# US Name Data Preparation

## Overview

The US Name Data Preparation Function is designed to parse an unstructured name string into a set of Name Components (See Table 1). The name components produced by the US Name Data Preparation.

## Semantics

The US Name Data Preparation Function takes an input as a single, unstructured character string representing a complete US names including  Prefix Title, Given Name, Surname, Generational Suffix and Suffix Title fields. The Data Preparation Function then parses the name string into a set of Name Components that can be references in other parts of the Entity Script in which the Data Preparation function appears.

The Name Components and Component Codes defined by the US Name Data Preparation Function are shown in Table 1 below:

|  |
| --- |
| Table 1:US Name Components |
| Prefix Title |
| Given Name |
| Surname (Last Name) |
| Generational Suffix |
| Suffix Title |

## Requirements

1. Tokenization of the name string into a token list. The name line input shall be parsed into separate word tokens according to the following parsing rules
   1. The name string shall be split into name token strings where a name token string is a
      1. Consecutive sequences of letters, dot(.) and hyphens (-)
   2. All other characters shall be treated as delimiters (token separators) and discarded with the exception of the comma (,) character. The comma character shall not be placed in the token list, but it shall be preserved in the pattern string as explained in Requirements Section 2.
   3. Each token string parsed from the input string shall be added to a Token List in the same order as it occurred left-to-right in the name string.
   4. All letters in each token shall be converted to upper case.
   5. A Pattern Array shall be built by representing each token in the Token List with single-character pattern code (See Table 2). Note that at all times there must be a one-to-one relationship between the items in the name token list and the single-character symbols in the pattern array.
   6. The pattern codes shall be assigned by one method. It is by searching for the token value in the US Name Token Table.
   7. Assignment of pattern codes by table search.
      1. Each row of the US Name Token Table shall comprise a keyword-value pair
      2. The keyword shall be a character string conforming to a name token as defined in the previous requirements for name parsing rules.
      3. The keyword shall be separated from its value by a tab character.
      4. The value for each keyword shall be a single letter pattern code indicating the most common function of the name token in a US name.
      5. The pattern code shall be one of the single-character codes from Table 2 under the Table section,
      6. If a name token is found as a keyword in the US Name Token Table, the token shall be assigned the pattern code given by the table.

|  |  |  |
| --- | --- | --- |
| Table 2: US Name Token Codes | | |
| Type | Pattern Code | Token Description |
| Table | P | Prefix Title |
| G | Given Name |
| L | Surname (Last Name) |
| J | Generational Suffix |
| Q | Suffix Title |

* 1. In the case that the token in not found as a keyword in the US Name Token Table, then pattern code assigned to the token shall be assigned.

Example 1: Input String: “DR BRYCE ABBOTT, JR, IQCP”

Initial Tokenization After Lookup

|  |  |  |
| --- | --- | --- |
| Pos | Token | Code |
| 1 | DR | P |
| 2 | BRYCE | G |
| 3 | ABBOTT | L |
| 4 | JR | J |
| 5 | IQCP | Q |

|  |  |
| --- | --- |
| Pos | Token |
| 1 | DR |
| 2 | BRYCE |
| 3 | ABBOTT |
| 4 | JR |
| 5 | IQCP |

1. After the Token List and Pattern Array have been completed, the values of the US Name Components (See Table 1) shall be assigned values using the tokens from the Token List. The rules for assigning the values to the components are as follows:
   1. First, the codes in the pattern array shall be concatenated into a pattern string as follows
      1. If two consecutive codes in the pattern array correspond names token strings that had a comma (,) character between them in the original names string, then the two pattern codes will be concatenated together with a comma character between them.
      2. Else if the corresponding name token strings did not have a comma character between them in the original name string, then the two pattern codes will be concatenated together without any other character between them.
   2. After all the pattern codes are concatenated into a single pattern string, the pattern string shall be used as a lookup argument in the US Name Pattern Table. If the pattern is found in the table, the table shall return a set of assignment instructions coded into a string value, and the assignment instructions returned from the pattern table shall be used to populate the US Name Items.
   3. A component code listed in the assignment instruction string shall be populated by concatenating the tokens. The tokens shall be concatenated in the same order as it occur in the assignment instructions. The concatenated values shall be separated by one blank (space).

Example 2: Assignment of Component Values

Concatenated Pattern String “PGL,J,Q”

|  |  |  |
| --- | --- | --- |
| Pos | Token | Code |
| 1 | DR | P |
| 2 | BRYCE | G |
| 3 | ABBOTT | L |
| 4 | JR | J |
| 5 | IQCP | Q |

1. Standardization of US Name Functional Components. After all of the components have been assigned values, each component shall be standardized according to the following rules
   1. Name components with an empty string value shall be ignored
   2. Name components with non-empty string values shall be first be standardized by using US Name Standardization Table.
      1. The US Name Standardization Table shall have a key-value pair structure, and the key shall be separated from the value by a tab character.