Feature Summary: Automatic Constituent Input Data Formatting/Validation

Module: Cross Platform

Release: Summer 2010
Document Owner: Thomas Bressie

IMPORTANT NOTE: THIS DOCUMENT INTENDS TO PROVIDE A COMPLETE AND ACCURATE REPRESENTATION OF WHAT CONVIO FULLY EXPECTS TO DELIVER IN SUMMER 2010 (AS OF SEVERAL WEEKS PRIOR TO LAUNCH) BASED ON COMPLETED SOFTWARE DEVELOPMENT. THIS DOCUMENT AIMS TO SERVE AS THE BASIS FOR PLANNING AND PREPARATION WELL IN ADVANCE OF ACCESS TO THE NEW FEATURES. GIVEN THIS DOCUMENT WAS DRAFTED PRIOR TO COMPLETED QUALITY ASSURANCE, ALL FEATURES, CAPABILITIES, SCREENSHOTS, ANSWERS, CLIENT IMPACTS, SITE DATA, PARAMETERS, AND LIMITATIONS LISTED HEREIN, WHILE REASONABLY CERTAIN FOR INCLUSION IN THE RELEASE, ARE SUBJECT TO CHANGE AT ANY POINT UP UNTIL ROLLOUT ON YOUR CONVIO SITE.

Overview

In the Summer 2010 release of Convio Online Marketing (COM), Convio will provide for automatic formatting, validation, and verification of constituent data at the time of capture (such as new online registration). This functionality will increase the quality of data captured in constituent records by increasing data field uniformity from record to record.. As a result, COM will offer a higher quality of constituent record, capturing more accurate constituent information, yielding an overall higher quality housefile for your organization.

The constituent data formatting/verification tools provided in the Summer 2010 release address the following constituent data types:

- First/Middle/Last Name
- Email Address
- Phone Number
- Mailing Address

First/Middle/Last Name Formatting, Verification

The Summer 2010 release of COM includes an algorithm to process each name field – first, last, middle – and normalize the constituent-submitted content in properly capitalized and punctuated name strings. Below is a functional description of the algorithm.

1. Remove any whitespace (leading, trailing, duplicate) from the name string:
    a. *Examples:* "mackey " => "mackey", "de la reza" => "de la reza"
2. If the name string is mixed case, do not change the capitalization:
    a. *Example: "lavelle"* will be corrected; *"LaVelle"* will not.
3. After performing a lookup against the keys listed in the FORMAT_NAME_EXCEPTIONS table (see Appendix A), and ignoring any differences in capitalization, if any matches are found, replace the current name string with the match in the FORMAT_NAME_EXCEPTIONS table and do not process this name string further.
    a. *Example: "mackey" => "Mackey"; no further processing performed.*
4. Capitalize the first letter in each word in the string, as delimited by space, hyphen, or period:
    a. Examples: "de la reza" => "De La Reza" , "st.clair" => "St.Clair"
5. Apply all the rules defined in the FORMAT_NAME_RULES table (see Appendix B) to the resulting string:
    a. *Examples:* "De La Reza" => "de la Reza" , "O'hara" => "O'Hara", "Von Den Linden" => "von den Linden"

After each name string component of the constituent input has been processed by the above algorithm, the properly formatted and validated name strings are written to the constituent record.

Email Address Formatting, Validation/Verification

Understanding the importance of useful email contact information, Convio has added the capability to the Summer 2010 release to ensure email addresses received from the constituent are properly formatted and valid. While exhaustive email validity confirmation is outside the scope of this improvement, COM not only adds the capability in this release to verify that email addresses conform to the RFC822 standard for internet email addresses and also corrects common user input errors that previously resulted in unusable email addresses. Below is a functional description of the algorithm to format, validate and verify email addresses captured from the constituent.

