Cheatography

C# Naming Conventions Cheat Sheet

by Greg Finzer (GregFinzer) via cheatography.com/15280/cs/17465/

Casing Rules

Identifier	Case	Example
Namespace	Pascal	System.Drawing
Class	Pascal	Customer
Class Field	Camel	_transferConfig
Interface	Pascal	IPerson
Property	Pascal	TransferConfig
Method	Pascal	TransferAccount
Parameter	Camel	accountNumber
Constant	Pascal	InfiniteThrottle
Enumeration Type	Pascal	BrowserType
Enumeration Value	Pascal	InternetExplorer
Event	Pascal	TransferComplete
Exception Class	Pascal	TransferException
Pre-Processor	Upper	NETSTANDARD

Except for parameters, class fields, and pre-processor directives, all other identifiers follow a Pascal naming convention.

Namespaces

Choose names that indicate functionality

The general format for a namespace name is as follows:

<Company>.(<Project>)[.<Feature>][.<Subnamespace>]

Example: NewtonSoft.Json.Linq

Assemblies and DLL Names

Choose names that suggest large chunks of functionality.

It is advisable if assembly and DLL names follow the namespace names.

The following pattern may be followed for naming DLLs:

<Company>.<Component>.dll

Where *<component>* contains one or more dot separated clauses.

Example: NewtonSoft.Json.dll

Parameters

Choose parameter names that indicate what data is being affected.

Good: *firstName* - Uses camel casing and is descriptive

Bad: *decimalSalary* - Name should not be based on type

Resources

Nested identifiers with clear hierarchy

Example: *Menus.File.Close.Text*

Classes, Structs, and Interfaces

Use pascal cased nouns, noun phrases or adjective phrases like Customer or Invoice. This distinguishes type names from methods, which are named with verb phrases like SaveCustomer or LoadInvoice.

Use of suffixes and prefixes

Derived class should have suffix representing the base class. e,g OvalShape

TransferCompleteEventHandler – EventHandler suffix for handlers

TransferCompleteCallback – Callback suffix to delegates

TransferException – Exception suffix for deriving from Exception

AccountDictionary – Dictionary suffix for dictionary implementations

SocketStream - Stream suffix for inheriting from System.IO.Stream

Do use the prefix I for Interfaces. Example: ITransfer

General Naming Conventions

Do	Don't
Use easily readable identifier names. Favor readability over brevity.	Use underscore, hyphen or Hungarian notation.
Use semantically interesting generic names. eg. GetAmountDue vs GetDecimalValue	Use identifiers that conflict with C#
Use acronyms only if required, and only use widely accepted ones	Use abbreviations as part of the identifiers.

Types

Fields

Typically nouns or noun phrases are used as names for the fields. e.g. _salary

Properties

Nouns, noun phrases or adjectives are used for naming properties

Properties and Get methods should not be named alike.

Boolean properties should be named with phrases like Is or Has.

Methods

Typically verbs or verb phrases are used as names for the methods. e.g. GetEncodingString()

Events

Typically verbs or verb phrases are used as names for the events.

In event handlers, use two parameter named sender and e.

Concept of before and after should be given, e.g Closing, Closed, etc

By **Greg Finzer** (GregFinzer)
cheatography.com/gregfinzer/
www.kellermansoftware.com

Published 15th October, 2018.
Last updated 15th October, 2018.

Cheatography

C# Naming Conventions Cheat Sheet

by Greg Finzer (GregFinzer) via cheatography.com/15280/cs/17465/

Enums

Do not use prefixes or suffixes
Usually names are plural nouns. E.g Teams, Colors
Do not use flag as suffix for the names of flag enumerations

By **Greg Finzer** (GregFinzer)
cheatography.com/gregfinzer/
www.kellermansoftware.com

Published 15th October, 2018.
Last updated 15th October, 2018.