First step was to Sign up/Activate AI Builder

The output for the following software agreements were tested utilizing the steps outlined in this document:

* ADS (Keysight)
* Anaconda2
* Ansys
* Ansys Electromagnetic
* Ansys Mechanical
* ANSYS/Ansoft AnsysEM HFSS
* AnyLogic PLE
* AutoCAD Electrical
* Autodesk 3ds Max
* Autodesk AutoCAD
* Autodesk AutoCAD 2019
* Autodesk AutoCAD Mechanical
* Autodesk Civil 3D
* Autodesk Inventor
* Autodesk MotionBuilder
* Autodesk Mudbox
* Autodesk Navisworks Manage
* Autodesk Revit
* Bluebeam Revu
* Comsol 5.0
* Comsol 5.3a
* Comsol 5.5
* Maple
* Solid Works
* Synopsys - CIDSE Only
* Synopsys - EEE HEP - Higher Education Program

NOTE: Blanket license agreements may have been utilized to cover all software packages from a single developer.

## Obtain text from PDF document using text recognition

* 1. Create new flow (Text Recognition)
     1. Manually trigger flow with file input
     2. Add “Predict” action using Text recognition model and input image defined in manual trigger.
     3. Instantiate string variable for text (This variable will store all recognized document text.)

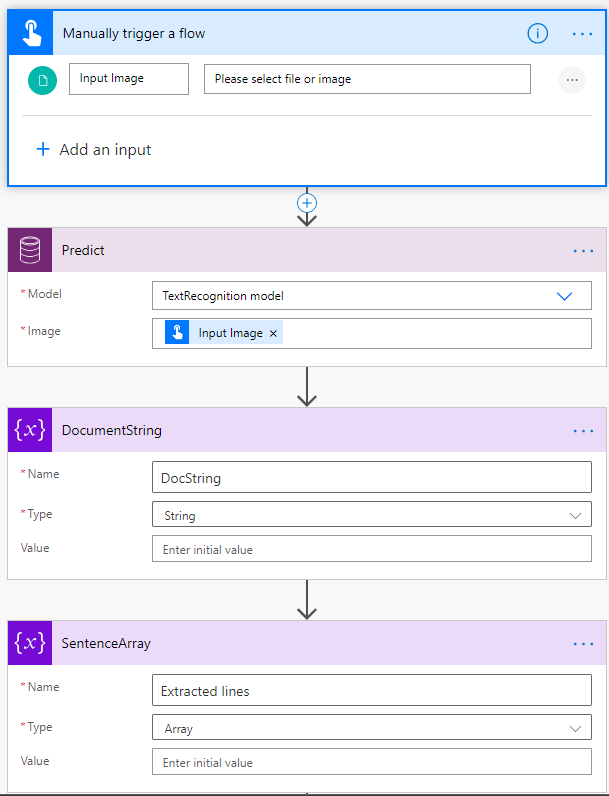


Figure : Steps 1 - 3

* + 1. Add apply to each action using “results” from previous steps.
       1. Add apply to each action using “lines” as the input, then use the variable “append to string” action to append all lines.
    2. Output to file or use scope to view variable in flow. (example text below)

