



Architect • Consultant • Engineer

ARCHITECT TRAINING MANUAL VOL.2

ADVANCED INVOICE PROCESSING

13900 N. Harvey Ave • Edmond, OK • 73013 • 405-507-7000

www.grooper.com

TABLE OF CONTENTS

Phase 1 – Acquire	3
Object Portability in GROOPER	3
Importing ZIP Objects into GROOPER.....	3
Creating a Scanner Profile to Leverage the Source Batch.....	6
Phase 2 – Condition	8
A Focus on OCR	8
Creating and Testing a New OCR Profile	8
Phase 3 – Organize	16
Separation – Change in Value	16
Setting up a Separation Profile to Leverage a Change in Value.....	16
Classification – Positive Extractor on Document Type	20
Setting up Positive Extractors for Document Types.....	20
Phase 4 – Collect	24
Establishing a Data Model and Working with a Data Table	24
Setting up New Data Fields and Data Table	24
Setting up the Table Row Extractor	26
The Grooper Field Class Extractor	35
Setting up a Field Class for the First Time	35
Setting up the Invoice Date Data Field.....	46
The Next Field Class – Building a Feature Extractor.....	47
The Next Field Class – Building a Feature Extractor with FuzzyRegEx	54
The Remaining Field Classes – Lexicons, Cheat Codes, and Arrays	65
Building a Feature List Lexicon	65
Create the Data Type that will Leverage the Field Labels Lexicon.....	71
Tackling Multi-Line Field Labels with Ordered Arrays.....	79
The Next Field Class – Using the Newly Built Field Labels – DT Data Type.....	85
The Next Field Class – Value Extractor Leveraging an Exclusion Extractor	93
The Last Field Class	98
The Remaining Data Fields	101
The Final Required Field – Payment Terms.....	101
Setting up the Freight Data Field	104
Setting up the Discount Data Field.....	109
Setting up the Sales Tax Data Field – Data Element Profile Overrides	113
Final Data Model Adjustment and Review.....	119
Data Model and Data Field Appearance Settings.....	119
Extraction Testing	121
Phase 5 – Deliver	123
Building a Batch Process	123
Creating the Steps of a Process	123
A Final Note	131

Grooper™

In this [Architect Training Manual Vol. 2 • Advanced Invoice Processing](#) guide for **Grooper**, we will flesh out our understanding by continuing to work with a familiar set of documents. This understanding will be achieved by focusing mainly on creating a more complex [Content Model](#) with new [extractors](#) and [data elements](#).

While the approach won't be quite as repeated and linear as what was covered in [Overview and Concepts](#), it is still best practice to think in terms of the [Five Phases](#) of processing documents through **Grooper**.

- [**• Phase 1 - Acquire**](#)
- [**• Phase 2 - Condition**](#)
- [**• Phase 3 - Organize**](#)
- [**• Phase 4 - Collect**](#)
- [**• Phase 5 - Deliver**](#)

It should also be noted that as the document progresses, the pace of the steps provided will advance to prevent repetition and save time and space.

PHASE 1 – ACQUIRE

This process will begin in a similar fashion to [Overview and Concepts](#), by scanning files into **Grooper**. This will be done with a slight twist, however.

OBJECT PORTABILITY IN GROOPER

Follow the below link to obtain a zip file. Extract the zip files from within this file to a directory you have quick access to, as those files will be imported into **Grooper**.

