

Net
Gender Microsoft for .net
3.5

Name Parsing & Gender Determination Component

The Software Company, Inc.
www.SoftwareCompany.com

The Software Software is our middle name
Company™

NetGender for .NET allows you to quickly and easily build name verification, parsing and gender determination into your custom applications. Accurately verify whether or not a particular field contains a valid individual or company name. And, for a more appealing presentation, let NetGender set the proper capitalization.

Using the parse feature, you can accept names free-form letting NetGender automatically split each name into standard components: Prefix, First, Middle, Last and Suffix no matter what the original format. Next, using a 100,000+ name look-up table in combination with an 8,000+ word company table, the gender is determined with unmatched precision. Easily updated tables control the entire process.

NetGender can process all styles of names including inverse, natural order, hyphenated and multi-part last names. Multiple names in the same field and companies can be easily separated providing you with powerful formatting control.

With the help of the “Nicknames” property, you can easily identify obscure duplicate names such as “Bill”, “Billy”, “Will”, “Willy” and “William”.

As a bonus, when you combine NetGender with our NetAddress product, you can handle even the impossible task of identifying data that has been entered free-form, where the names, addresses and C/S/Z “float” from field to field. You’ll always be certain of what data you’re working with.

Features

- 100,000 Name Lookup Table for Pinpoint Accuracy
- Quickly Identify Incomplete or Incorrect Names
- Automatic Name Style Identification – Standardize Lists of Various Formats
- Nicknames Property - Invaluable for Finding Duplicate Names
- User Updatable Tables So Your Applications Never Go Out-of-Date
- Proper Case Conversion for More Attractive Data Presentation
- No Recurring Update Charges
- Free Upgrades for a Full Year
- Royalty Free Run-Time (first 100 end-user installations)
- Small Run-Time Package
- Designed for use with all .NET Compatible Programming Languages
- Bullet-Proof and *FAST*

NetGender for .NET starts by meticulously identifying each individual name element based on the `Name_Style` you've selected. Intuitive algorithms examine the results and a selection is made of the most complete and correct data. Next, the name elements are split according to `Name_Style` and the user-specified prefix/suffix abbreviations are applied. If not already coded by the gender override table, the gender is now determined according to the 100,000+ name system gender table. The `Name_Quality` flag is then set to indicate how complete and correct the name is. Finally, the standardized name components are returned to your application along with a complete and cleansed composite name.

NetGender is the ONLY name verification and parsing software that can reliably extract a name when the format of the name is variable or unknown.

Examples

Name_In: MR DE LA REY, JOHN R JUNIOR OR JANE

Name_Out: (M) Mr John R De La Rey Jr

Name2_Out: (F) Jane De La Rey

Name_In: SMITH-DE LA REY JR, JOHN A & JANE B, PHD

Name_Out: (M) John A Smith-De La Rey Jr

Name2_Out: (F) Jane B Smith-De La Rey PhD

Name_In: JANE SMITH-JONES C/O THE SOFTWARE COMPANY

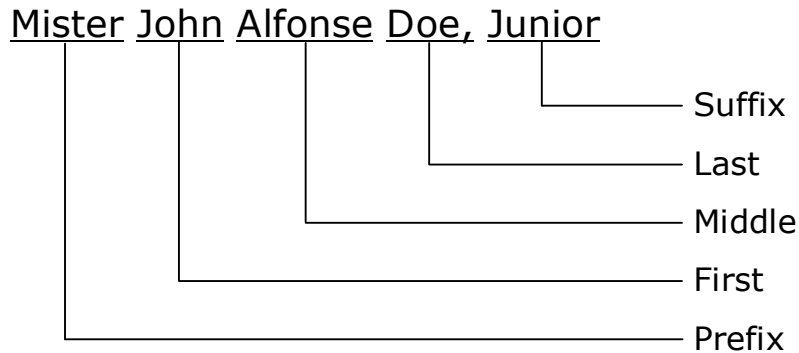
Name_Out: (F) Jane Smith-Jones

Name2_Out: (C) The Software Company

Name_In: O'LEARY-DE LA REY, DALE, OFFICE MANAGER

Name_Out: (N) Dale O'Leary-De La Rey Ofc Mgr

Name Elements



A Word About Gender Coding

Not all gender identification systems are created the same. Most use a simple table of about 8,000 to 18,000 first names. NetGender uses an extensive 100,000+ name lookup table that was specially created with a rich ethnic diversity.

