

Guidelines Quick Reference

.NET Guidelines

General Casing

Identifier	Case	Example
Class	Pascal	MyClass
Property	Pascal	BackColor
Private fields (class)	_Camel	_fieldName
Private fields (function)	Camel	fieldName
Method	Pascal	ToString
Parameter	Camel	firstName
Read-only static field	Pascal	RedValue
Interface	Pascal	IDisposable
Namespace	Pascal	System.Drawing
Enumeration type	Pascal	ErrorLevel
Enumeration values	Pascal	FatalError
Event	Pascal	ValueChanged
Exception class	Pascal	WebException

Naming

- Do choose easily readable identifier names.
- Do favor readability over brevity.
- Do not use underscores, hyphens, or any other nonalphanumeric characters.
- Do not use Hungarian notation.

Acronyms

- Do not use abbreviations or contractions as parts of identifier names.
 - Exception: Id(id) and Ok(ok) are allowed and should be camel cased.
- Do not use any acronyms that are not widely accepted.

- Do capitalize both characters of two-character acronyms, except the first word of a camel-cased identifier. Ex. DBConnection, dbConnection.
- Do capitalize only the first character of acronyms with three or more characters, except the first word of a camel-cased identifier. Ex. Xml, Sql, xml, sql.

Properties

- Do create read-only properties if the caller should not be able to change the value of the property.
- Do not provide set-only properties.
- Do provide sensible default values for all properties.
- Do allow properties to be set in any order.

Fields

- Do not provide instance fields that are public or protected.
- Do use constant fields for constants that will never change.

Style

- Always place curly braces on a new line
 - getter/setter are an exception
- Indents are always tabs (default = 4 spaces)
- Use Regions to group code blocks
- Do use constants for repeated values
- Always use curly braces for if, while, foreach, do, etc...
- 1 space around operators
- Do use code formatter
 - Do not format code AND make changes at the same time
- Do use utility classes for helper functions

Best Practices

- Do make business logic classes that contain the database calls and business logic.
- Do create Data Access Layer classes for excessive database calls.
- Do not comment obvious code
 - i.e. /*send It*/ email.Send();
- Do comment above line of code
- Do add block comments for all classes
- Do add block comments for complex functions
 - no need to comment the function "SendEmail()"
- Use comments to explain WHY choices were made when its not obvious
- Use comments to explain assumptions that are not obvious.
- No empty exception catches
- Do not comment out code - delete it!
- Do not check-in unused code

- Do use shelves for long running work
- Do use .cs files for code behind (aspx.cs, ascx.cs)
- Do name the solution and project files something appropriate. e.g. not web.sln or project.sln.

SQL Server

- Do not prefix table names, procedure names, view names with tbl_, sp_, view_
- Do not suffix table names, procedure names, view names with _tbl, _sp, _view
- Do use singular table names (i.e. Customer vs Customers)

General Javascript Casing

Identifier	Case	Example
Class	Pascal	MyClass
Property	Camel	backColor
Function	Camel	toString
Parameter	Camel	firstName
Constant	UpperCase_	SERVER_NAME
Namespace	Pascal	SilverTech.Utility

General Javascript Rules

- Always indent code with 4 spaces.
- No whitespace at the end of line or on blank lines.
- No filler spaces in empty constructs (e.g., {}, [], fn())
- New line at the end of each file.
- Strings always use double quotes (single quotes in string)

Naming

- Do choose easily readable identifier names.
- Do favor readability over brevity.
- Do not use underscores, hyphens, or any other non alphanumeric characters.
- Do not use Hungarian notation.

Curly Braces

Curly Braces start on same line, end on new line.

if/else/for/while/try always have braces and always go on multiple lines.

Examples:

```
function toCelsius(fahrenheit) {  
    return (5 / 9) * (fahrenheit - 32);  
}
```

```
for (i = 0; i < 5; i++) {  
    x += i;  
}
```

Chained Method Calls

When a chain of method calls is too long to fit on one line, there must be one call per line, with the first call on a separate line from the object the methods are called on. If the method changes the context, an extra level of indentation must be used.

```
$(".foo")  
  .addClass( "bar" )  
  .children()  
    .html( "Heya" )
```

Objects:

Declared on one line, new line for each property setter.
ending brace always on new line.

```
var person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 50,  
  eyeColor: "blue"  
};
```

General HTML\CSS Casing

Identifier	Case	Example
Class	hyphens	silvertch-button-active
Id	hyphens	body-container
Function	Camel	toJson
Parameter	Camel	firstName
Constant	UpperCase_	SERVER_NAME
Namespace	Pascal	SilverTech.Utility

Html Rules:

- Always declare the document type as the first line in your document
- Always use lowercase element names
- Always use lowercase attribute names
- Always close elements
- Always quote attribute values
- Do not make class = blue (eric will stab you)