1. Scan received email address string for '@' character and domain name. If either of these components is missing, the correction algorithm exits and returns to the user interface an error message indicating the email address submitted is invalid.
2. Attempt to match the email domain with the following domains, allowing for detection and correction of single character misspellings (such as, Yaho vs. Yahoo, hatmail vs. hotmail)
    a. yahoo.com, aol.com, hotmail.com, comcast.net, sbcglobal.net, msn.com, cox.net, verizon.net
3. Check the domain for a valid MX (mail exchanger) record. If a valid MX record is found, no further correction is done.
4. Attempt to match the email TLD (top-level domain) with the following TLDs in listed order, allowing for detection and correction of single character misspellings (such as, con vs. com, orf vs. org)
    a. com, net, org, edu, gov, info, mil, biz, aero, asia, cat, coop, int, jobs, mobi, museum, name, pro, tel, travel

After the email text string constituent input has been processed by the above algorithm, the properly formatted and validated email address is written to the constituent record. If the email address formatting/validating/verifying algorithm cannot continue processing at any step due to errors in the constituent input data, the algorithm prompts the user interface to indicate to the constituent that invalid input was detected and to request input correction.

Phone Number Cleaner

Given the many formats in which a phone number can be submitted to COM, Convio has implemented in the Summer 2010 release a phone number formatting/verification algorithm that will result in not only a more uniform capture of phone number data, but also detection of some basic error types in that data capture. Below is a functional description of the algorithm used to format, validate and verify email addresses captured from the constituent.

The phone number formatting/verification algorithm consists of two phases: validation and correction.

Phase 1 - Format Validation

- The phone number string received from the constituent input is split into two parts: number and extension.
- The number is checked to determine if it is in NANP Format (refer to description following this section). If the submitted number matches one of the possible format options, the validation is deemed successful.
- If the submitted number does not match one of the possible NANP formats, the number is then checked to determine if it is an International number, matching one of the international format options. If the number does not match any of the possible International formats, the validation algorithm prompts the user interface to inform the constituent of the verification error and requests correction.

NANP Format

- NANP stands for North America Numbering Plan. Convio's implementation of conformity to the NANP standard verifies two key elements:
  - Capture of the correct number of digits, ensuring for domestic phone numbers that the area code value has been captured in addition to the base seven digit phone number.
  - The phone number has been submitted in an appropriate format for a domestic phone number, as represented in the format examples below. Note: the formats listed below may be combined to form many valid combinations; in no way is this format list exhaustive.
    - (512)652.2600
    - 512 652 2600
    - 512-652-2600
    - 512.652.2600
    - 512/652/2600
    - 5126522600

International Format

- A valid international number is defined as a string of digits separated into groups of two, three, or four digits by one of the following characters: dot (.), forward slash (/), hyphen (-), or space ( ).
- The first group of digits can be surrounded by parentheses.
- The phone number string may be preceded by the 'plus' symbol '+' indicating an international phone number.
- The string may be between ten and fifteen digits, with the expectation that strings of length exceeding thirteen digits contain the '011' prefix combination of digits indicating an international phone number.

Phase 2 – Format Correction

Phone number format correction is performed only on phone number strings successfully passing the first validation phase.

Format Correction Algorithm

- Number and extension (if present) are corrected separately and then concatenated.
- Phone number extension correction transforms the received string into 'ext ABC' where ABC represents the numeric string value representing the extension. See the section below on extension input guidelines for valid formats.
- All non-numeric characters are removed, and the remaining numeric characters are written to the constituent record.

Extension Input Format Guidelines

- A valid extension is a string consisting of: a prefix, zero or more white spaces, and a group of digits.
  - A valid prefix is one of: extension, ext, x followed be an optional character dot (.) or colon (:)
  - A group of digits is defined a group of integer values [0-9].
- Examples of valid extensions:
  - ext.234
  - ext234
  - ext:234
  - ext*234
  - x.234
  - x234
  - x. 234
  - x 234
  - extension 234
  - extension. 234
  - extension 234

Mailing Address Formatting/Validation

Convio has implemented in the Summer 2010 release the capability for COM to call an existing address correction service to ensure that physical/mailing addresses provided by the constituent are not only properly formatted, but are also valid physical/mailing addresses.

Both correction and validation of an address are done by calling the existing address correction service. COM provides the constituent input data to the service, and receives either a properly formatted and verified physical mailing address (US addresses only) or an error indicating the address could not be verified. Addresses that cannot be corrected will be written to the constituent record as provided by the constituent, after prompting the constituent once for their review/correction indicating the verification operation failed.

Key Features & Benefits

Features

- The constituent data formatting/verification tools provided in the Summer 2010 release validate the following constituent input data types: First/Middle/Last Name, Email Address, Phone Number, Mailing Address.