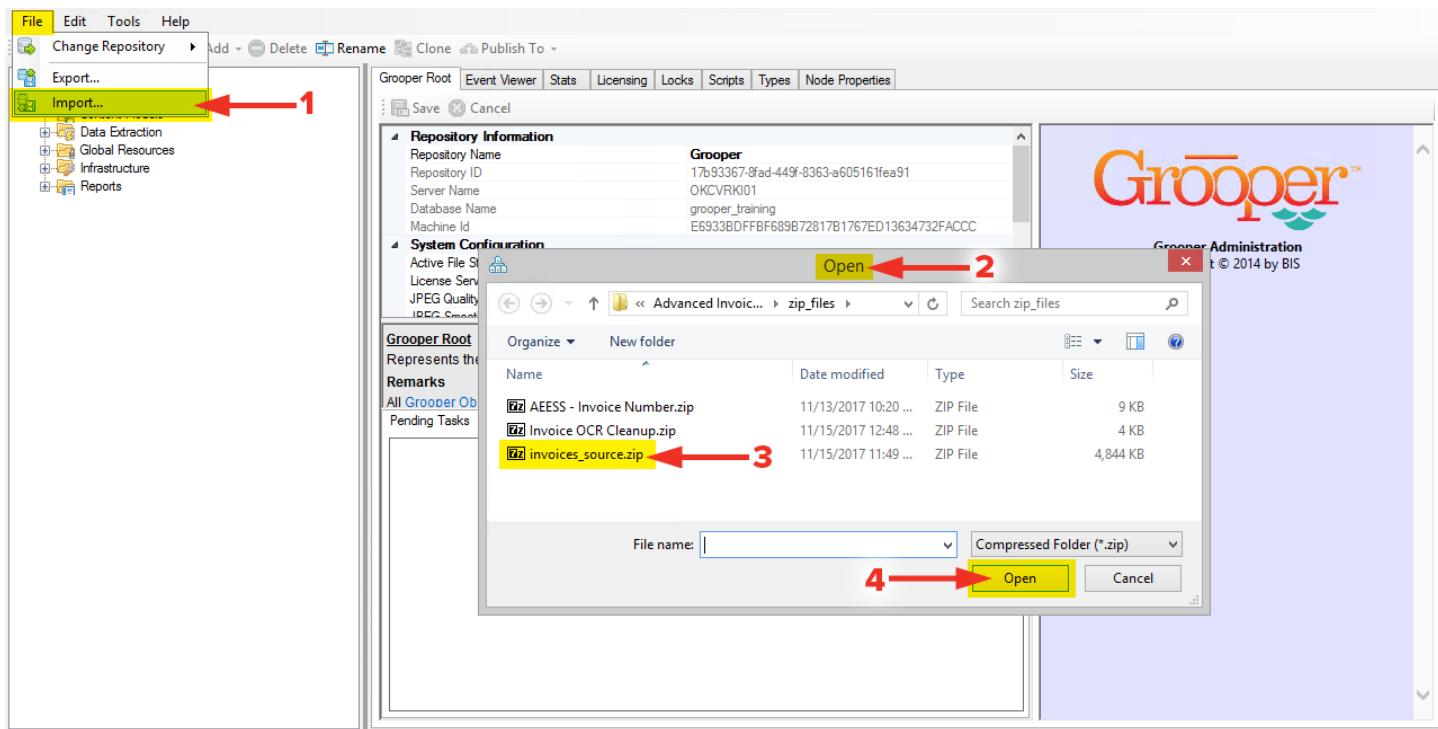
[Grooper A.C.E. - Architect Training Vol.2 - Advanced Invoice Processing - Zip Files](#)

IMPORTING ZIP OBJECTS INTO GROOPER

A key strength of **Grooper** is the flexibility gained by its object-oriented programming. Any object made in **Grooper** can be easily exported as a zip object, and subsequently just as easily imported into another Grooper environment.

