EAB

Text Parsing and Handling Strings in Construct

A Rapid Insight Deep Dive

March 10, 2022

Presentation Description

Many data projects hit a real slowdown when it comes to a need for processing text data. Multi-key fields often need to be parsed, code values might need to be translated, manual entry might have introduced typos, or maybe you only need the first letter out of the whole string (another word for text data). The fix might seem easier to do by hand but scaling up a manual fix to include thousands of rows isn't feasible. Even so, it may seem insurmountable to create a systematic process to parse, clean, substitute, or otherwise transform the text you need to analyze.

In this webinar, expect to learn how Construct can:

- Parse out multi-key fields (e.g., "AARN, 20FA, ")
- Replace values within strings at multiple levels of precision
- Account for typos in manual entry fields (e.g., "main street" vs. "main st.")
- Shorten or lengthen values (e.g., adding or removing leading zeroes)
- Extract subfields from strings dynamically

This webinar will help you leverage Construct towards scalable text transformations on your data. We're not assuming any background experience with text manipulations, so come as you are, and find out how you can expertly manipulate strings.

Meet Your Presenters

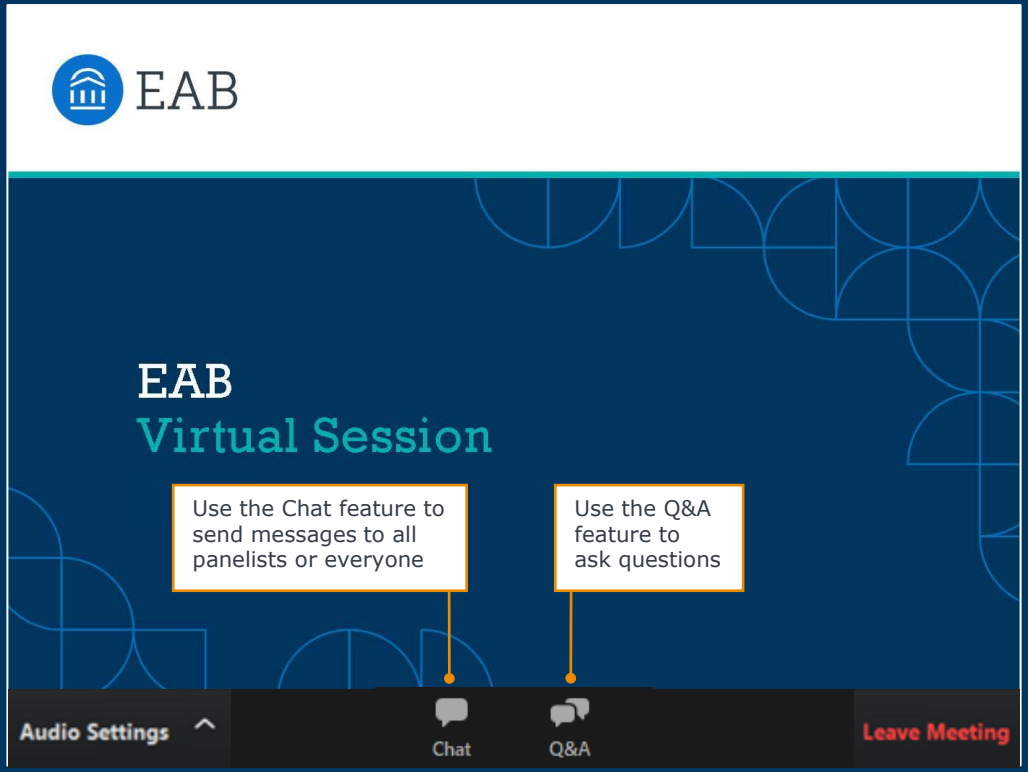
James Cousins

Edify Product Analyst

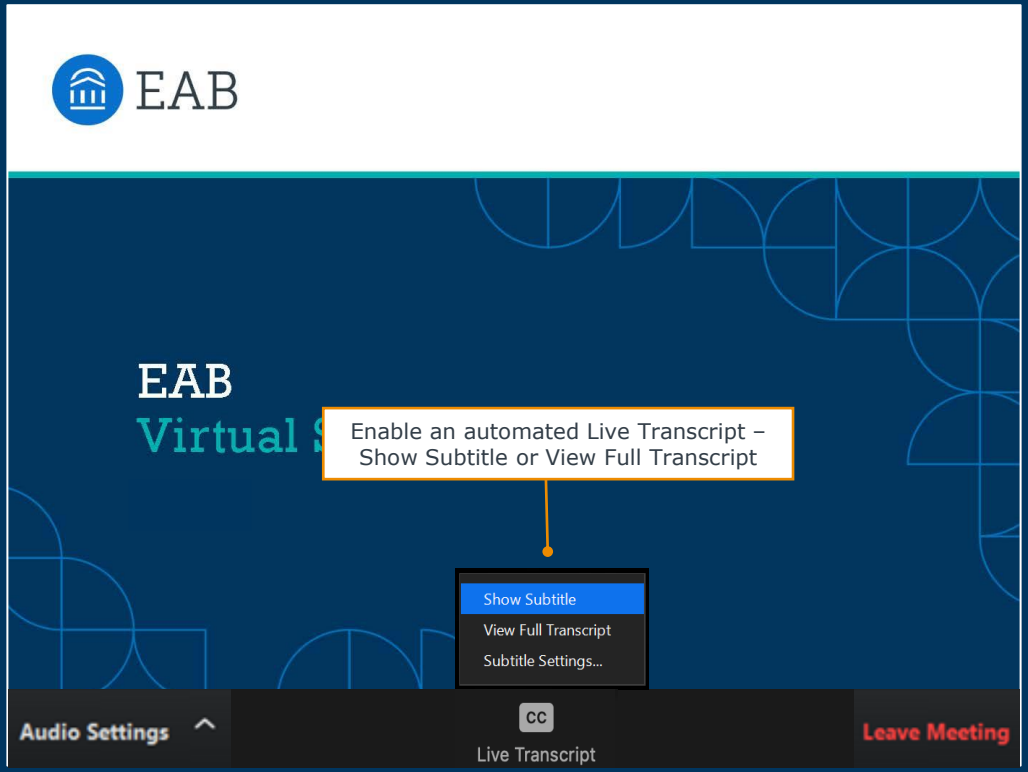
Lily Brennan

Edify Product Analyst

Submit a Question or Comment



Turn on Captions



1. Rapid Insight and EAB
2. Overview of the Presentation Structure