NetGender also uses a unique gender percentage factor. This factor is based on the proportion of males to females for a particular name. This allows the programmer to set the point at which certain names will be returned as a neutral gender. See "Gender_Confidence" property.

NetGender then applies the pre-programmed rules from an 8,000+ word Company Table to yield astonishing accuracy.

This unique and powerful strategy produces results unmatched by any other software.

Name_In

Syntax: Name_In = String

Description:

Set this property to the name string to be processed.

When the “Parse” method is invoked, the Name_In string is standardized and formatted then placed into the Name_Out property. In addition, each element of the Name_In string is placed into the corresponding name component property.

Name_Style (changed in v3.0+)

Syntax: Name_Style = StringLiteral (“FML”, “FML&FM”, etc.)

Description:

Set this property to the style that most closely matches the format of the Name_In string. In the examples below, F=First, M=Middle, L=Last. **Default is “FML”.**

| | | |
|-------------|-------------------------------|---|
| FML | John A Doe | Names are in Natural Order |
| FML&FM | John A Doe & Jane A | Never a Last Name in Name 2 |
| FML&FM(L) | John A Doe & Jane A (Jones) | Sometimes a Last Name in Name 2 |
| FML&FML | John A Doe & Jane A Jones | Always a Last Name in Name 1 & 2 |
| FM(L)&FML | John A (Doe) & Jane A Jones | Sometimes a Last Name in Name 1 |
| FM(L)&FM(L) | John A (Doe) & Jane A (Jones) | Sometimes a Last Name in Name 1 or 2 |
| FM&FML | John A & Jane A Doe | Never a Last Name in Name 1 |
| LFM | Doe, John A | Names are in Inverse Order |
| LFM&FM | Doe, John A & Jane A | Never a Last Name in Name 2 |
| LFM&FM(L) | Doe, John A & Jane A (Jones) | Sometimes a Last Name in Name 2 |
| LFM&(L)FM | Doe, John A & (Jones)Jane A | Sometimes a Last Name in Name 2 |
| LFM&LFM | Doe, John A & Jones Jane A | Always a Last Name in Name 1 & 2 |
| VARIABLE | Inconsistent & Unknown | Names are mixed Natural & Inverse |
| SPLIT | Anything & Everything | Split at Name Connector only |
| COMPANY | Sanford & Son | Forces a company name check without splitting at the name connector |

The input names can be separated by various delimiters such as: “c/o”, “&”, “dba”, “and”, “attn”, etc. The “&” above, generically represents these delimiters. The delimiters are user-defined. *See “Updating User Control Tables” later in this guide.* Parenthesis “()” indicate that this element may or may not be present in all of the input name strings. If two individuals or a company and an individual co-exist in the input name string, you should choose one of the multi-name styles such as: FML&FML, FML&FM, etc. for best results.

The closer the style matches your data the better the results. Use “Variable” when the name format is inconsistent or unknown.

Name Style Examples

- FML** **Use when there is only one name in the input string.**
Name is split into First, Middle, Last:
 Name_In: JOHN A DOE
 Name_Out: John A Doe
- FML&FM** **Use when last name is NEVER present in name two.**
Name two ALWAYS inherits its last name from Name one:
 Name_In: JANE A DOE AND JOHN A
 Name_Out: Jane A Doe
 Name2_Out: John A Doe
- FML&FM(L)** **Use when last name is SOMETIMES present in name two.**
Name two keeps its own last name if present:
 Name_In: JOHN A DOE AND JANE A JONES
 Name_Out: John A Doe
 Name2_Out: Jane A Jones
Name two inherits its last name from Name one:
 Name_In: JOHN A DOE AND JANE A
 Name_Out: John A Doe
 Name2_Out: Jane A Doe
- FML&FML** **Use when a full name is ALWAYS present in name one & two.**
Name one & two ALWAYS keep their own last names:
 Name_In: JOHN DOE DBA THE SOFTWARE COMPANY
 Name_Out: John Doe
 Name2_Out: The Software Company
- FM(L)&FML** **Use when last name is SOMETIMES present in name one.**
Name one keeps its own last name if present:
 Name_In: JOHN A DOE AND JANE A JONES
 Name_Out: John A Doe
 Name2_Out: Jane A Jones
Name one inherits its last name from Name two:
 Name_In: JOHN A DOE AND JANE A
 Name_Out: John A Doe
 Name2_Out: Jane A Doe

Name Style Examples

FM(L)&FM(L) Use when last name is **SOMETIMES** present in name one or two.

