

**Net**  
**Case 5.1**

**Proper-case Conversion and Data Transformation API User's Guide**

**The Software Company, Inc.**  
<https://SoftwareCompany.com>

**The Software Company**<sup>TM</sup>  
Software is our middle name

**NetCase for .NET** is a data quality component that converts existing data formats into proper cased formats that are more attractive and easier to read. NetCase applies proper case to your data through the use of special capitalization algorithms and Style Tables. Style Tables are easily-customizable conversion lists that allow you to apply different capitalization rules to the same word or phrase depending on its context. For example, using the predefined “NameStyle”, words like “co” can be converted to “Co.” or “Company” where “CityStateZipStyle” will convert “co” to “CO” or “Colorado”.

To use NetCase, set the Capitalization property to one of the six basic types: “Title”, “Sentence”, “EachWord”, “Upper”, “Lower” or “None”, then set the Style\_Table property to the name of the desired style table. Now, the same word can have different capitalization rules applied to it depending on the Capitalization and Style you’ve chosen.

A total of 16 different custom styles can be created and several predefined styles are included.

### Benefits

- **Save Keying Time** – automatically convert symbols and translate abbreviations into their correctly-capitalized full-spelling.
- **Instantly Correct Typos** – using the built-in list of common misspellings.
- **Precise Control** – 16 user-defined style tables; several pre-defined tables included.
- **Field Length Control** – create or expand abbreviations using the built-in USPS-approved abbreviation tables containing more than 2,300 entries.
- **Personalization** – over 1,000 proper-cased and punctuated Irish surnames are included.
- **Free upgrades** for a full year.

### Features

- Apply different capitalization rules to the same word depending on its context.
- Eliminate conflict between street address and city/state: Ct = Court, CT = Connecticut
- Correctly case surnames names like: “McDonald” and “O’Brien”
- Standardize name prefixes like: “M/M” and “M&M” to “Mr. & Mrs.”
- Correctly case medial caps such as: “IPHONE” to “iPhone” or “FEDEX” to “FedEx”
- Control field lengths by using the built-in table of USPS-approved abbreviations
- Royalty-free runtime
- Designed for use with all .NET-compatible programming languages

**NetCase for .NET** is controlled by user-defined style tables. Each customizable table can have its own unique definitions. NetCase starts by applying the algorithm for the selected capitalization. Next, every word / phrase from the input string is compared to the style table that was selected for that string. When a match is found, a replacement is made using the corresponding word / phrase from the selected style table overriding any other capitalization settings. The result is a precise transformation to proper case no matter what type of input string.

### Examples

The examples below use style tables that are predefined in the “NetCase.ref” file. These can be customized and new styles of your choice can be added.

**Capitalization** = “EachWord”  
**Style\_Table** = “NameStyle”  
**Text\_In** = “M/M JOHN OBRIEN PHD”  
**Text\_Out** = “Mr. and Mrs. John O’Brien, Ph. D.”

**Capitalization** = “EachWord”  
**Style\_Table** = “AddressStyle” (eliminate conflict between Street Suffix & CSZ)  
**Text\_In** = “1234 SE 41ST CT, APT 26E” (CT can be confused with Connecticut)  
**Text\_Out** = “1234 SE 41st Ct, Apt. 26E”

**Capitalization** = “EachWord”  
**Style\_Table** = “CityStateZipStyle” (avoid confusion with “company” & “Colorado”)  
**Text\_In** = “aspen, co 81611”  
**Text\_Out** = “Aspen, CO 81611”

### Text\_In

**Syntax:** Text\_In = String  
Text2\_In = String (optional)  
Text3\_In = String (optional)  
Text4\_In = String (optional)

**Description:**

Set these properties to the text strings to be processed.

When the “Convert” method is invoked, each “Text\_In” string is capitalized according to the “Capitalization” property setting and the selected style in the “Style\_Table” property then placed into the corresponding “Text\_Out” property.

