OmniUpdate Training Tuesday
Advanced Find/Replace with Regex

WebEx Event #: 805 968 020

Audio will be heard on your computer speakers.

If you do not have working computer speakers, call 1-408-792-6300.

Enter event number and attendee ID or press # if no attendee ID.

Webcast will start at the top of the hour.

Presented by:
Robert Kiffe
Senior Support Specialist
Agenda

• Let’s Look!
Global Find and Replace
• A Difficult Fix
• Introduction to Regex
• A Solution
• Q&A
Let’s Look!

(Global) Find and Replace
Global ‘Find and Replace’

• Administrator’s Only (User Level 10)
• Searches a single site
• Adjust ‘Scope’ to limit searchable content
Global ‘Find and Replace’

- Find
  - Simple Search
- Preview Replace
  - Perform Find/Replace and display results (before actually making the changes)
A Difficult Fix

Where a simple text Find and Replace won’t ‘cut’ it
The Problem

• Setting External Links to open into a new tab/window

• Need to add attribute: target="_blank"
The Problem

• Pages currently contain a mix of links:
  – `<a href="http://www.google.com">…
  – `<a href="/academics/index.html">…

• Desired result:
  – `<a href="http://www.google.com" target="_blank">…
  – `<a href="/academics/index.html">…
The Problem

• How do we find all external links?
• How do we safely add an extra attribute?
Introduction to Regex

AKA ‘Regular Expressions’
Regex: Background

• A pattern that ‘describes’ a certain amount of text

• The concept arose in the 1950s when the American mathematician Stephen Cole Kleene formalized the description of a regular language. (Thanks Wikipedia)
Regex: Background

- Used in almost every major programming language
- There are multiple ‘engines’ that run Regex patterns
  - OU Campus uses “JAVA” syntax
Regex: Basics

• Literal Matches
  – Most characters match exactly themselves
  – Case Sensitive

Robert does not like to be called robert.
Regex: Basics

• Special Characters
  – Symbol characters that need to be escaped to match directly
  – Full List: `\^$.|?*+()[]{}`
  – Wildcard Character: . (matches any letter, number, or symbol)
  – Groups: () []
  – Escape/Shorthand: \
  – Quantifiers: ? * + ... {}
  – Start/End of ‘String’: ^$ (in order)
  – Logical ‘Or’: |
• Special Characters: Example
  – Wildcard: “period”

`Robert` does not like to be called `robert` or `Goobert`.
Regex: Basics

• Quantifiers
  – Symbol characters that define how many of the previous character(s) to match
  – 0 or 1: \(?\)
  – 0 or More: \(*\)
  – 1 or More: \(+\)
  – Use Curly Brackets to indicate an exact number
    • Example: \(A\{3\}\) matches “AAA”
  – Only modifies the nearest, previous character (or group)
Regex: Basics

- Quantifiers: Example
  - 0 or 1: “question mark”

Robert does not like to be called Goobert or Roberta.
Regex: Basics

• Character Sets

  – Characters encased inside square brackets define all possible matches for the defined quantity of characters
  – A quantifier placed directly after the set will affect the whole character set
  – Placing a “^” as the first item in the set creates a ‘negative pattern’.
• Character Sets: Example
  – With Quantifier: 1 or More “plus”

```
[aA]+rdvark
```

Aardvarks like to play with other aardvarks, but not with Aaardvarks.
Regex: Advanced

- Character Sets: Example
  - Negative Set: with “^”

```
href="[^]+"
```

```
<a href="/academics/index.pcf">Link</a>.
```
• Shorthand References
  – Certain characters can reference a range of characters when preceded by a backslash (\)
    • Common Examples:
      – \d : matches all digit characters (0-9)
      – \w : matches all ‘word’ characters (a-z, A-Z, and certain special characters)
      – \s : matches all ‘space’ characters (including line breaks)
    • Using the capital letter will ‘inverse’ the match
      – Example: \S matches all non-space characters
• Shorthand References: Example
  – Word Character “\w”

\wardvark

Aardvarks like to play with other aardvarks, but not with Aaardvarks.
Regex: Capture Groups

- Encapsulate a character sequence using parentheses
  - Example: ([Tt]he)
  - Add a quantifier to make it optional, or to find multiple matches: ([Tt]he )?Bandname
Regex: Capture Groups

• In your “Replace” string, refer to your groups using the “dollar sign” and then the group number

• Count the opening “(“ to determine the correct #
Regex: Advanced

• Capture Groups: Example

FIND

http://(www\.)?(google.com)


REPLACE

http://$2

A Solution

Remember that ‘problem’ with external links?
A Simple Solution

• Find (w/ Regex):
  – `<a\s*href="(https?[^\"]\)*"\s*>`

• Replace:
  – `<a href="$1" target="_blank">`
A Better Solution

• Find (w/ Regex):
  – (<a[^>]*)(href="https?[^"]*")([>]*>)

• Replace:
  – $1$2 target="_blank" $3
A Better Solution: Cleanup

- Find (w/ Regex):
  - (\<a[^>]*\>(target=\"[^\"]*\")\([^>]*\)target=\")

- Replace:
  - $1$3
Q&A
Next Training Tuesday

Next Month’s Training Tuesday will be held on February 28, on the topic of Accessibility. Be sure to visit the OCN or Support Site for further details.

http://ocn.omniupdate.com
http://support.omniupdate.com/oucampus10
Thank You!