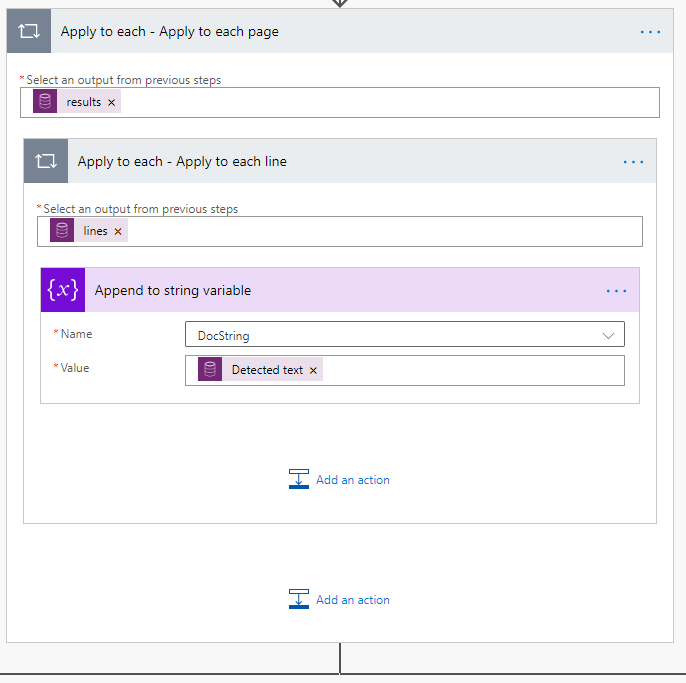


Figure 2: Step 4

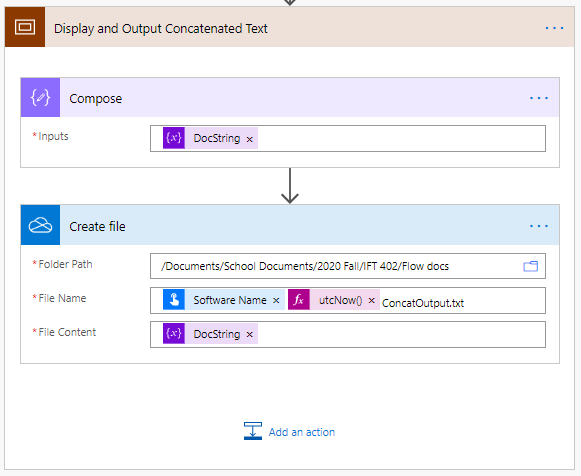


Figure : Step 5

* 1. Output

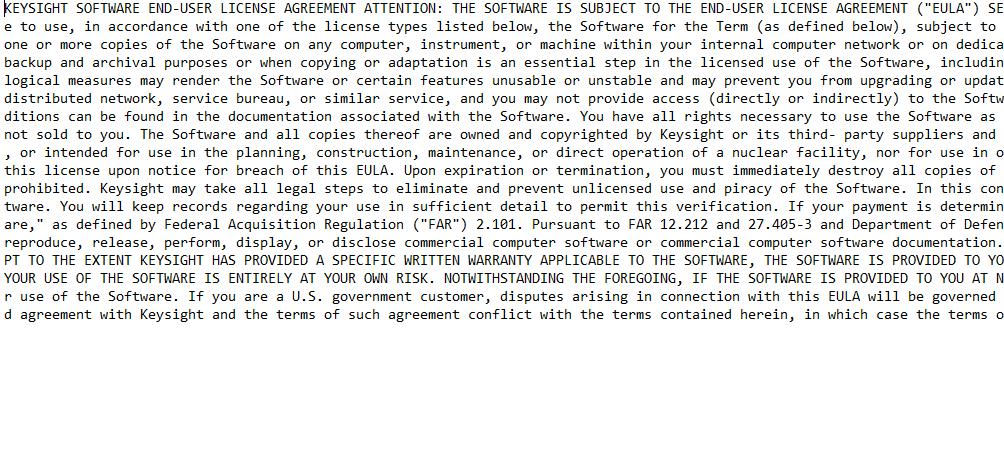


Figure : Recognized text output

## Splitting into sentences for analysis and Output

1. Split string into sentences
   1. The DocString varable is split into SentenceArray by using “split(variables('DocString'),'. ')”

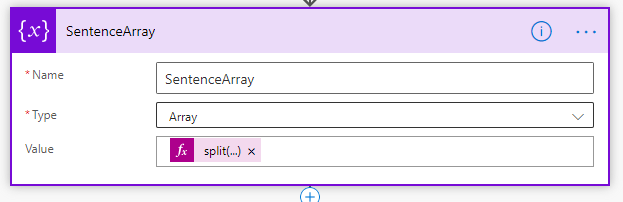


Figure : Step 1

* 1. A new variable SplitSentenceString is created to append sentences
  2. Each element is appended to SplitSentenceString with a carriage return
  3. The populated SplitSentenceString is Outputted via OneDrive and scope for viewing in power automate.