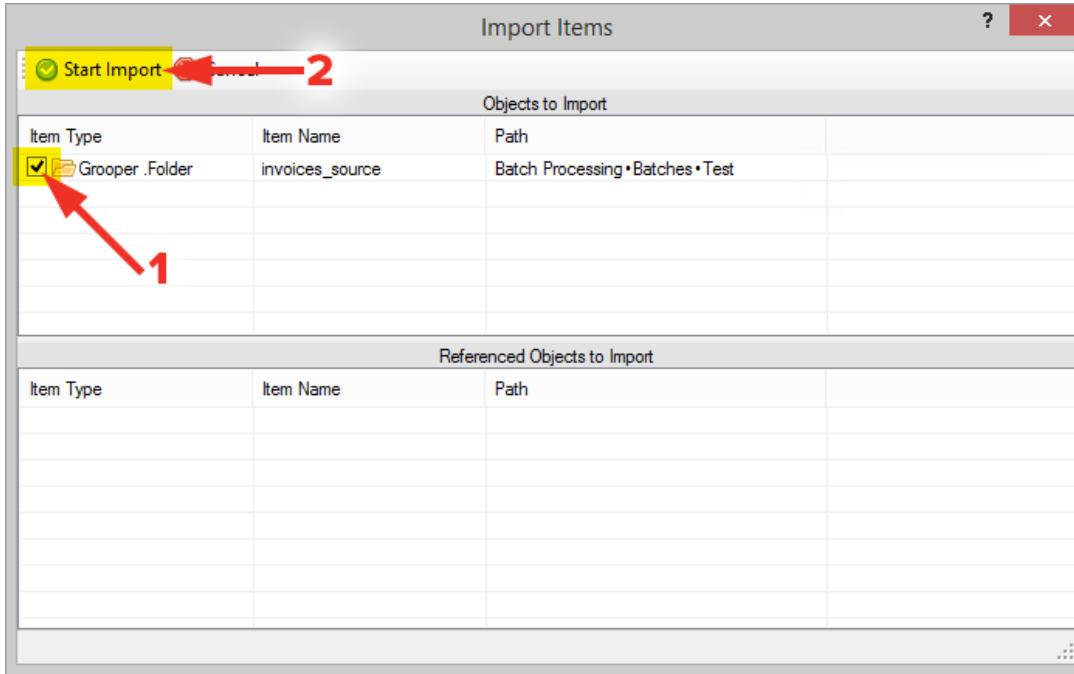
STEP 1 – FILE IMPORT

Start by launching **Grooper Administration** and (1) click **File** and select **Import**. (2) An **Open** window will appear. From here (3) select the **invoices_source.zip** file and (4) click **Open**.



STEP 2 – SELECT IMPORT ITEMS

In the Import Items window that opens, (1) check the object to be imported and (2) click Start Import.



STEP 3 – VIEW NEWLY IMPORT BATCH

Expand the **GROOPER** node tree to (1) Grooper > Batch Processing > Batches > Test > invoices_source and select the invoices batch object. From there, (2) click the Batch Viewer tab and take a moment to (3) look at the page objects within the batch.

The screenshot shows the Grooper interface with the following details:

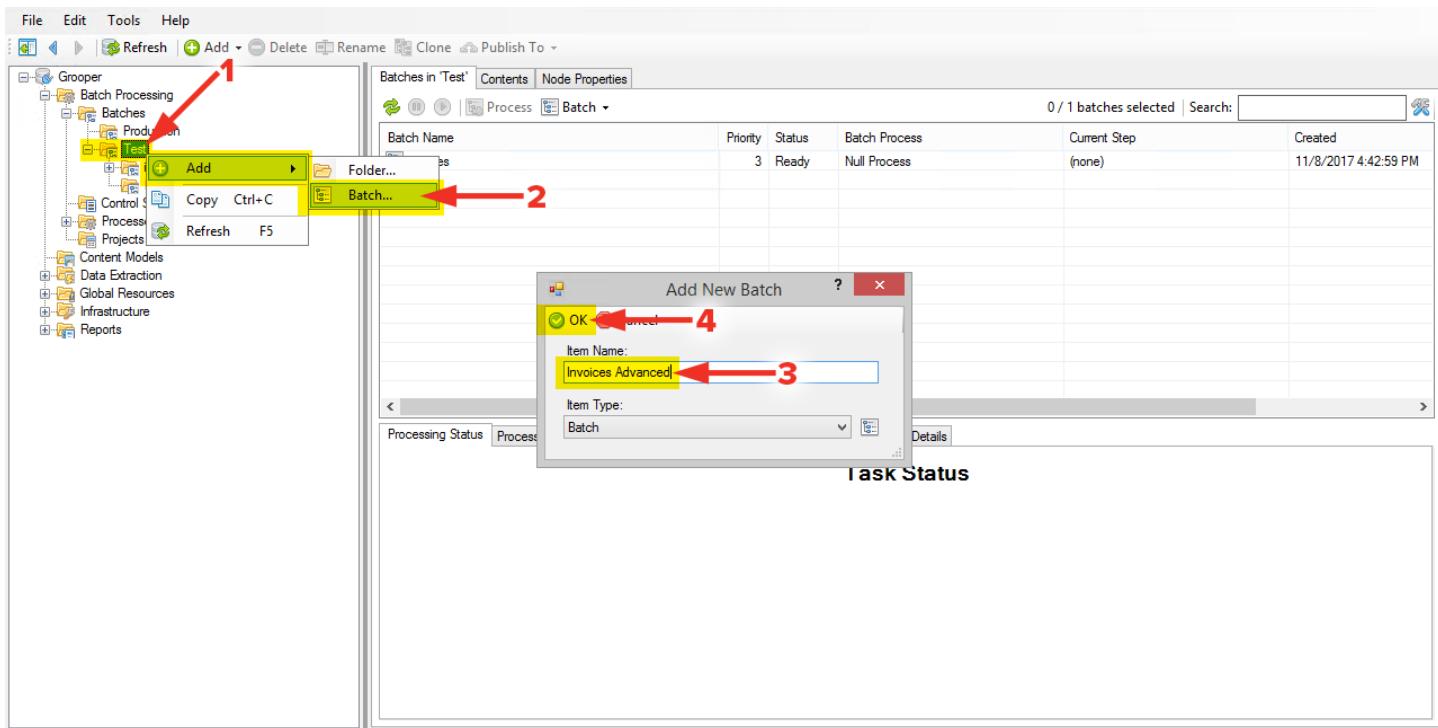
- Node Tree:** Shows the structure under "Grooper" including "Batch Processing", "Batches", "Production", "Test", and "invoices_source". A red arrow labeled "1" points to the "invoices_source" folder.
- Batch Viewer Tab:** The "Batch Viewer" tab is selected, indicated by a red arrow labeled "2".
- Page Objects:** The "invoices" batch contains five page objects labeled "Page 1" through "Page 5". A red arrow labeled "3" points to "Page 1".
- Invoice Preview:** The right panel displays an invoice for "ACME | INTERNATIONAL" with the following details:

ACME INTERNATIONAL	
Ship To:	ACME International, Inc 123 South Main Street Durham, NH 03824 Phone (603) 333-4444
Invoice	Ship To: Grooper Industries 13900 N Harvey Madison, OR 97513 405-507-7000
Customer Reference	Page 1 / 01
P.O. number	11/14/2008
Customer number	74451405
Customer contact	18103943
Currency	USD
Invoice amount	33201729
Payment terms	Delivery number Net 30 days
	Freight carrier Tracking number
- Invoice Details:** The "Invoice details" section shows two items:

Item	Material Description	Quantity	Unit Price	Value
000020	GB.C100003-00001 BRACKET	2 EA	984.53	1,969.06
	Gross Value		12.50- %	246.13-
	Cust. Discount %			
	Net Value for Item		861.47	1,722.93
000021	WS.FREIGHT0231 Freight Part#	1 AU		
	Gross Value		0.00	4.20

STEP 4 – CREATE EMPTY BATCH

The object just imported will be leveraged as a source for page objects, and as such, should not be affected by any processing. An empty batch will be created so testing can be done. (1) Navigate to **Grooper > Batch Processing > Batches > Test**, (2) right click and **Add > Batch**. In the **Add New Batch** window, (3) name the batch **Invoices Advanced** and (4) click **OK**.