---

### Capitalization

**Syntax:** Capitalization = StringLiteral  
Capitalization2 = StringLiteral (optional)  
Capitalization3 = StringLiteral (optional)  
Capitalization4 = StringLiteral (optional)

**Description:**

Set these properties to one of the following: “Title”, “Sentence”, “EachWord”, “Upper”, “Lower” or “None” to indicate your capitalization preference. **Default is “None”.**

### Capitalization Rules

**Title:** Capitalize first letter of each word except articles, coordinate conjunctions and prepositions of less than five letters. Capitalize each side of an apostrophe or ampersand as in: “De'Angelo” except plural possessive as in: “John's” and contractions such as “you're” or “I'm”. Lower case ordinal numbers: “1st”, “2nd”, etc.

**Sentence:** Capitalize first letter of each sentence

**EachWord:** Capitalize the first letter of each word. Capitalize each side of an apostrophe or ampersand as in: “De'Angelo” except plural possessive as in: “John's” and contractions such as “you're” or “I'm”. Lower case ordinal numbers: “1st”, “2nd”, etc.

**Upper:** Uppercase every letter of every word

**Lower:** Lowercase every letter of every word

**None:** Make only word / phrase replacements from the selected style table

### Style\_Table

**Syntax:** Style\_Table = "StringLiteral"  
Style\_Table2 = "StringLiteral" (optional)  
Style\_Table3 = "StringLiteral" (optional)  
Style\_Table4 = "StringLiteral" (optional)

**Description:**

Set these properties to the name of a previously defined "[StyleTable]" in the "NetCase.ref" file or "None" to indicate which style table to use for each corresponding "Text\_In" string. If set, this value must match the name of a previously defined style. **Default is "None"**.

*In every case, the selected style table definitions from the "NetCase.ref" file will override the capitalization algorithms. See "Updating User Control Tables" for information on customizing this file.*

---

### Convert\_Irish\_Surnames

**Syntax:** Convert\_Irish\_Surnames = Boolean (True/False)  
Convert\_Irish\_Surnames2 = Boolean (optional)  
Convert\_Irish\_Surnames3 = Boolean (optional)  
Convert\_Irish\_Surnames4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to convert Irish surnames for each corresponding "Text\_In" string. ("OHARA" or "O HARA" converts to "O'Hara", etc.) **Default is "False"**. *This flag is ignored when "Capitalization" property is set to "None"*.

---

### Typographic\_Apostrophe

**Syntax:** Typographic\_Apostrophe = Boolean (True/False)  
Typographic\_Apostrophe2 = Boolean (optional)  
Typographic\_Apostrophe3 = Boolean (optional)  
Typographic\_Apostrophe4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to use the typographic apostrophe (') in place of the standard apostrophe (') when converting Irish surnames. This property is only valid when "Convert\_Irish\_Surnames" property is set to "True". **Default is "False"**. *This flag is ignored when Capitalization property is set to "None"*.

### Alphanumeric\_Upper

**Syntax:** Alphanumeric\_Upper = Boolean (True/False)  
Alphanumeric\_Upper2 = Boolean (optional)  
Alphanumeric\_Upper3 = Boolean (optional)  
Alphanumeric\_Upper4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to convert alphanumeric words to all uppercase for each corresponding “Text\_In” string. (“3d” converts to “3D”, etc.) **Default is “False”.** *This flag is ignored when “Capitalization” property is set to “None”.*

---

### No\_Vowels\_Upper

**Syntax:** No\_Vowels\_Upper = Boolean (True/False)  
No\_Vowels\_Upper2 = Boolean (optional)  
No\_Vowels\_Upper3 = Boolean (optional)  
No\_Vowels\_Upper4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to convert alphabetic data containing no vowels to all uppercase for each corresponding “Text\_In” string. (“mlk” converts to “MLK”, etc.) **Default is “False”.** *This flag is ignored when “Capitalization” property is set to “None”.*

---

### Preserve\_Mixed\_Case

**Syntax:** Preserve\_Mixed\_Case = Boolean (True/False)  
Preserve\_Mixed\_Case2 = Boolean (optional)  
Preserve\_Mixed\_Case3 = Boolean (optional)  
Preserve\_Mixed\_Case4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to convert mixed-case text for each corresponding “Text\_In” string. When “Preserve\_Mixed\_Case” property is set to “True”, and mixed case text is detected in the input string, no text conversions take place and the original text is returned in the “Text\_Out” property. **Default is “False”.**