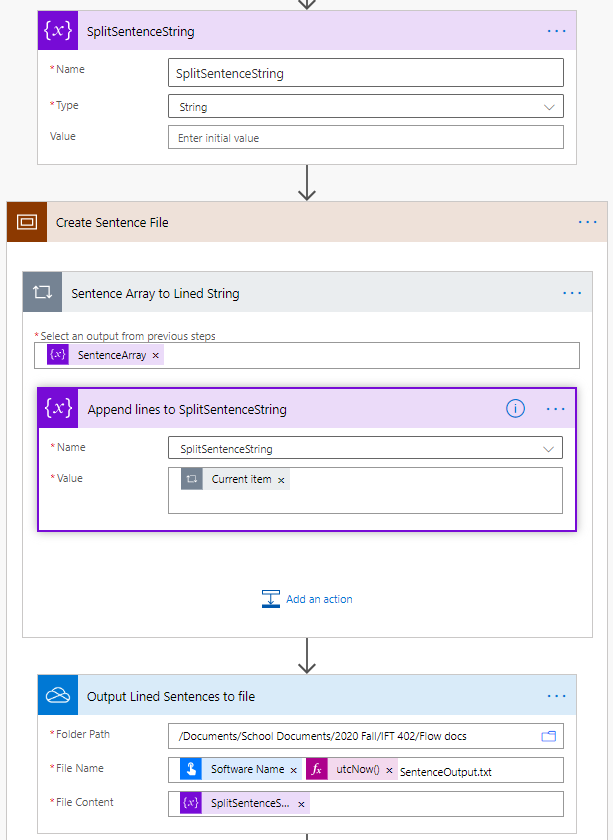


Figure : Steps 2 - 4

NOTE: Sentences split by delimiter causing inconsistent sentences. No other method has been discovered to tokenize in Microsoft Flow.

## Key Phrase Extraction

1. Add key phrase to define each sentence
2. Apply key phrase recognition
3. Append sentence to the KeyPhraseSentenceString