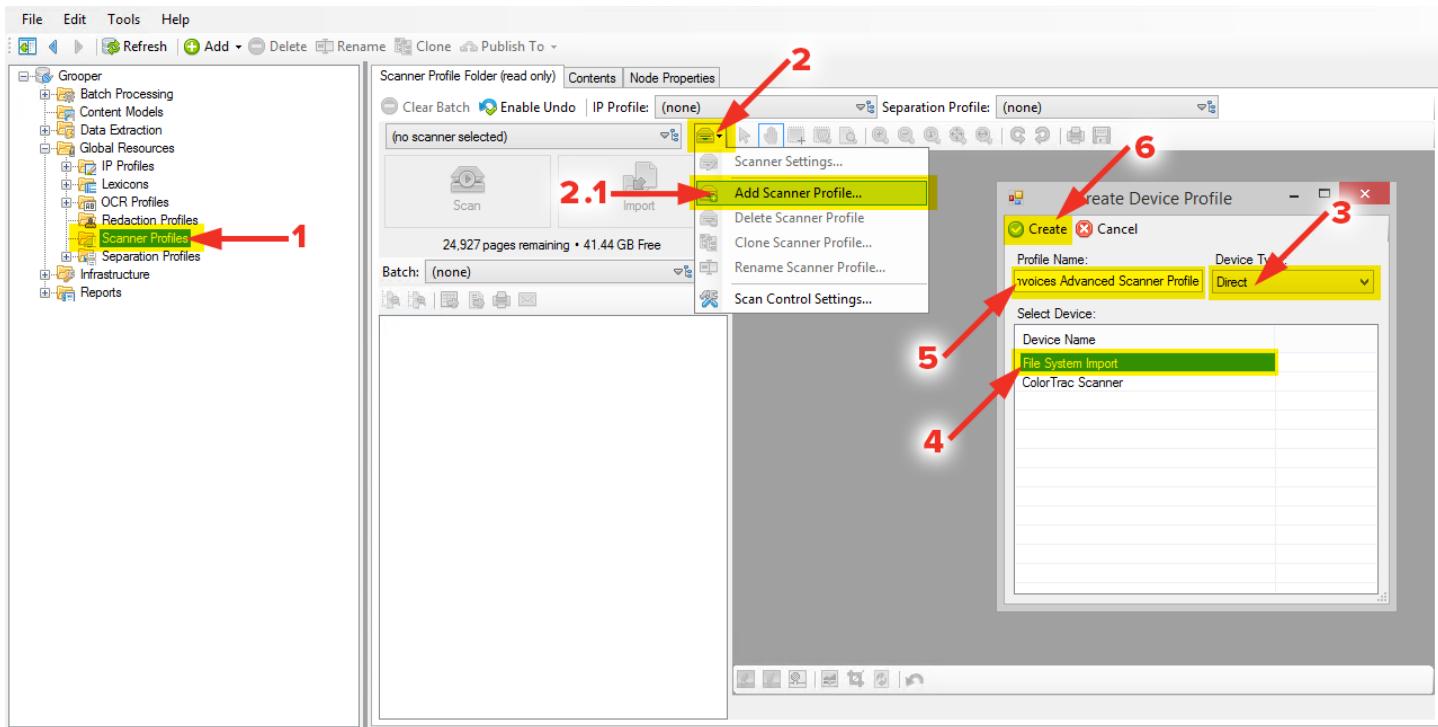


CREATING A SCANNER PROFILE TO LEVERAGE THE SOURCE BATCH

A Scanner Profile will now be created so that the page objects within the source batch can be injected into the newly created empty batch. This is done so that when a **batch process** is created later it can be run as an end to end process starting with **Scan**.