Benefits

- This functionality will ensure a uniform quality in your constituent record with more accurate constituent information (including some correction of the most common error types), and an overall higher quality housefile for your organization.
- More uniform constituent record creation enhances the duplicate checking algorithm by ensuring a more accurate duplicate candidate record pool before duplicate scoring. Employing the constituent data formatting/verification tools can assist in reducing the number of new duplicate constituent records created within your housefile.

Considerations for Clients

- All formatting algorithms described above operate independently of one another; these tools may be used individually, all together, or in any combination desired.
- When the above formatting algorithms encounter input that cannot be validated, they trigger the user interface to signal an error to the constituent and ask for resubmission of the unverifiable data. If the constituent resubmits the same data or if the same error recurs upon resubmission, the formatting tool is bypassed and the constituentprovided value is written to the constituent record.

Feature Configuration

- All formatting tools specified in this document for release with Summer 2010 are disabled by default, and can be enabled by contacting Convio support.

Appendix A – Exceptions to Name Cleaner Rules Actions (FORMAT_NAME_EXCEPTIONS)

- D'Amico
- DaCosta
- DaSilva
- DaVeiga
- DeAngelis
- DeAngelo
- DeBartolo
- DeBell
- DeBenedictis
- DeBerry
- DeBlasio
- DeBoer
- DeBonis
- DeBord
- DeBruin
- DeCamp
- DeCarlo
- DeCaro
- DeChant
- DeChirico
- DeCicco
- DeClue
- DeCosta
- DeCuir
- DeFalco
- DeFazio
- DeFelice
- DeFeo
- DeFilippis
- DeFilippo
- DeFoe
- DeFord
- DeForest
- DeFrancesco
- DeFranco
- DeGarmo
- DeGennaro
- DeGolyer
- DeGraff
- DeGraw
- DeGregorio
- DeGroat
- DeGroot
- DeHaan

- DeHart
- DeHaven
- DeJesus
- DeJohn
- DeJong
- DeLacy
- DeLancey
- DeLapp
- DeLaura
- DeLaurentis
- DeLay
- DeLeo
- DeLeon
- DeLisle
- DeLoach
- DeLong
- DeLorenzo
- DeLorey
- DeLozier
- DeLuca
- DeLucia
- DeMaio
- DeMarco
- DeMaria
- DeMars
- DeMartini
- DeMartino
- DeMasi
- DeMatteo
- DeMay
- DeMello
- DeMeo
- DeMeritt
- DeMichele
- DeMille
- DeMoss
- DeMott
- DeMuth
- DeNicola
- DePalma
- DePaolo
- DePasquale
- DePaula
- DePew
- DePhillips
- DePriest

- DePuy
- DeRosa
- DeRose
- DeRosier
- DeRouen
- DeSalvo
- DeSantis
- DeSanto
- DeSena
- DeShazo
- DeSilva
- DeSimone
- DeSoto
- DeSouza
- DeSpain
- DeStefano
- DeVaughn
- DeVault
- DeVere
- DeVille
- DeVincenzo
- DeVita
- DeVito
- DeVoe
- DeVore
- DeVos
- DeVries
- DeWald
- DeWalt
- DeWees
- DeWeese
- DeWitt
- DeWolf
- DeYoung
- DelGiudice
- DelGreco
- DelVecchio
- DellaLoggia
- DesJardins
- DesLauriers
- DiBello
- DiBenedetto
- DiBernardo
- DiBiase
- DiBona
- DiCamillo

- DiCarlo
- DiCicco
- DiCostanzo
- DiDomenico
- DiFazio
- DiFiore
- DiFrancesco
- DiGiacomo
- DiGioia
- DiGiorgio
- DiGiovanni
- DiGrazia
- DiJulio
- DiLallo
- DiLeo
- DiLorenzo
- DiLoreto
- DiMaggio
- DiMambro
- DiMarco
- DiMaria
- DiMartino
- DiMatteo
- DiMattia
- DiMauro
- DiMeglio
- DiNardo
- DiNatale
- DiPalma
- DiPaola
- DiPaolo
- DiPasquale
- DiPierro
- DiPietro
- DiRocco
- DiSalvo
- DiSanto
- DiStefano
- DiTullio
- DiVito
- DuBois
- DuBose
- DuFresne
- DuPre
- FitzGerald
- LaBarbera