3. Text Parsing Challenges
4. Audience Q & A

A New Partnership to Accelerate Data Democratization in Higher Education

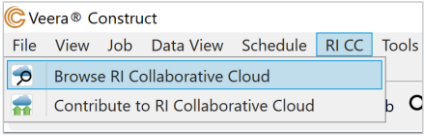
Learn More:

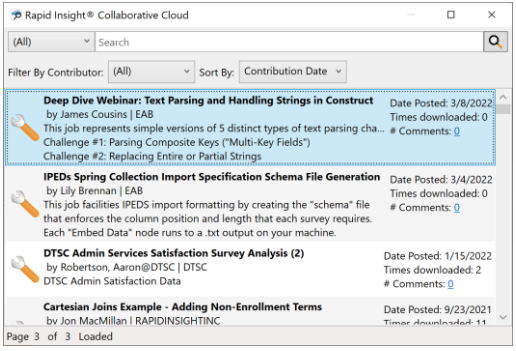
Read more about EAB's latest partnership at https://eab.com/rapidinsight/

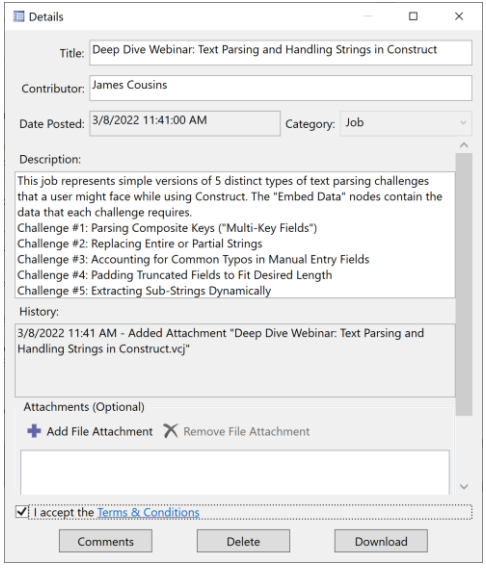
Source: EAB interviews and analysis.

1. Rapid Insight and EAB
2. **Overview of the Presentation Structure**
3. Text Parsing Challenges
4. Audience Q & A

Rapid Insight Collaborative Cloud







Presentation Structure

Classifying the Challenge

It helps to recognize the type of challenge so that you know where to begin solving it.

Common Solution

While there's typically not one singular solution, there's usually a useful go-to technique.

Live Example

We'll show the solution in action, highlighting several alternative solutions along the way.

1. Rapid Insight and EAB
2. Overview of the Presentation Structure
3. **Text Parsing Challenges**
4. Audience Q & A

Poll

How would you rate your proficiency using Construct to handle text data?

a) I have never used Construct to work with text data

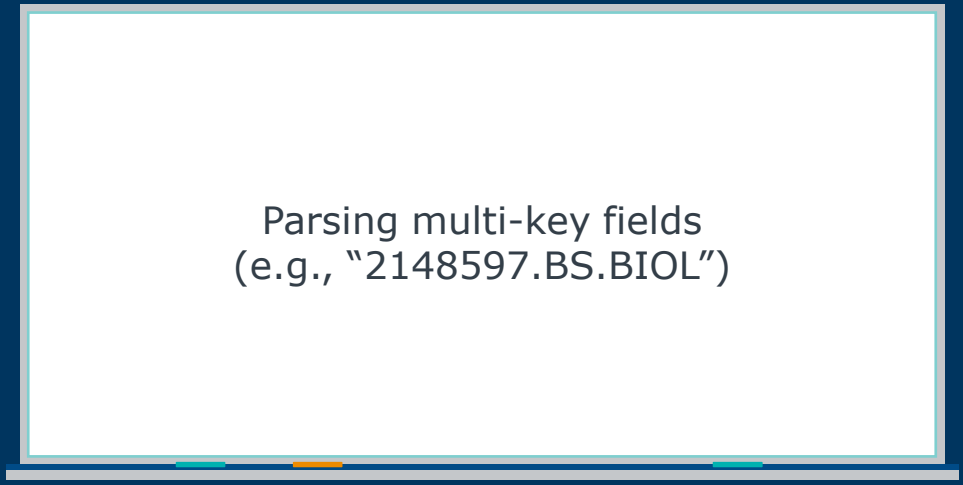
b) I have used Construct to work with text data, but I would like to learn more best practices

c) I am confident handling text data in Construct

Text Parsing Challenges

1. Parsing multi-key fields (e.g., "2148597.BS.BIOL")
2. Replacing targeted text at multiple levels of precision
3. Accounting for typos in manual entry fields
4. Shortening (trimming) or lengthening (padding) values
5. Extracting subfields from strings dynamically