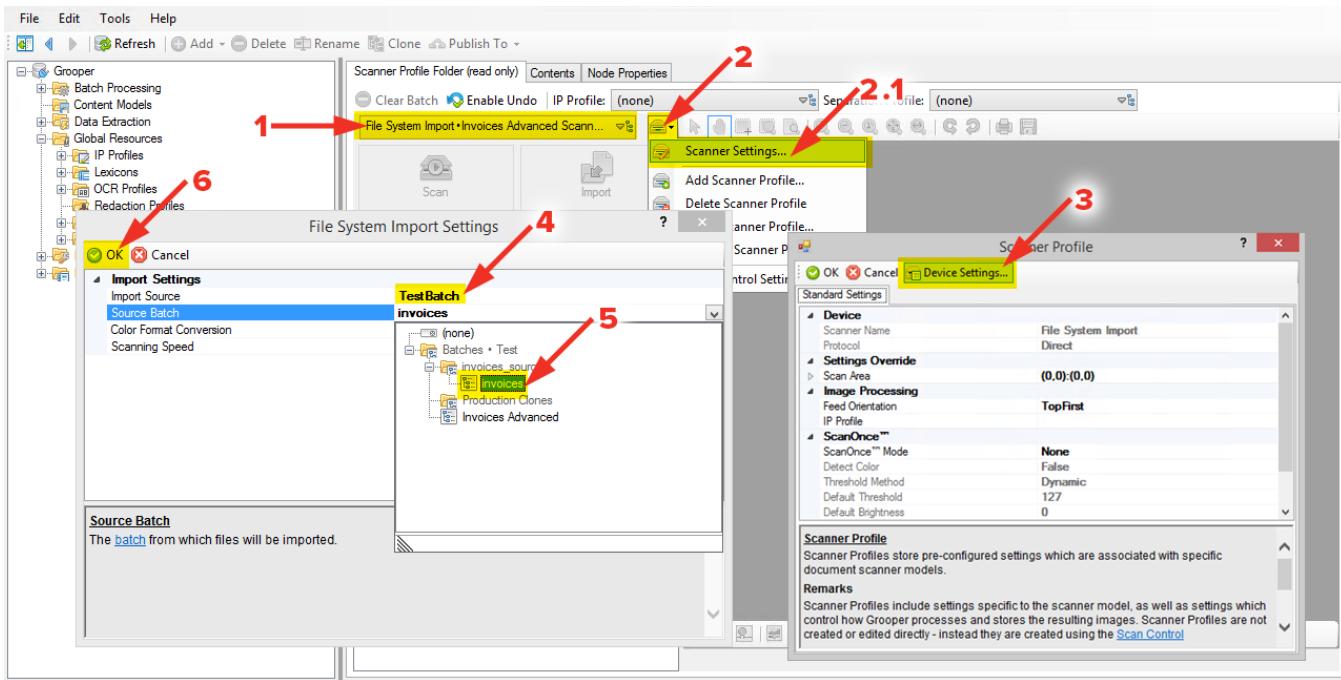
STEP 1 – CREATING A NEW SCANNER PROFILE

(1) Navigate to **Grooper > Global Resources > Scanner Profiles**. (2) Click the Scanner Profile drop-down button and select **Add Scanner Profile...** In the **Create Device Profile** window, (3) select **Direct** for the **Device Type**, and (4) select **File System Import** for the **Device Name**. (5) Name the profile **Invoices Advanced Scanner Profile** and (6) click **Create**.



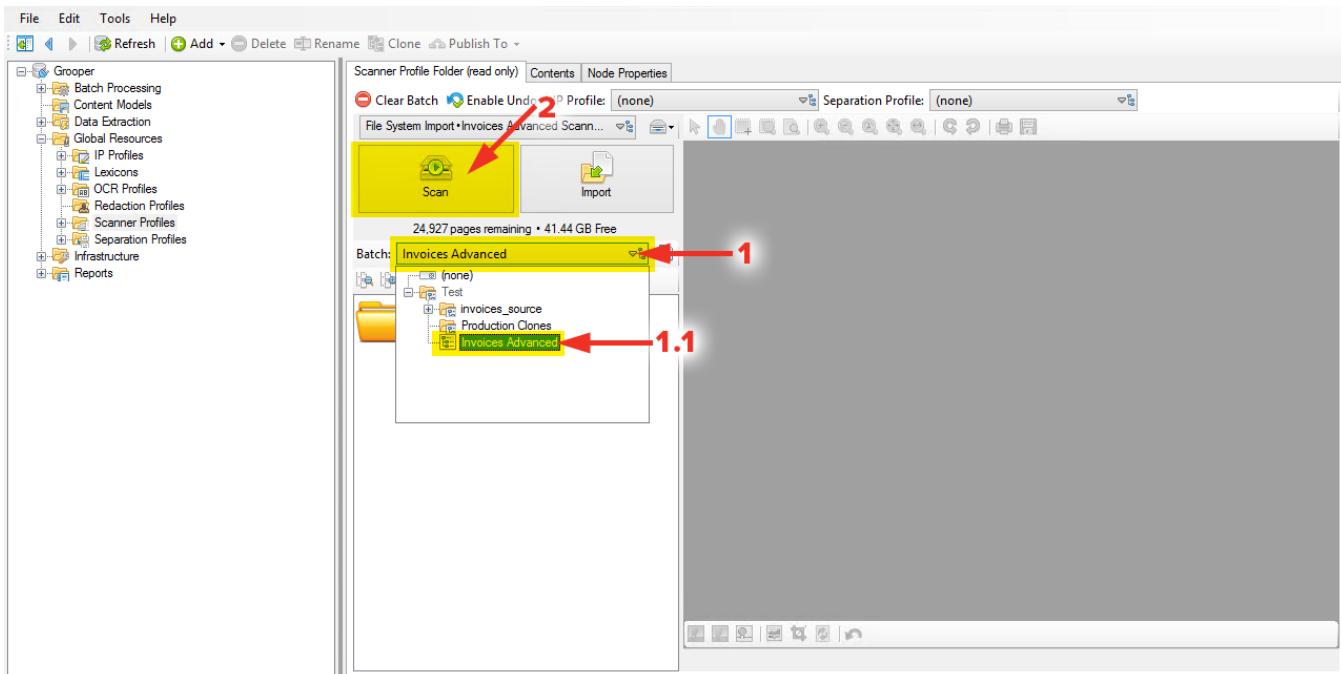
STEP 2 – SCANNER PROFILE SETTINGS

(1) With the newly created Scanner Profile selected in the drop-down, (2) click the Scanner Profile drop-down button and select Scanner Settings. In the Scanner Profile window that opens, (3) click Device Settings... From the File System Import Settings window, (4) set Import Source to TestBatch, and (5) set Source Batch to the Invoices batch object from within the invoices_source area, then (6) click OK to close both open windows.