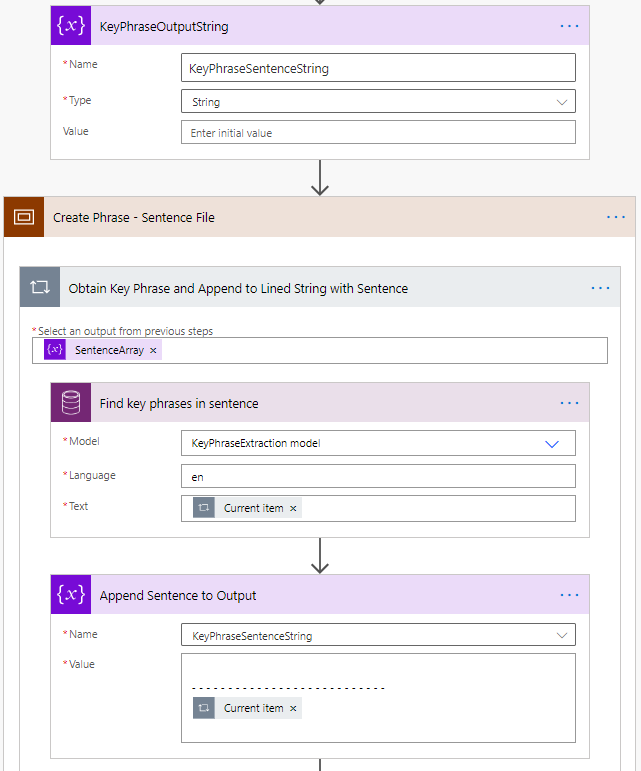


Figure : Steps 1 - 2

1. Append each key phrase to the KeyPhraseSentenceString
2. Output to file

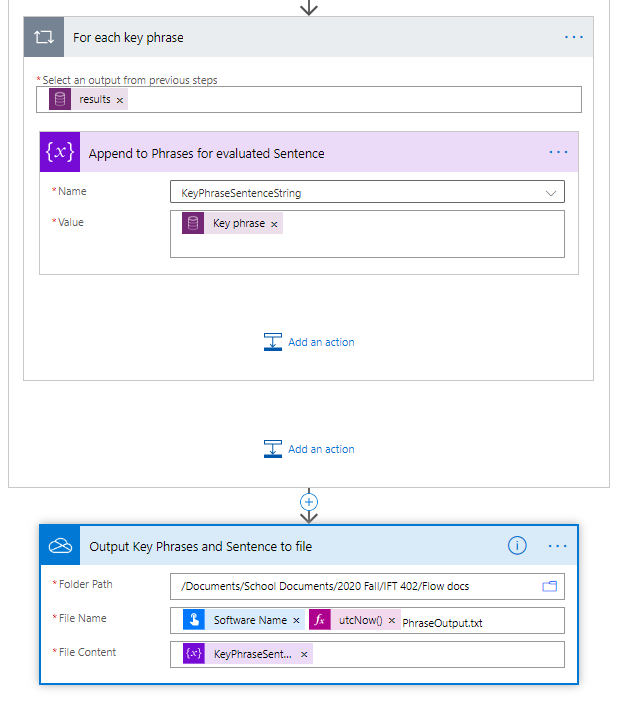
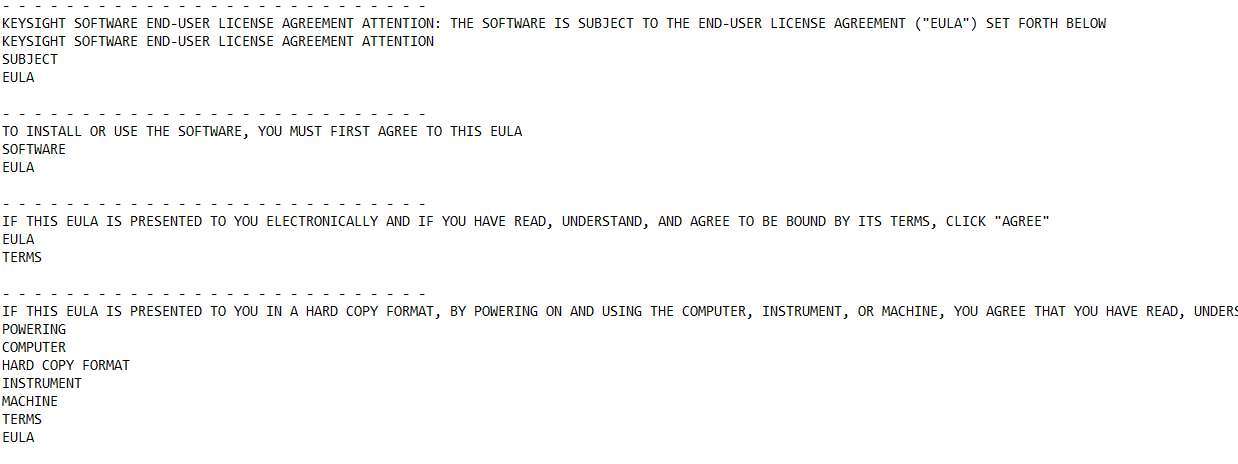


Figure : Steps 3 - 4

NOTE: Errors occur with strings that exceed 64 characters (e.g. URL) which prevent file processing.

* 1. Example Output



## Entity Extraction

1. Entity extraction is almost identical to the key phrase extraction process. Two arrays were defined to extract entities that were identified as an “Organization” or “URL”.
   * 1. Create variable to store entities and sentences
     2. Create arrays to store identified entites
     3. Use scope to contain the processes (optional)