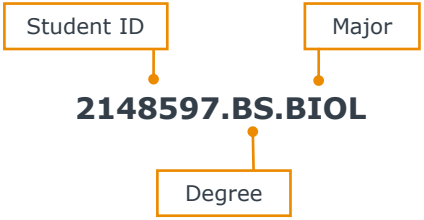
Challenge 1



Multi-Key Fields

Multi-Key Fields

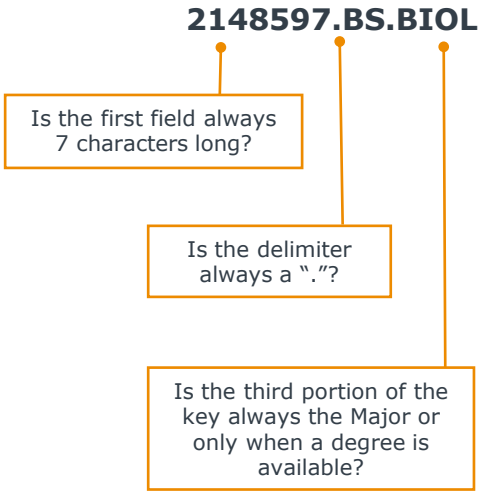
- Technically referred to as a Composite Key
- Concatenates multiple characteristics as a singular key for the record



Common Solutions for Multi-Key Fields

Useful Strategies

- Other Columns in the same table
  - Composite keys are usually comprised of information stored in the same table, across multiple columns
- SUBFIELD()
- SUBSTRING()
- LEFT(), potentially combined with REMOVELEFT()
  - See also, RIGHT(), REMOVERIGHT()



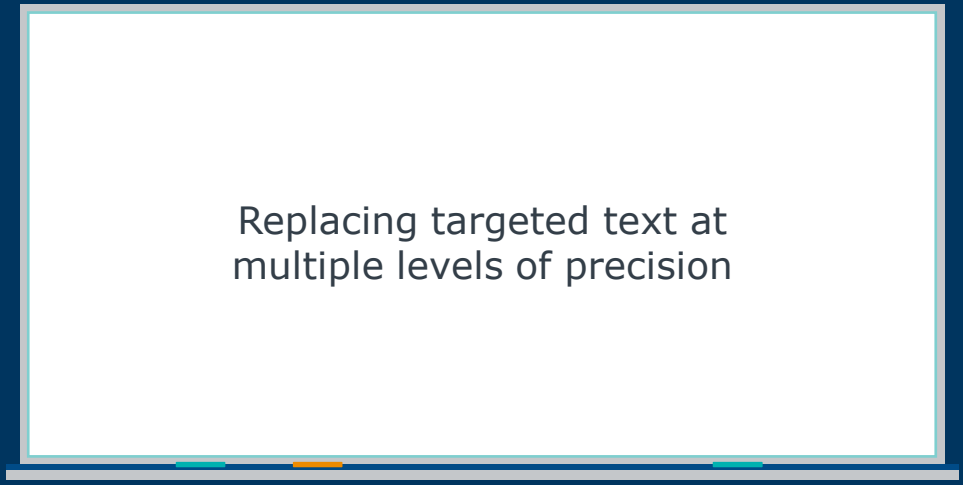
Multi-Key Field Example

What Transform Formulas Would Get You What You Need?

Original Field	Student ID	Degree	Major
2148597.BS.BIOL	2148597	BS	BIOL

Parsed Variable	Using LEFT()/RIGHT() and REMOVELEFT()/REMOVERIGHT()	Using SUBFIELD()	Using SUBSTRING()
Student ID	**LEFT**([A],7)	**SUBFIELD**([A],1,'.')	**SUBSTRING**([A], 1, 7)
Degree	**LEFT**(**REMOVELEFT**([A],8), 2)	**SUBFIELD**([A],2,'.')	**SUBSTRING**([A], 9, 2)
Major	**LEFT**(**REMOVELEFT**([A],11),4)	**SUBFIELD**([A],3,'.')	**SUBSTRING**([A], 12, 4)

Challenge 2



Take What You Like, Leave What You Don't

Program
Automotive Technology – AS
Marine Biology – BS
Religious Studies - MA

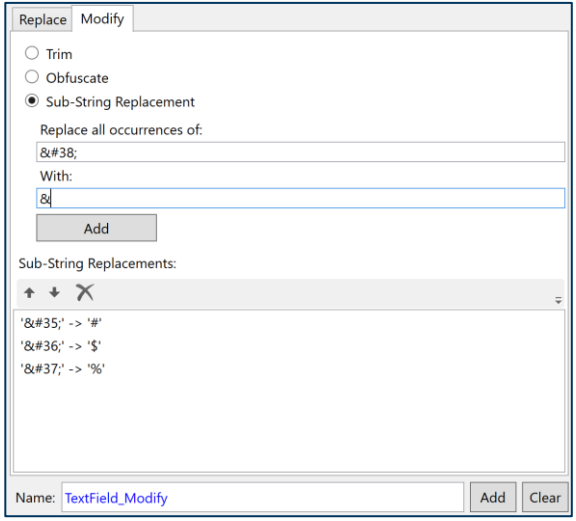
Text Replacements

- The idea is that ***at least some*** of the text needs to be replaced
  - If the entire string needs replacing, it is just a Text Replacement
  - If just some portion of the string needs replacing, it is a "Sub-String" Replacement

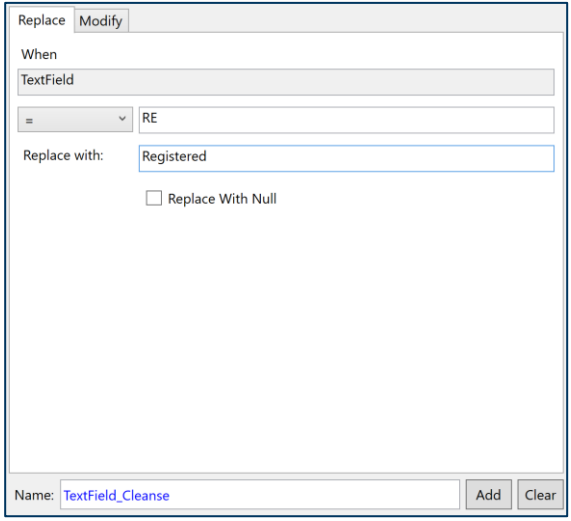
Common Solutions

Partial Replacement | **Cleanse Node** | **Entire Replacement**

"Replace Sub-String"