STEP 3 – PERFORM SCAN

(1) From the Batch: drop-down select the **Invoices Advanced** batch object and (2) click the **Scan** button. The page objects from the source batch will be scanned into the **Invoices Advanced** batch.



PHASE 2 – CONDITION

The second phase in **Grooper** is just as critical as before as all information that will be worked with moving forward will come from conditioning the documents.

A FOCUS ON OCR

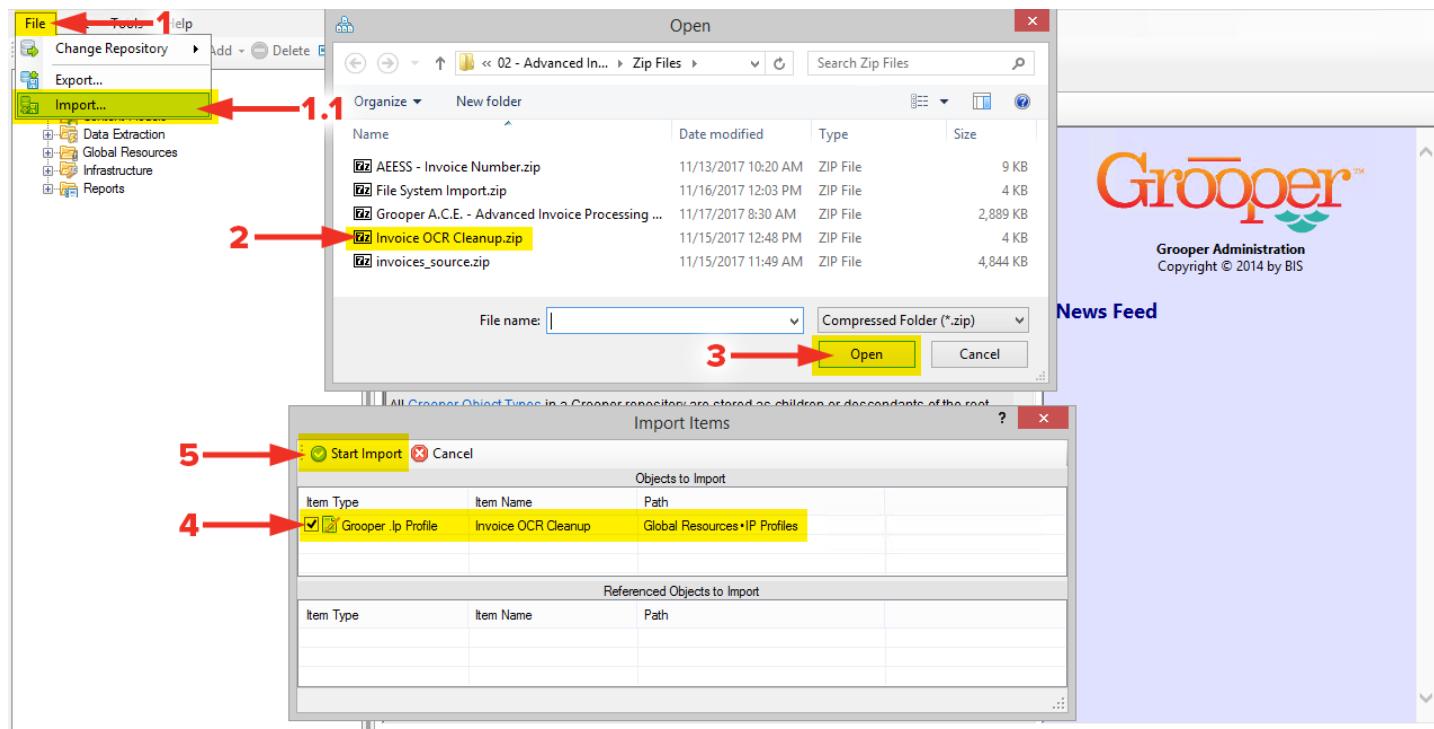
The previous documentation, [Overview and Concepts](#), focused on the creation of an **IP Profile** during this **Conditioning** phase. The documents being worked with for these exercises don't require permanent cleanup, so the focus this time will be on creating an [OCR Profile](#).