- LaBarge
- LaBarre
- LaBella
- LaBelle
- LaBianca
- LaBonte
- LaBounty
- LaBranche
- LaBrecque
- LaBrie
- LaCasse
- LaChance
- LaChapelle
- LaClair
- LaCour
- LaCroix
- LaCrosse
- LaDue
- LaFave
- LaFever
- LaFlamme
- LaFleur
- LaFollette
- LaFon
- LaFontaine
- LaForce
- LaForest
- LaForge
- LaFrance
- LaFreniere
- LaGorga
- LaGrone
- LaGuardia
- LaLiberte
- LaLonde
- LaManna
- LaMantia
- LaMarca
- LaMarche
- LaMonica
- LaMontagne
- LaMothe
- LaMotte
- LaPlaca
- LaPlant
- LaPlante

- LaPoint
- LaPointe
- LaPorta
- LaPorte
- LaPrade
- LaReau
- LaRocca
- LaRocco
- LaRoche
- LaRochelle
- LaRock
- LaRocque
- LaRoe
- LaRosa
- LaRose
- LaRue
- LaSalle
- LaTorre
- LaTour
- LaValle
- LaValley
- LaVerne
- LeAnn
- LeBaron
- LeBeau
- LeBlanc
- LeBoeuf
- LeBrun
- LeClair
- LeClaire
- LeCompte
- LeCroy
- LeDoux
- LeDrew
- LeDuc
- LeFevre
- LeGrand
- LeJeune
- LeMaire
- LeMaster
- LeMasters
- LeMay
- LePage
- LeRoux
- LeRoy
- LeSage

- LeTourneau
- LeVan
- LeVasseur
- LeeAnn
- LoCicero
- LoGerfo
- LoPreste
- LoPresti
- LuAnn
- Mac
- Macaluso
- Macartney
- Macaulay
- Macauley
- Maccarone
- Macchia
- Mace
- Maceda
- Macedo
- Macek
- Macey
- Mach
- Machacek
- Machado
- Machen
- Machin
- Macias
- Maciejewski
- Mack
- Mackay
- Macke
- Mackel
- Mackey
- Mackie
- Mackin
- Macklin
- Macko
- Macky
- Macomber
- Macon
- Macri
- Macy
- O'Brien
- O'Connell
- O'Connor
- O'Dell

- O'Donnell
- O'Grady
- O'Hara
- O'Keefe
- O'Leary
- O'Leyar
- O'Loughlin
- O'Malley
- O'Neal
- O'Neil
- O'Neill
- O'Rourke
- O'Sullivan
- StClair
- StJohn
- VanBuren
- VanBuskirk
- VanDyke
- VanEtten
- VanHorn
- VanHouten
- VanLandingham
- VanPelt
- VanSickle
- VanWormer
- deAndrade
- deForest

Appendix B – Name Cleaner Rules (FORMAT_NAME_RULES)

- Capitalize the letter following Mac
- Capitalize the letter following Mc
- Capitalize the letter following a **single** letter and an apostrophe (Hebrew and Arabic transcription use apostrophe medially for glottal and pharyngial stops; Russian transliteration uses it for myagkiy znak and sometimes palatalized consonants. In neither case should we capitalize the following letter, so this rule produces "O'Hara" but "Ye'or" instead of incorrect "Ye'Or".)
- **ab** Welsh ab is lower-case
- **ad** Arabic ad is lower-case
- **al** Arabic al is lower-case
- **ap** Welsh ap is lower-case
- **as** Arabic as is lower-case
- **d'** French d' is lower-case
- **da** Portugese da is lower-case
- **de** French, Dutch, and Spanish de is lower-case
- **den** German den is lower-case
- **der** German/Dutch der is lower-case

- **dos** Portuguese dos is lower-case
- **ibn** Arabic ibn is lower-case
- **la** Spanish/French la is lower-case
- **n** Arabic/Gaelic n' is lower-case
- **ten** Dutch ten is lower-case
- **ul** Urdu ul is lower-case
- **und** German und is lower-case
- **van** Dutch van is lower-case
- **von** German von is lower-case
- **y** Spanish y is lower-case
- **zu** German zu is lower-case