"Replace Text"



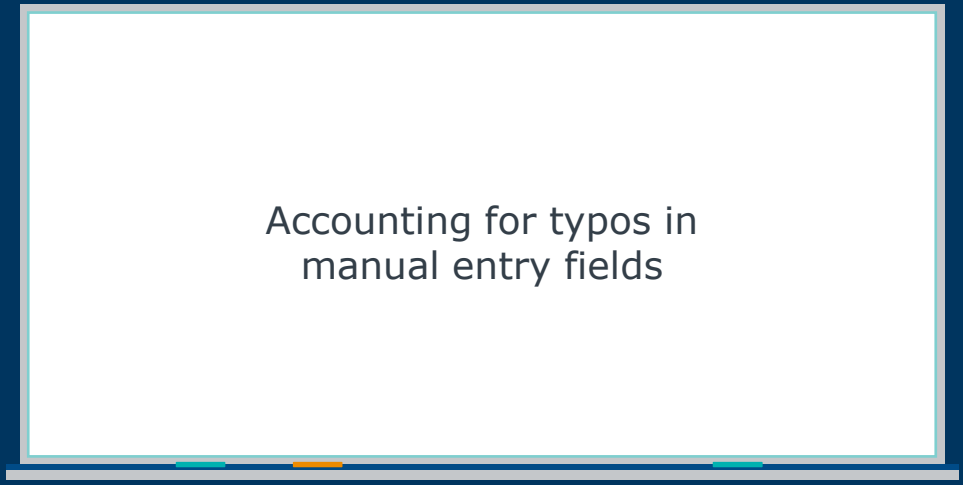
Example: Merging Degree Data

Program
Automotive Technology – AS
Marine Biology – BS
Religious Studies – MA
Psychology - BS

Program
Automotive Technology – A.S.
Marine Biology – B.S.
Religious Studies – M.A.
Psychology – B.S.

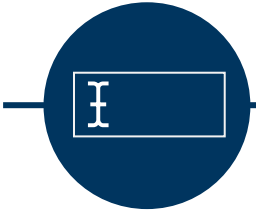
- Replacing the entire string in this case would work, but be sub-optimal
- To replace "Marine Biology – BS" and "Psychology – BS" with "Marine Biology – B.S." and "Psychology – B.S.", you'd need an operation for each *entire string*.
- Instead, use "Replace Sub-String" to just replace all instances of "BS" with "B.S.", "AS" with "A.S.", and so forth

Challenge 3



How Typos End Up in Data

Two Conditions Must Be Met



1

Manual Entry Fields

Naturally, *unexpected* typos only occur when data entry occurs through manual entry. Surveys and administrative data entry without validation are frequent causes.

2

A Known Set of Possibilities

You can't systematically diagnose text as having typos without a known set of "correct" values. These valid values become the foundation of corrective rules in your data preparation.

Common Solutions

Local Solutions:

- Use a Cleanse node and build replacement rules
  - For names, departments, and other entities with official names, use text replacements
  - For acronyms, a sub-string replacement may be worth considering
- Merging with a Lookup Table is a more robust resource that can be modified and maintained easily

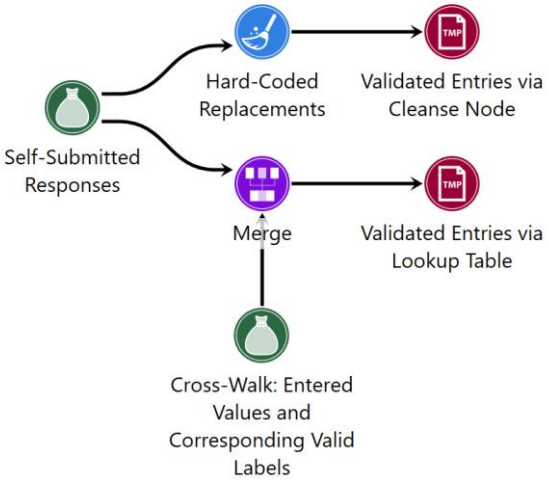
Systematic Solution:

- If possible, reduce the number of manual entry fields
- Introduce validation rules to precent typos from cascading into operational data

Example: A Manually Entered "Role"

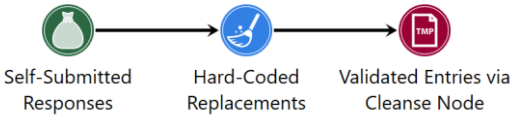
What is your role at the institution?
Student
Student
Studnet
Faculty
Daculty
Stfaf
Staff

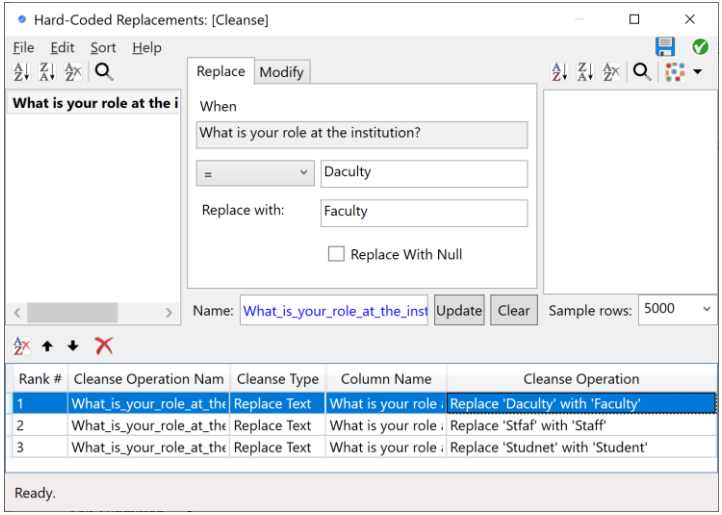
Known Valid Entries
Student
Faculty
Staff