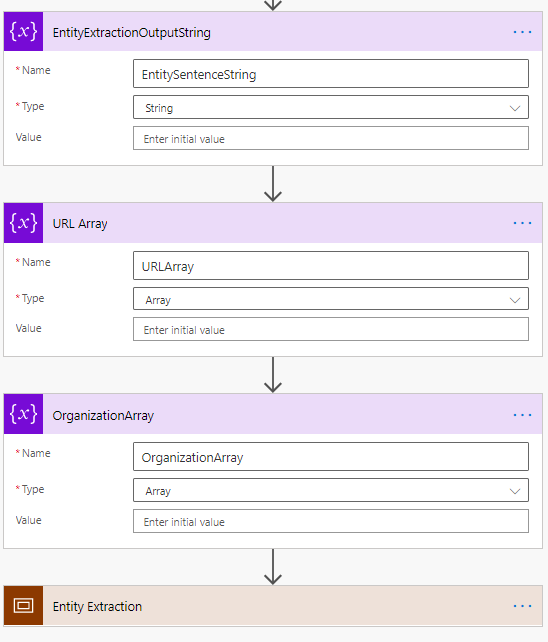
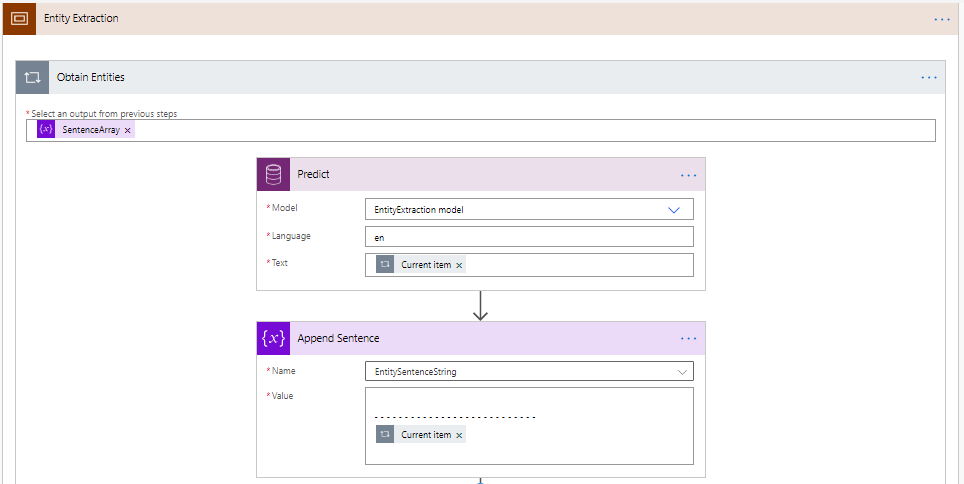
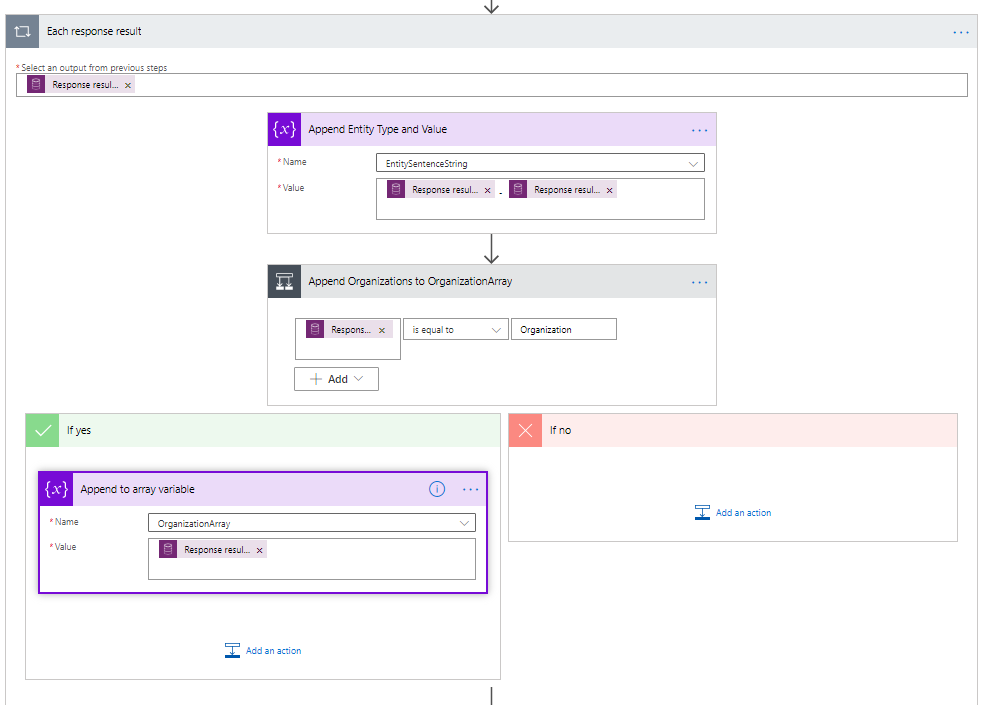