Name one and two keep their own last names if present:

Name_In: JOHN A DOE AND JANE A JONES

Name_Out: John A Doe

Name2_Out: Jane A Jones

Name one inherits its last name from Name two:

Name_In: JOHN A AND JANE A JONES

Name_Out: John A Jones

Name2_Out: Jane A Jones

FM&FML Use when last name is **NEVER** present in name one.

Name one **ALWAYS** inherits its last name from Name two:

Name_In: JANE A AND JOHN DOE

Name_Out: Jane A Doe

Name2_Out: John Doe

VARIABLE Use when the format of the name(s) is **inconsistent or unknown**.

Style is automatically determined based on the format, punctuation and gender lookup results. Name one is FML, Name two is LFM:

Name_In: JOHN A DOE & JONES, JANE A

Name_Out: John A Doe

Name2_Out: Jane A Jones

Also, see Variable_Default property

SPLIT Use when you want to split name one & two at the [NameConnector]

Filters are removed, no other formatting takes place.

[NameConnector] of “c/o” was specified in the “NetGender.ref” file:

Name_In: DOE JOHN A c/o THE SOFTWARE COMPANY

Name_Out: Doe John A

Name2_Out: The Software Company

Use “COMPANY” style to avoid conflict with company names that also contain Name Connectors. When [NameConnectors] such as “&” are specified in the user-defined “NetGender.ref” file, certain company names such as Dewey Cheatum & Howell can sometimes be confused with a compound individual name such as John & Mary Smith. If this is the case, use the COMPANY style first to positively identify the field as a company.

If an exact match on a company name is found in the “NetGender.ref” file or certain company keywords such as “ABC” are found in the system gender table, the Gender property is returned as “C” (company) and the Name_Quality property is returned as “High”. A Name_Quality of “Medium” is returned when certain keywords such as “CRANE” are found in the system gender table indicating that this could possibly be either a company or individual.

Variable_Default, Variable_Default2

Syntax: Variable_Default = StringLiteral
 Variable_Default2 = StringLiteral

Description:

Set this property to “FML” (natural order) or “LFM” (inverse order) to indicate a default output format. When ambiguous name components make it impossible to determine the name format, NetGender will default to this setting. This setting is only active when the Name_Style is set to “Variable”.

Note:

As an alternative to the “VARIABLE” style, another very successful approach is to try several different styles checking the Name_Quality and Gender properties after each parse. Start by calling the “Parse” method using “LFM” then “FML” styles. Next, try the “VARIABLE” style and then try each other style in order down the line. Lastly, try “SPLIT” style and then the default ”LFM”, ”FML” and then “VARIABLE” again. After each parse check the Name_Quality and Gender properties and exit at the first sign of a good parse.

Company_Check

Syntax: Company_Check = Boolean

Description:

Set this property to Boolean (True/False) to indicate if an attempt should be made to identify the Name_In string as a company. Set Company_Check to False if your data has no company names. **Default is “False”.**

Nickname_Check

Syntax: Nickname_Check = Boolean

Description:

Set this property to Boolean (True/False) to indicate if an attempt should be made to identify any nicknames associated with Name and Name2. Set Nickname_Check to True if you want nicknames returned in Nicknames and Nicknames2. **Default is “False”.**

Gender_Confidence

Syntax: Gender_Confidence = Integer

Description:

Set this property to a value between 51% and 100% representing the cutoff point below which a gender is considered neutral. Each name in the System Gender Table is encoded with a percentage from 51% to 100% based on the proportion of males to females for a particular name. After the name lookup, if a percentage is found that is below the “Gender_Confidence” level, NetGender will return a gender of Neutral for that name. **Default is 70%.**