Solving the Issue Using the Cleanse

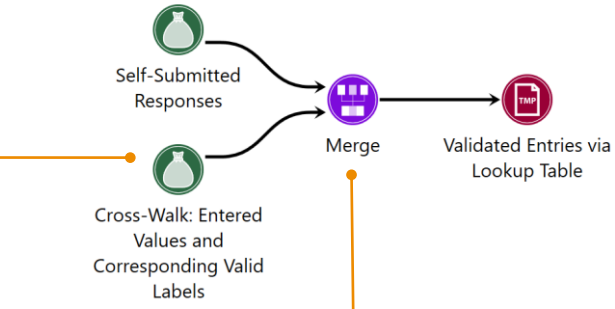
Manual Entry	Validated
Student	Student
Student	Student
Studnet	Student
Faculty	Faculty
Daculty	Faculty
Stfaf	Staff
Staff	Staff

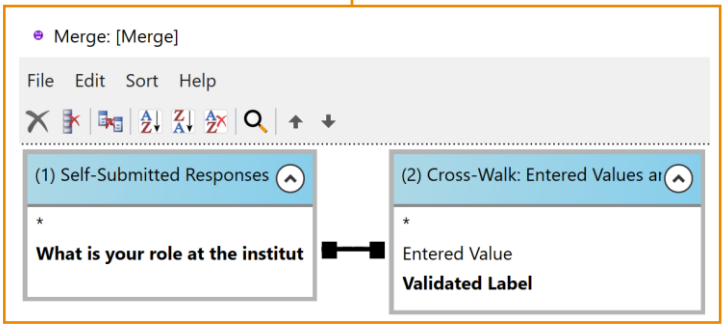




Solving the Issue Using a Merge and a Crosswalk

Entered Value	Validated Label
Student	Student
Student	Student
Studnt	Student
Stundet	Student
Faculty	Faculty
Daculty	Faculty
Favulty	Faculty
Facluty	Faculty
Staff	Staff
Stff	Staff
Staf	Staff
Stfaf	Staff





1) "Crosswalk" is a common term for a "reference table", "lookup table", and perhaps other terms. It provides a mapping from one value to a corresponding one, as in the case of a numeric code corresponding to a department name.

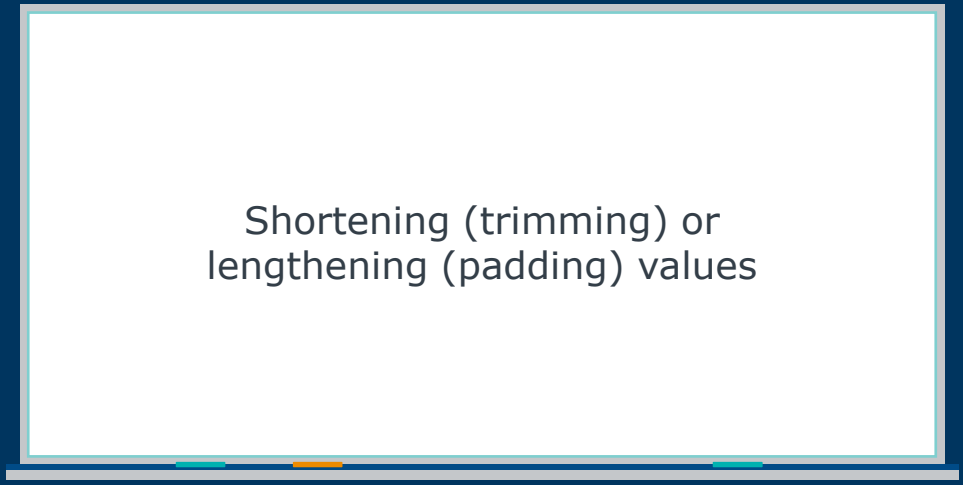
Poll

Which approach would you choose?

a) Use the Cleanse node to find and replace values

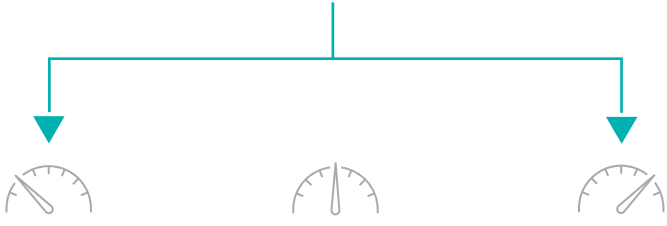
b) Use a Merge node to cross reference a lookup table

Challenge 4



Trimming or Padding Values

Values that are not the right length



Too Short

9 out of 10 times, this means that leading zeroes got dropped by some data-read or formatting process

Too Long

Either because A) only the first portion is valuable, or B) because the value is formatted to include blank characters ("whitespace") out to a certain length

Common Solutions for Field Length

Fixing values that are not the right length

Too Short: Transform

The Transform's "PADLEFT()" and "PADRIGHT()" functions allow users to lengthen a field and impute specific values in case the original text is not long enough.