Figure : Steps 1 - 3

* + 1. Obtain entities by using the SentenceArray output
    2. Run the prediction model for entity extraction using “current item” in the text field



* + 1. Obtain the entity type for each response result
    2. Perform append to EntitySentenceString
    3. If the response type is organization, append the value to the OrganizationArray



* + 1. If the response type is URL, append to the URL Array
    2. Output EntitySentenceString to file
    3. Compose the view organizations and URL array for viewing in the flow (see next page)

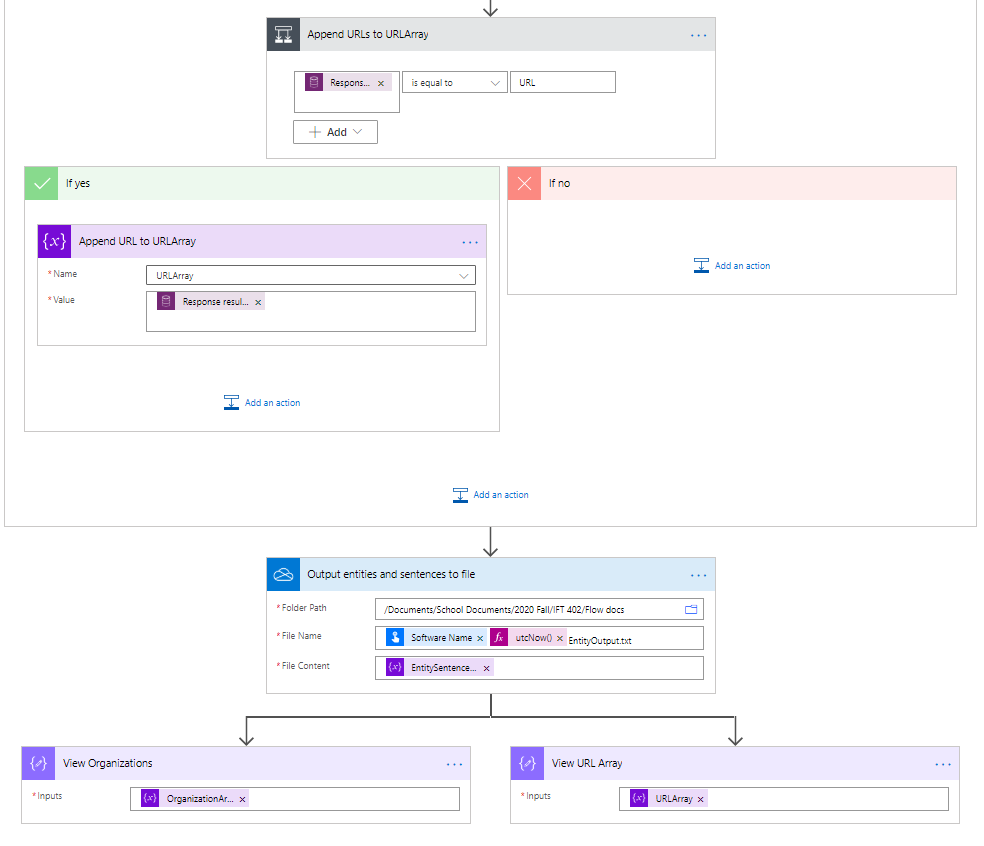
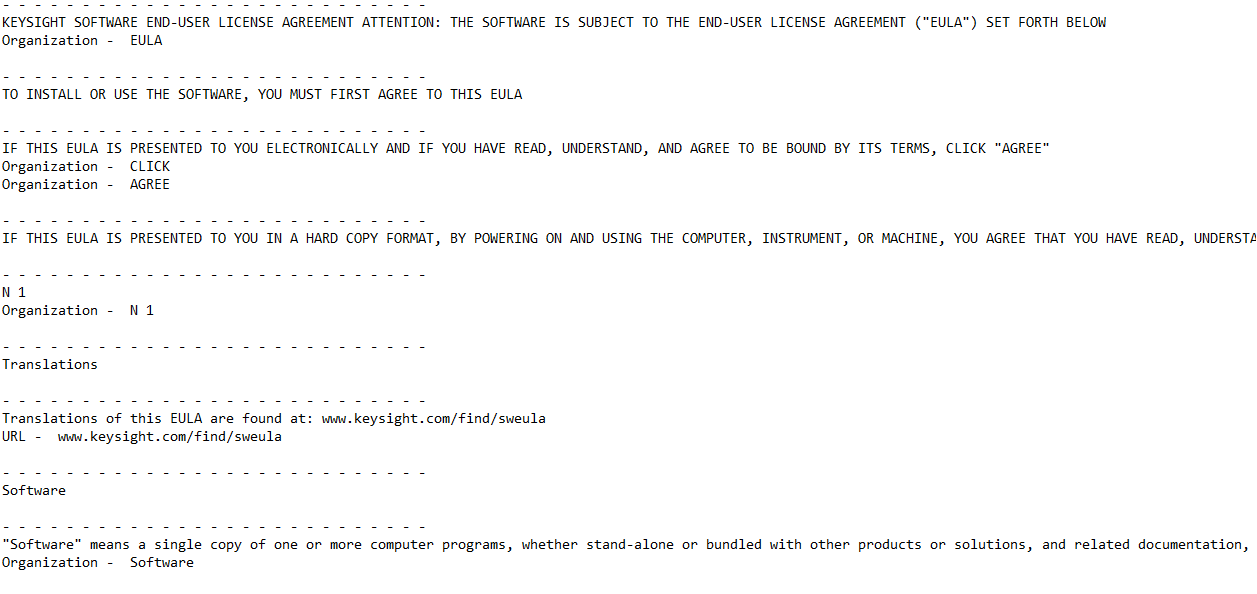
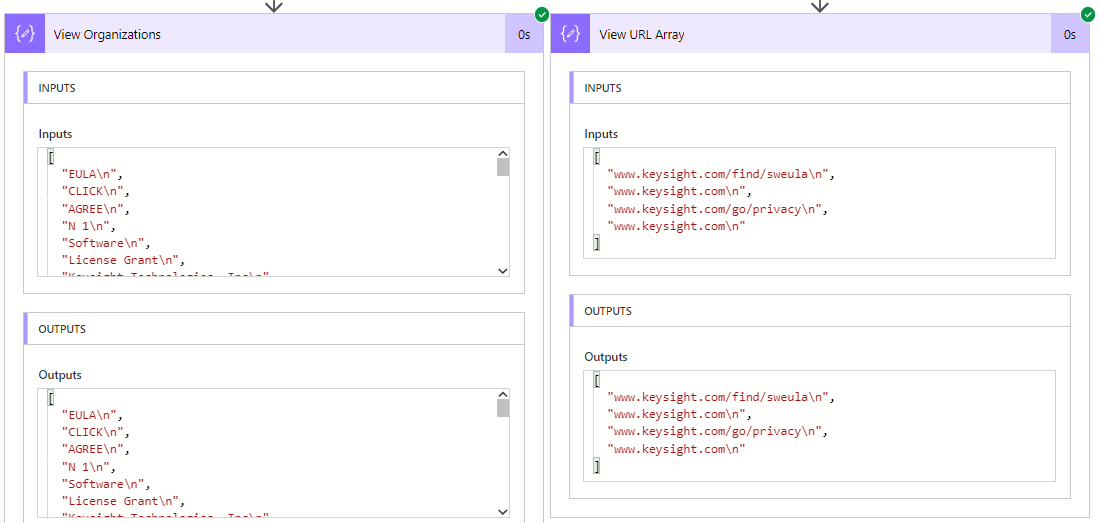