Neutral Gender Breakdown:

| <u>Gender_Confidence</u> | <u># of Neutral</u> | <u>% of Gender Table</u> | |
|--------------------------|---------------------|--------------------------|------------------------|
| 100% | 7,800 | 7.8% | |
| 95% | 6,400 | 6.4% | |
| 90% | 5,100 | 5.1% | |
| 80% | 3,900 | 3.9% | |
| 70% | 2,400 | 2.4% | <u>Default Setting</u> |
| 60% | 900 | .9% | |
| 51% | 0 | .0% | |

Reference_File_Path, Gender_File_Path (changed in v3.0+)

Syntax: Reference_File_Path = String
 Gender_File_Path = String

Description:

Set this property to the full path and file name of the following system and user files:

NetGender.gnd is a system file that contains the main Genderization Control Tables.

NetGender.ref is a user-defined file containing the tables for Prefix, Suffix, Filter and Connector identification as well as Gender and Nickname Overrides.

A standard set of these files is supplied and installed in the NetGender installation folder. You can relocate NetGender.ref and NetGender.gnd files to any other folder as long as you provide the complete path and file name information in each respective path string. **Default is NetGender installation folder.**

See “Updating User Control Tables” later in this guide for instructions on customizing this file.

Output_Case (changed in v3.0+)

Syntax: Output_Case = StringLiteral

Description:

Set this property to “Upper”, “Mixed” or “None” to indicate your capitalization preference for the output name and its components. **Default is “None”.**

Static_Key_Name (new in v3.0+)

Syntax: Static_Key_Name = String

Description:

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

Static_Key (new in v3.0+)

Syntax: Static_Key = String

Description:

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

Name_Out, Name2_Out (read only)

Syntax: String = Name_Out
 String = Name2_Out

Description:

After invoking the “Parse” method, this property will contain the corrected name string, in natural order (FML), from the Name_In property.

Name_Filtered_Data, Name2_Filtered_Data (read only)

Syntax: String = Name_Filtered_Data
 String = Name2_Filtered_Data

Description:

After invoking the “Parse” method, this property will contain all data that was filtered out before processing according to the “[NameFilter]” section of NetGender.ref file.

See “Updating User Control Tables” later in this guide for instructions on customizing this file.

Name_Quality, Name2_Quality (read only) (changed in v3.0+)

Syntax: String = Name_Quality
 String = Name2_Quality

Description:

After invoking the “Parse” method, this property is set to “Low”, “Medium” or “High” to indicate how complete a name appears.

Prefix, Prefix2 (read only)

Syntax: String = Prefix
 String = Prefix2

Description:

After invoking the “Parse” method, this property is set to the name Prefix component of Name_In. Values will be valid prefixes (Mr, Mrs, etc.) or blank.

First, First2 (read only)

Syntax: String = First
 String = First2

Description:

After invoking the “Parse” method, this property is set to the First Name component of Name_In. Value will be an individual’s first name or blank.

Middle, Middle2 (read only)

Syntax: String = Middle
 String = Middle 2

Description:

After invoking the “Parse” method, this property is set to the Middle Name component of Name_In. Value will be an individual’s middle name or blank.

Last, Last2 (read only)

Syntax: String = Last
 String = Last2

Description:

After invoking the “Parse” method, this property is set to the Last Name component of Name_In. Value will be an individual’s last name or blank.

Suffix, Suffix2 (read only)

Syntax: String = Suffix
 String = Suffix2

Description:

After invoking the “Parse” method, this property is set to the Suffix component of Name_In. Values will be valid suffixes (Jr, Sr, Esq, etc.) or blank.

Company, Company2 (read only)

Syntax: String = Company
 String = Company2

Description:

After invoking the “Parse” method, this property is set to the Company component of Name_In. Values will be a company name or blank.

Gender, Gender2 (read only)

Syntax: String = Gender
 String = Gender2

Description:

After invoking the “Parse” method, this property is set to the Gender indicated by the “First” property. Values will be “M”, “F”, “N”, “C” OR “U”. (Male, Female, Neutral, Company, Unknown).

Nicknames, Nicknames2 (read only)

Syntax: String = Nicknames
 String = Nicknames2

Description:

After invoking the “Parse” method, this property is set to a list of common nicknames associated with the names returned in First and First2. Values will be a list of one or more common nicknames. Nicknames are only returned when Nickname_Check property is set to True.