Too Long: Cleanse

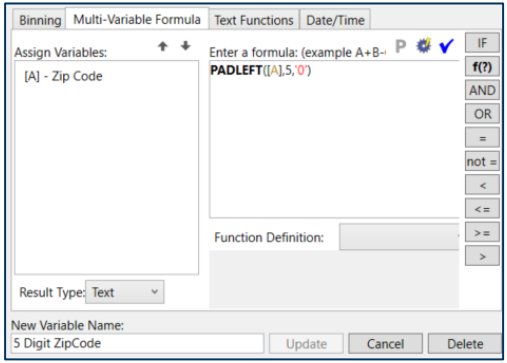
The "trim" functionality in a Cleanse node removes all leading and trailing whitespace in a field.

Interestingly, the Transform *does* have the "TRIM()" and the "LEFT()" functions, which can help with trimming down fields too.

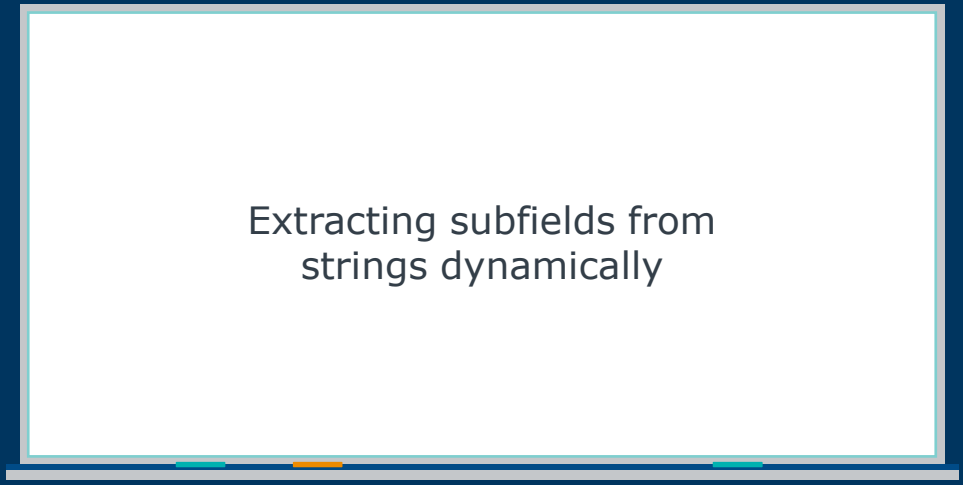
Padding Truncated Zip Codes

Obs #	City	State	Zip Code
1	Madison	NH	3849
2	Madison	WI	53558
3	Madison	CT	6443

5 Digit ZipCode
03849
53558
06443



Challenge 5



Extracting Subfields Conditionally

Subfields not Conforming to Constant "Length" or "Order" Assumptions

- This general requirement can get *very* complex in some cases
  - These are cases where you can't count on the target sub-string being the "first", or any specific position as a rule
  - You also can't count on the target sub-string beginning at a certain character position (e.g., "the 8th through 14th characters")
- You'll need to find some pattern, even if it's a complex one

47419,50544,0
67306, 69092
47419, 80479, 22583, 129084, 60802

This is an excellent case, where just the highlighted element should be returned from each row's value.

Go-To Tools for Dynamic Subfields

Functions	Description	Frequency of Use
SUBFIELD()	Returns the *n*th subfield from within a string as delimited by a specified character.	Common
SUBFIELDCOUNT()	Returns the number of subfields found within a string as delimited by a specified character.	Uncommon
CHARINDEX()	Returns the starting position of a specified expression within a string- and a "0" if not found.	Uncommon
MATCHESREGEX()	Allows the user to stipulate a "regular expression" and tests whether the string matches that set of rules.	Rare
LEFT()/RIGHT()	Returns the specified number of characters, starting from either the left or the right.	Common
REMOVELEFT()/REMOVERIGHT()	Removes the specified number of characters from a string, starting from either the left or the right.	Uncommon

Example: Returning the Last, Non-Zero Code

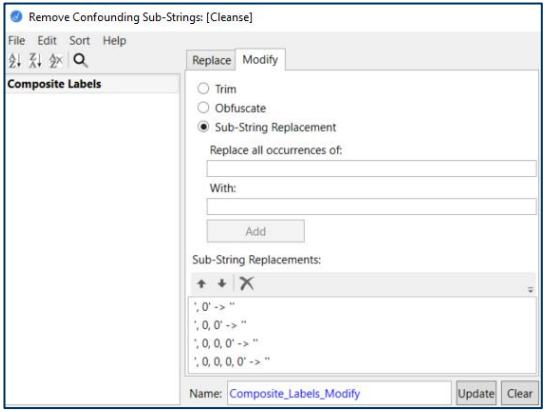
Target Outcome:

Systematically creating the column on the right

Obs #	Composite Labels	Last Label in String
1	47419, 50544, 0	50544
2	67306, 69092	69092
3	67306, 69092	69092
4	67306, 69092	69092
5	67306, 69092	69092
6	67306, 69092	69092
7	67306, 69092	69092
8	67306, 69092	69092
9	67306, 69092	69092
10	35977, 69092, 0	69092
11	35977, 69092, 0	69092
12	35977, 69092, 0	69092
13	35977, 69092, 0	69092
14	35977, 69092, 0	69092
15	35977, 69092, 0	69092
16	35977, 69092, 0	69092
17	47419, 80479, 22583, 129084, 50186, 46092, 60802	60802
18	47419, 80479, 22583, 129084, 50186, 46092, 60802	60802
19	61387, 0, 0, 0	61387
20	61387, 0, 0, 0, 0	61387

Step 1

Use a sub-string replacement operation to eliminate the undesirable values from each string