### **Create\_Abbreviations**

**Syntax:** Create\_Abbreviations = Boolean (True/False)  
Create\_Abbreviations2 = Boolean (optional)  
Create\_Abbreviations 3 = Boolean (optional)  
Create\_Abbreviations 4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to create business word abbreviations for each corresponding “Text\_In” string. Words are abbreviated from right to left until the target “Abbreviation\_Length” is reached or there are no more words to abbreviate. Set “Create\_Abbreviations” to “True” to compress common business words to their USPS-approved abbreviations. **Default is “False”**. See “*Abbreviation\_Length*” and “*Expand\_Abbreviations*” properties. *This flag is ignored when “Capitalization” property is set to “None”.*

---

### **Abbreviation\_Length**

**Syntax:** Abbreviation\_Length = Integer (0–1,024)  
Abbreviation\_Length2 = Integer (optional)  
Abbreviation\_Length3 = Integer (optional)  
Abbreviation\_Length4 = Integer (optional)

**Description:**

Set these properties to an integer in the range of 0–1,024 to indicate the extent of abbreviation you want. Set “Abbreviation\_Length” to zero to abbreviate every word in the “Text\_In” string. **Default is zero**. See “*Create\_Abbreviations*” property. *This property is ignored when “Create\_Abbreviations” property is set to “False”.*

---

### **Expand\_Abbreviations**

**Syntax:** Expand\_Abbreviations = Boolean (True/False)  
Expand\_Abbreviations2 = Boolean (optional)  
Expand\_Abbreviations 3 = Boolean (optional)  
Expand\_Abbreviations 4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to expand business word abbreviations for each corresponding “Text\_In” string. Set “Expand\_Abbreviations” to “True” to expand USPS-approved business word abbreviations to their full-spelling. **Default is “False”**. See “*Create\_Abbreviations*” property. *This flag is ignored when “Capitalization” property is set to “None”.*

### Remove\_Punctuation

**Syntax:** Remove\_Punctuation = "StringLiteral"  
Remove\_Punctuation 2 = "StringLiteral" (optional)  
Remove\_Punctuation 3 = "StringLiteral" (optional)  
Remove\_Punctuation 4 = "StringLiteral" (optional)

**Description:**

Set these properties to any string of characters you want removed from each corresponding Text\_In string. Each character in the Remove\_Punctuation string is compared to the Text\_In string and individually removed. Remove\_Punctuation is typically used to convert embedded, non-alphanumeric punctuation characters to spaces.

---

### Correct\_Common\_Misspell

**Syntax:** Correct\_Common\_Misspell = Boolean (True/False)  
Correct\_Common\_Misspell2 = Boolean (optional)  
Correct\_Common\_Misspell3 = Boolean (optional)  
Correct\_Common\_Misspell4 = Boolean (optional)

**Description:**

Set these properties to Boolean (True/False) to indicate whether or not to correct common misspellings for each corresponding "Text\_In" string. ("acn" is corrected to "can", etc.) **Default is "False"**. *This flag is ignored when "Capitalization" property is set to "None"*.

### Reference\_File\_Path

**Syntax:** Reference\_File\_Path = String

**Description:**

Set this property to the full path and file name of the user-defined file containing the style tables. Default tables are supplied and installed in the NetCase installation folder under the name: "NetCase.ref". You can rename and relocate this file to any other folder as long as you set the "Reference\_File\_Path" property to the full path and file name. **Default "Reference\_File\_Path" is first the folder of the invoking application:**

**"AppDomain.CurrentDomain.BaseDirectory" then the NetCase installation folder.**

*See "Updating User Control Tables" for information on customizing this file.*

---

### Static\_Key\_Name (licensed version)

**Syntax:** Static\_Key\_Name = String

**Description:**

Set this property to the name portion of the static key assignment or blank.

