces, classes, and structures. The contents of the System.Collections namespace represent the "old-school" way of collections programming. By this I mean that the collections defined here store only object references. You can insert any type of object into a collection like an ArrayList, Stack, etc., but, when you access an element in the collection and want to perform an operation on it specific to a particular type, you must first cast the object to a type that supports the operation. (By "performing an operation" I mean accessing an object member declared by its interface or class type.)

I recommend avoiding the System.Collections namespace altogether in favor of the generic versions of its members found in the System.Collections.Generic namespace. In most cases, you'll be trading cumbersome "old-school" style programming for more elegant code and improved performance offered by the newer collection classes.

### System.Collections.Generic

.NET 2.0 brought with it generics and the collection classes found in the System.Collections.Generic namespace. In addition to providing generic versions of the "old-school" collections contained in the System.Collections namespace, the System.Collections.Generic namespace added several new collection types, one of them being LinkedList<T>.

Several collection classes within the System.Collections.Generic namespace can be used off-the-shelf, so to speak, to store and manipulate strongly-typed collections of objects. These include List<T>, LinkedList<T>, Queue<T>, Stack<T>.

Other classes such as Dictionary<TKey, TValue>, SortedDictionary<TKey, TValue>, and SortedList<TKey, TValue> store objects (values) in the collection based on the hash values of keys. Special rules must be followed when implementing a key class. These rules specify the types of interfaces a key class must implement in order to perform equality comparisons. They also offer suggestions regarding the performance of hashing functions to optimize insertion and retrieval. You can find these specialized instructions in the **Remarks** section of a collection class's API documentation page.

## System.Collections.ObjectModel

The System.Collections.ObjectModel namespace contains classes that are meant to be used as the base classes for custom, user-defined collections. For example, if you want to create a specialized collection, you can extend the Collection<T> class. This namespace also includes the KeyedCollection<TKey, TItem>, ObservableCollection<T>, ReadOnlyCollection<T>, and ReadOnlyObservableCollection<T> classes.

The KeyedCollection<TKey, TItem> is an abstract class and is a cross between an IList and an IDictionary-based collection in that it is an indexed list of items. Each item in the list can also be accessed with an associated key. Collection elements are not key/value pairs as is the case in a Dictionary, rather, the element is the value and the key is extracted from the value upon insertion. The KeyedCollection<TKey, TItem> class must be extended and you must override its GetKeyForItem() method to properly extract keys from the items you insert into the collection.

The ObservableCollection<T> collection provides an event named CollectionChanged that you can register event handlers with to perform special processing when items are added or removed, or the collection is refreshed.

The ReadOnlyCollection<T> and ReadOnlyObservableCollection<T> classes implement read-only versions of the Collection<T> and ObservableCollection<T> classes.

## System.Collections.Specialized

As its name implies, the System.Collections.Specialized namespace contains interfaces, classes, and structures that help you manage specialized types of collections. Some of these include the BitVector32 structure, the ListDictionary, which is a Dictionary implemented as a singly linked list intended for storing ten items or less, StringCollection, which is a collection of strings, and StringDictionary, which is a Dictionary whose key/value pairs are strongly typed to strings rather than objects.

## Mapping Non-Generic To Generic Collections

In some cases, the System.Collection.Generic and System.Collections.ObjectModel namespaces provide a corresponding replacement for a collection class in the System.Collections namespace. But sometimes they do not. Table 14-1 lists the non-generic collection classes and their generic replacements, if any, and the underlying data structure implementation.

| Non-Generic | Generic | Underlying Data Structure |
|---|---|---|
| ArrayList | List<T> | Array |
| BitArray | *No generic equivalent* | Array |
| CollectionBase | Collection<T> | Array |
| DictionaryBase | KeyedCollection<TKey, TItem> | Hash Table & Array |
| HashTable | Dictionary<TKey, TValue> | Hash Table |
| Queue | Queue<T> | Array |
| ReadOnlyCollectionBase | ReadOnlyCollection<T> | Array |
| SortedList | SortedList<TKey, TValue> | Red-Black Tree |
| Stack | Stack<T> | Array |
| *No Non-Generic Equivalent* | LinkedList<T> | Doubly Linked List |
| *No Non-Generic Equivalent* | SortedDictionary<TKey, TValue> | Red-Black Tree |
| *No Non-Generic Equivalent* | SynchronizedCollection<T> † | Array |

Table 2-1: Mapping Non-Generic Collections to Their Generic Counterparts

| Non-Generic | Generic | Underlying Data Structure |
|---|---|---|
| *No Non-Generic Equivalent* | SynchonizedKeyedCollection<TKey, TItem> † | Hash Table & Array |
| *No Non-Generic Equivalent* | SynchronizedReadOnlyCollection<T> † | Array |
| *† Provides thread-safe operation* | | |

Table 2-1: Mapping Non-Generic Collections to Their Generic Counterparts

## Quick Review

"Old-school" style .NET collections classes store only object references and require casting when elements are retrieved. You should favor the use of generic collections as they offer strong element typing on insertion and retrieval and improved performance. The classes found in the System.Collections.ObjectModel namespace can serve as the basis for user-defined custom collections. The System.Collections.Specialized namespace contains classes and structures you will find helpful to manage unique collections.

## Summary

The Microsoft Developer Network (MSDN) contains the latest .NET framework API documentation. It's a good idea to explore the site and bookmark the .NET Framework Class Library page for future reference.

Use Google or your favorite web browser to quickly navigate to documents pertaining to individual .NET namespaces or components.

To get a feel for the functionality a particular class or structure provides, explore the component's inheritance hierarchy, along with its declaration as given in the **Syntax** section of the component's main documentation page. As of .NET framework version 3.0, the **Extension Methods** section lists methods that have been added to the component in addition to those defined by inheritance or interface implementation.

"Old-school" style .NET collections classes store only object references and require casting when elements are retrieved. You should favor the use of generic collections as they offer strong element typing on insertion and retrieval and improved performance. The classes found in the System.Collections.ObjectModel namespace can serve as the basis for user-defined custom collections. The System.Collections.Specialized namespace contains classes and structures you will find helpful to manage unique collections.

## References

.NET Framework 3.5 Reference Documentation, Microsoft Developer Network (MSDN) [www.msdn.com]

Rick Miller. *C# For Artists: The Art, Philosophy, and Science of Object-Oriented Programming*. ISBN-13: 978-1-932404-07-1. Pulp Free Press

## Notes