Example Continued

Returning the Last, Non-Zero Code

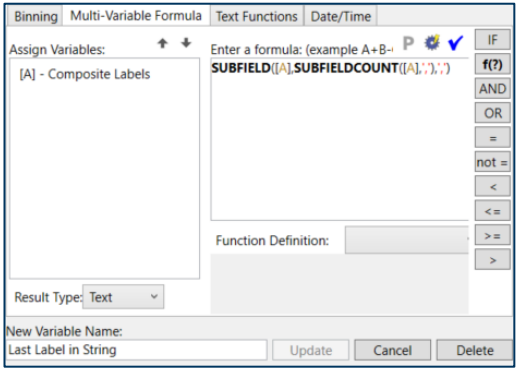
Target Outcome:

Systematically creating the column on the right

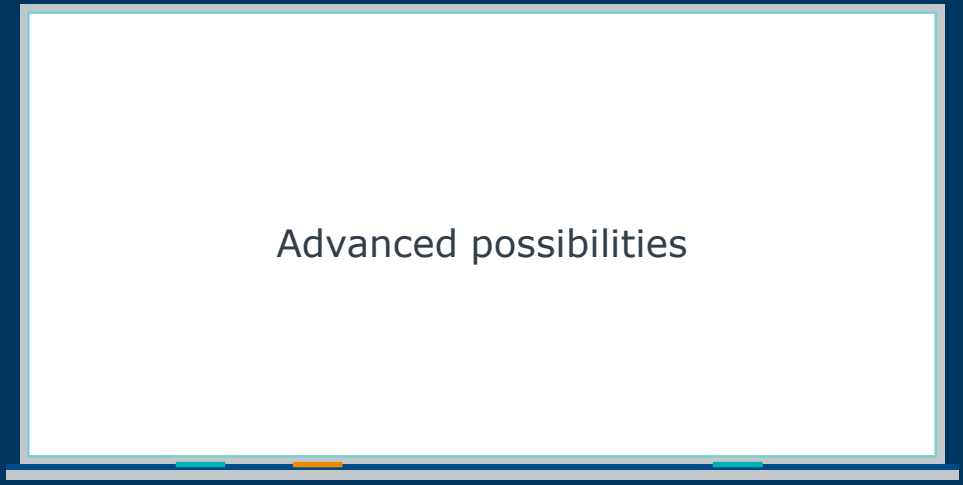
Obs #	Composite Labels	Last Label in String
1	47419, 50544, 0	50544
2	67306, 69092	69092
3	67306, 69092	69092
4	67306, 69092	69092
5	67306, 69092	69092
6	67306, 69092	69092
7	67306, 69092	69092
8	67306, 69092	69092
9	67306, 69092	69092
10	35977, 69092, 0	69092
11	35977, 69092, 0	69092
12	35977, 69092, 0	69092
13	35977, 69092, 0	69092
14	35977, 69092, 0	69092
15	35977, 69092, 0	69092
16	35977, 69092, 0	69092
17	47419, 80479, 22583, 129084, 50186, 46092, 60802	60802
18	47419, 80479, 22583, 129084, 50186, 46092, 60802	60802
19	61387, 0, 0, 0	61387
20	61387, 0, 0, 0, 0	61387

Step 2

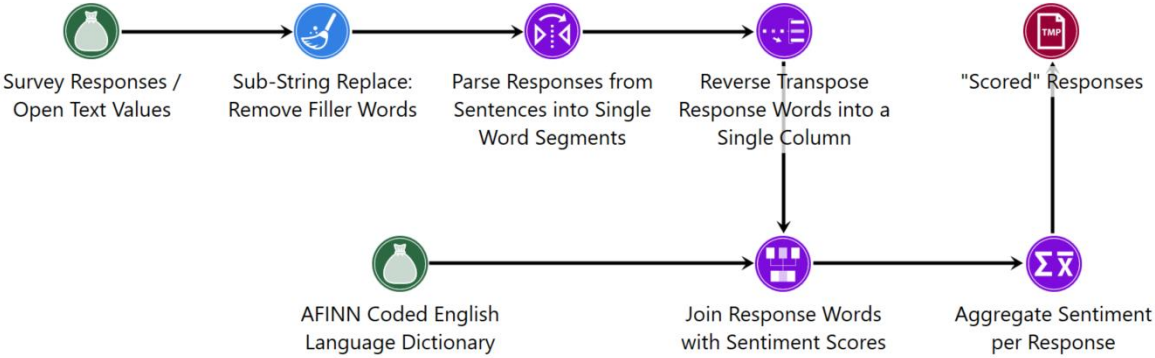
- Use a Transform to count the number of subfields left, and return the last subfield from each value
- SUBFIELD([A],SUBFIELDCOUNT([A],','), ',')
  - SUBFIELDCOUNT([A],',') returns the number of fields each row has
  - SUBFIELD([A], …,',') uses that result to return the last subfield's text for each row



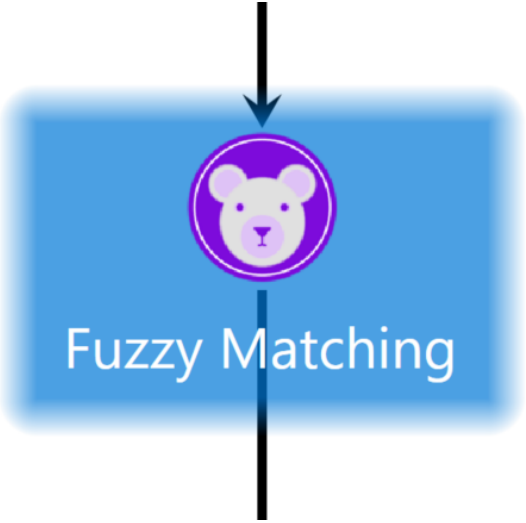
Bonus!