Accessing Individual Nicknames - VB Sample Code

```
Dim Nickname As String
Dim Gender As String
Dim i As Short

cbNicknames.Items.Clear()

For i = 1 To 1009 Step 16
    If Mid(NetGender.Nicknames, i, 16) = "" Then Exit For
    Nickname = Mid(NetGender.Nicknames, i, 15)
    Gender = Mid(NetGender.Nicknames, i + 15, 1)
    cbNicknames.Items.Add(Nickname & Space(16 - Len(Nickname)) & Gender)
Next i

If cbNicknames.Items.Count > 0 Then cbNicknames.SelectedIndex = 0
```

Return_Code (read only) (new in v3.0+)

Syntax: String = Return_Code

Description:

After invoking the “Parse” method, this property is set to blank upon successful completion. Some common exceptions are listed below. Most exceptions usually occur on the first call to the Parse method. *This property should be examined on each return from the Parse method.*

Common Return Codes:

| | |
|------------------|--|
| R35 | Reference File Not Found |
| G04 | Gender File Corrupt |
| G35 | Gender File Not Found |
| S00 | Invalid Name Style |
| T00 | Suffix Table Limit Reached (1024) |
| T01 | Nickname Override Table Limit Reached (1024) |
| L00 | License Expired |
| L01 | Static Key Validation Failed |
| L50 - L69 | License Validation Failed |

Clear

Syntax: NetGender.Clear

Description:

When this method is invoked, all output properties are set to null.

Parse

Syntax: NetGender.Parse

Description:

When this method is invoked, each element of the Name_In property will be inspected. Multiple names are separated, each name is gender coded and placed into the Name_Out property. Each individual element of the Name_In property will be placed into the appropriate name component property. And, if the Nickname_Check property is set to True, a list of common nicknames is returned in Nicknames and Nicknames2. The Return_Code property is also set and should be checked after each invocation of the Parse method. *See "Return_Code" property.*

Updating User Control Tables

NetGender.ref contains the tables that control Prefix, Suffix, Filter and Connector recognition in addition to Gender and Nickname Overrides. It is located in the “NetGender” application folder. Use Notepad or a similar text editor to edit the file. Detailed instructions on the format of the entries are contained within the file. You can also relocate this file. *See Reference_File_Path property.*

NetGender allows you to specify which Prefixes and Suffixes are to be recognized as well as your preferred abbreviations.

The filter section of the table allows you to specify which, if any, characters, words or phrases are to be ignored during processing. All filters that were found in the Name_In string will be stored in the Name_Filtered_Data property.

Extensive tables are included. Below are a few examples:

[NamePrefix]

| | |
|-----|------------|
| M&M | Mr. & Mrs. |
| M/M | Mr. & Mrs. |

[NameSuffix]

| | |
|---------|-------|
| PHD | Ph.D. |
| MANAGER | Mgr. |

[NameConnector]

&
C/O
GUARDIAN OF

[NameFilter]

ETAL
TRUST

[GenderOverride]

| | |
|---|-------------|
| M | BINKIE |
| C | TED'S SHEDS |

[NicknameOverride]

BINK BINKY BINKIE BINKO

If Installation Doesn't Start Automatically:

- Select **Start > Run** from the Task Bar.
- Type CD-ROM drive letter followed by “:\NetGender35.msi” and press enter.

In the folder “Program Files\The Software Company\NetGender 3.5” you will find a sample program named: VBSample.vbproj. There is also a compiled version called VBSample.exe that you can run to demonstrate NetGender.

Deploying Your Applications

Be sure to include the following in your installation package:

NetGender.dll
NetGender.ref
NetGender.gnd

In addition to the above, there is a small runtime package that is installed into the Global Assembly Cache (GAC) of the target machine. There are two ways to do this:

Include the **MergeModuleRuntimeNET20.msm** with your installation package

- OR -

Execute **StandaloneRuntimeNET20.exe** for each new installation

There's an explanation of each at the following link:

http://www.SoftwareCompany.com/dotnet/dotnet_runtime.htm

This product is initially licensed for a period of 30 days or up to 1000 calls. It must be registered to continue using it after this evaluation period. Please contact us at:

SalesDept@SoftwareCompany.com

P: 303/838-1223

F: 303/838-1224