---

### Static\_Key (licensed version)

**Syntax:** Static\_Key = String

**Description:**

Set this property to the key portion of the static key assignment or blank.

**Text\_Out** (read only)

**Syntax:**      String = Text\_Out  
                 String = Text2\_Out (optional)  
                 String = Text3\_Out (optional)  
                 String = Text4\_Out (optional)

**Description:**

After invoking the “Convert” method, these properties are set to the converted text string of “Text\_In” after applying the “Style\_Table” and “Capitalization” property settings.

---

**Return\_Code** (read only)

**Syntax:**      String = Return\_Code

**Description:**

After invoking the “Convert” method, this property is set to blank upon successful completion. Most exceptions occur on the first invocation. The most common ones are listed below. *This property should be examined on each return from NetCase.*

**Common Return Codes:**

<b>C00</b>	Unrecognized capitalization setting ( <i>see “Capitalization” property</i> )
<b>R30</b>	Reference file open/read error
<b>R35</b>	Reference file not found ( <i>see “Reference_File_Path” property</i> )
<b>S00</b>	Undefined style – the selected style has not been defined in the “NetCase.ref” tables ( <i>see “Style_Table” property</i> )
<b>S01</b>	More than 16 defined styles ( <i>see “Updating User Control Tables”</i> )
<b>T00</b>	One of the style table definitions has reached capacity (4,096)
<b>L00</b>	Evaluation period expired
<b>L01</b>	Static key validation failed ( <i>see “Static_Key” property</i> )
<b>L50</b>	Evaluation license error

### Clear

**Syntax:** NetCase.Clear

**Description:**

When this method is invoked, all properties are cleared with the exception of “Static\_Key”, “Static\_Key\_Name” and “Reference\_File\_Path”.

---

### Convert

**Syntax:** NetCase.Convert

**Description:**

When this method is invoked, each “Text\_In” property is inspected and the capitalization converted according to the “Capitalization” property and the selected “Style\_Table”. The transformed data is then placed into the corresponding “Text\_Out” property. The “Return\_Code” property is also set and should be checked after each invocation of the “Convert” method. *See “Return\_Code” property.*

### Updating User Control Tables

**NetCase.ref** is a file containing the proper case control tables. It is located by default in the NetCase installation folder. Use Notepad or a similar text editor to edit the contents. Detailed information on the format of the entries is contained within the file. This file can also be relocated. See “*Reference\_File\_Path* property”.

The use of the wildcard character (\*) allows you to capitalize groups of words depending on their prefix. If you need to exclude certain words from the wildcarding, place them in the table before the wildcarded word. The wildcard character may appear following an entry in the first column or both columns.

**Extensive tables are included. Below are a few examples:**

NetCase style table entries . . .

<u>Common</u>	<u>Replacement</u>
DIGENOVA	diGenova
DIGITAL	digital
DIG*	DiG*

After NetCase conversion . . .

<u>DIGENOVA</u> <b>adjusted to</b>	diGenova
<u>DIGITAL</u>	digital
<u>DIGIORNO</u>	DiGiorno

All words prefixed by “DIG” are replaced with the prefix “DiG” with the exception of the words “DIGENOVA” and “DIGITAL” since they appear in the Style Table *before* “DIG\*”.

### Deploying Your Applications

Be sure to include the following in your deployment package:

**NetCase.dll** – usually placed in the application folder or Global Assembly Cache (GAC)

**NetCase.ref** – usually placed in the application folder\*

\* “NetCase.ref” reference file can be placed anywhere on the target machine as long as the full path to it is specified in the “Reference\_File\_Path” property.

---

In addition to the above, there is a common runtime that can be placed in the application folder or the Global Assembly Cache (GAC) of the target machine.

**Fujitsu.COBOL.dll**

---

### Evaluation License

The evaluation license is valid for a period of 7 days or up to 1,000 calls.

[Sales@SoftwareCompany.com](mailto:Sales@SoftwareCompany.com)

[Support@SoftwareCompany.com](mailto:Support@SoftwareCompany.com)