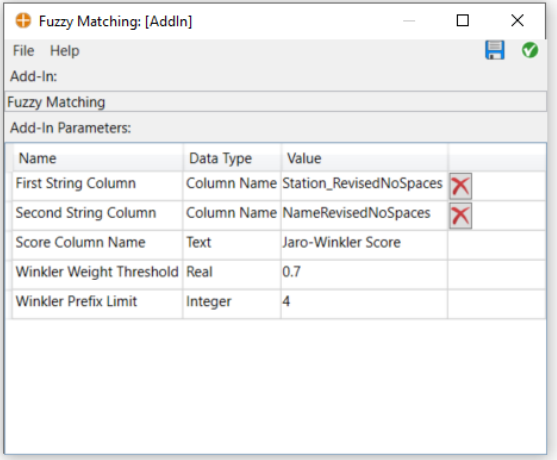


Sentiment Analysis

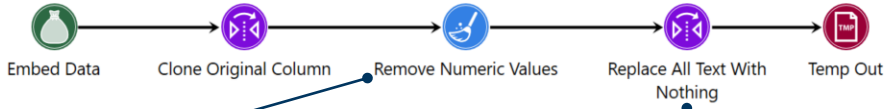


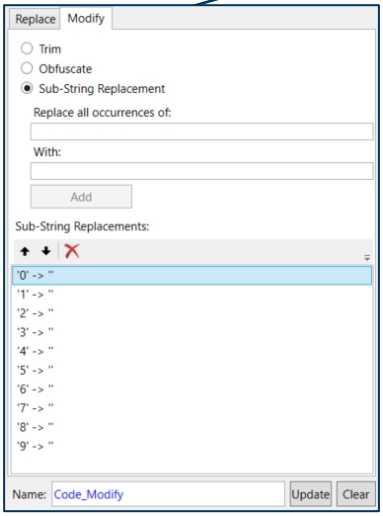
Fuzzy Matching

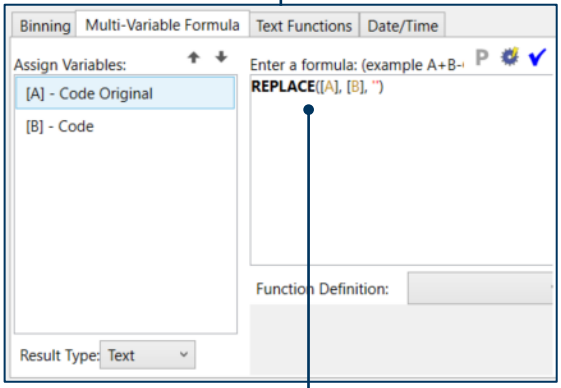




Returning Numeric Fields Only







REPLACE() returns a text field where all occurrences of a specified expression are replaced with a new expression

Poll

Do you feel more comfortable with text data in Construct now?

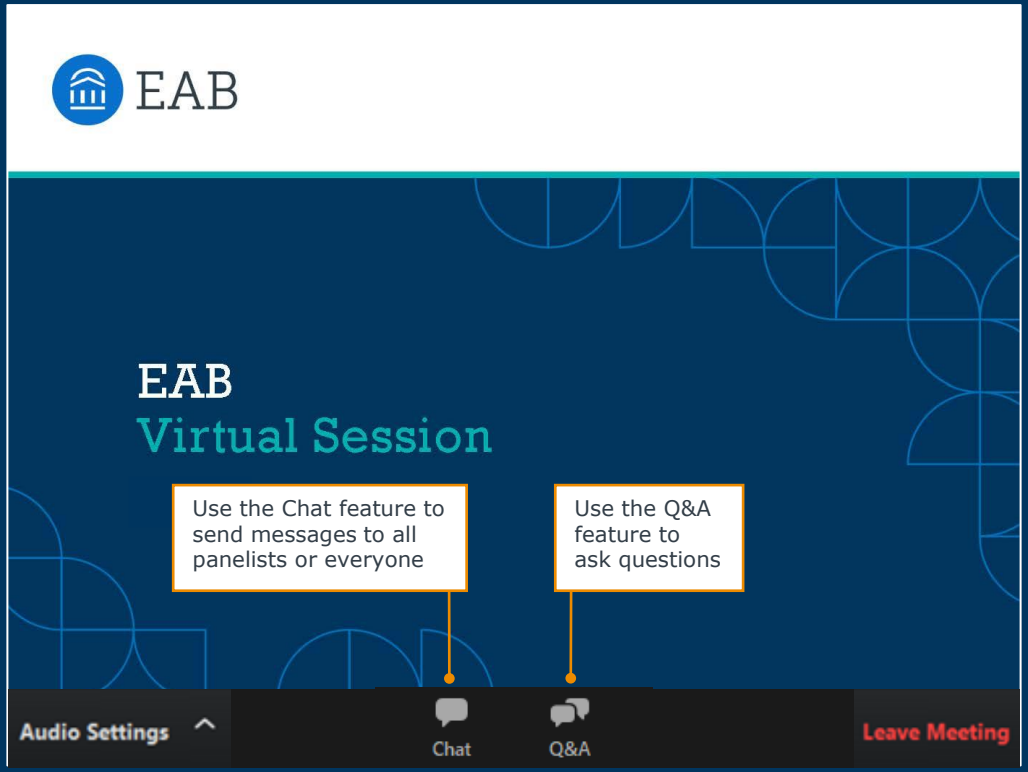
a) Yes, definitely more comfortable than before

b) I feel the same level of confidence as before

c) I am more confused now than I was before

1. Rapid Insight and EAB
2. Overview of the Presentation Structure
3. Text Parsing Challenges
4. **Audience Q & A**

Submit a Question or Comment



Quick Poll

How was today's session?

Please take a few minutes to complete the survey to provide additional feedback! A link will be placed in the Chat and you'll receive a follow-up email.

Washington DC | Richmond | Birmingham | Minneapolis

202-747-1000 | **eab.com**

@eab **@eab_** **@WeAreEAB** **@eab.life**