Figure : Steps 4 - 11

1. File output example



1. Flow array outputs



## Performance

Text recognition accurately converts images to text, however, the organization of the text can cause problems. There were other anomalies where the text was out of order and missing characters as shown below. Scrambling the sentences opens the possibility of the string being improperly interpreted.



Figure : Original document text



Figure : Recognized text

Documents containing text in multiple columns have the same problem with organization. In addition, documents containing “recognized” words longer than 64 characters caused a flow failure and would not provide any output for the failed action. In this case, the presence of a URL causes the failure.

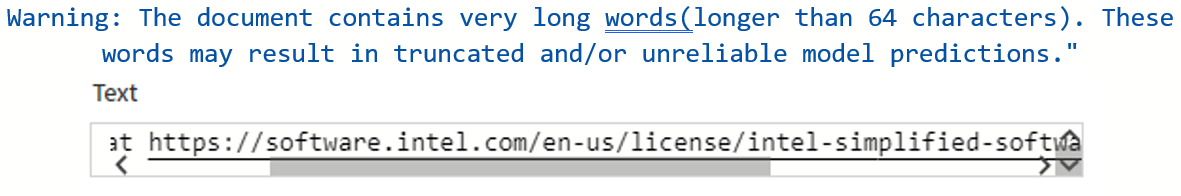


Figure : Long word failure

Processing time for the previous operations ranged from 13 minutes on a single run to 38 minutes during batch processing. The maximum processing time observed during development and testing was 90 minutes.

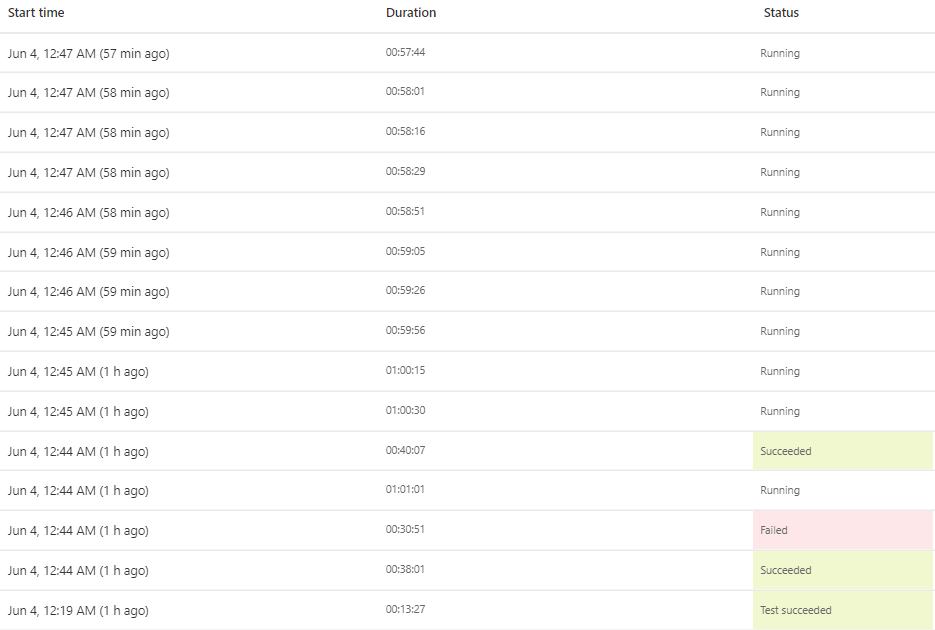


Figure : Processing times

## Next Steps

Build/Train a model – requires multiple examples for each restriction unless another dataset can be found.