CREATING AND TESTING A NEW OCR PROFILE

An **IP Profile** will be leveraged for the non-permanent [OCR Cleanup](#) property, and that's been built and supplied.

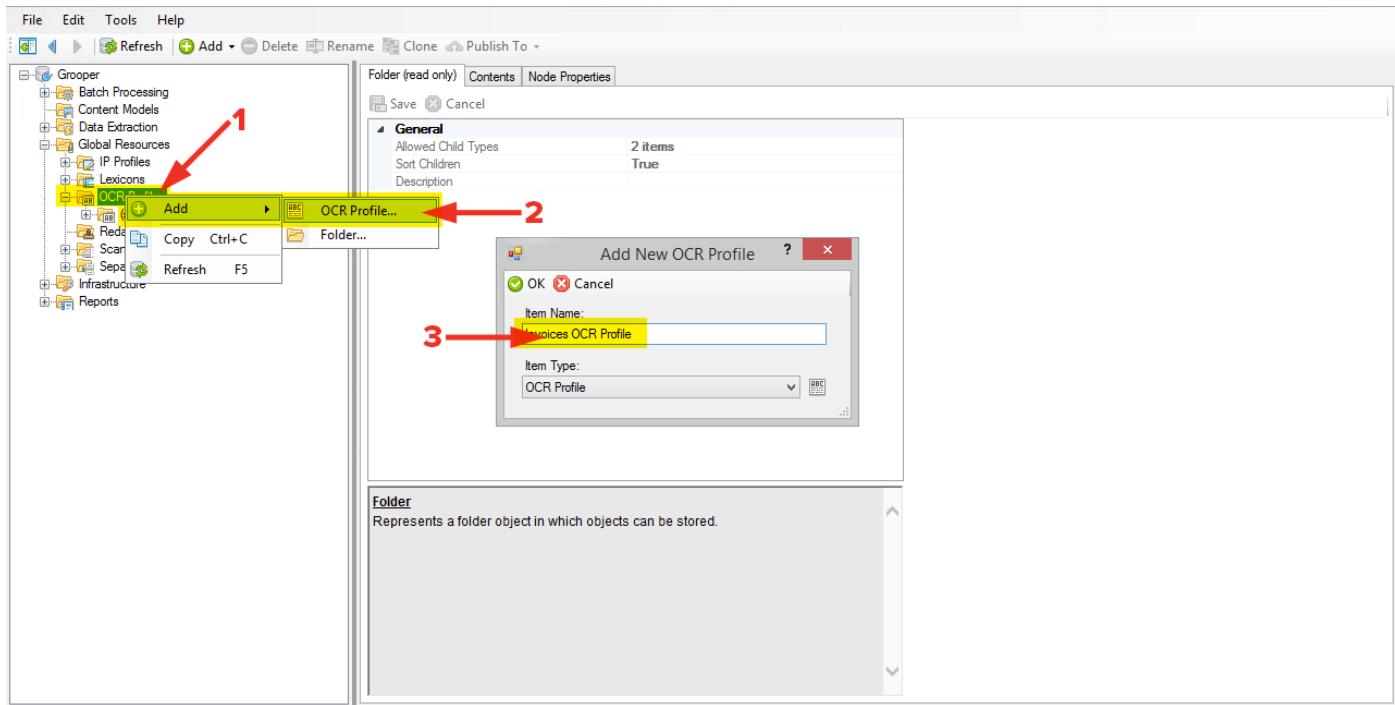
STEP 1 – IMPORT PRE-BUILT IP PROFILE

Start by (1) clicking **File > Import**, and from the **Open** window (2) select the [Invoice OCR Cleanup.zip](#) file and (3) click **Open**. In the **Import Items** window, (4) select the **Grooper .Ip Profile** object and (5) **Start Import**.



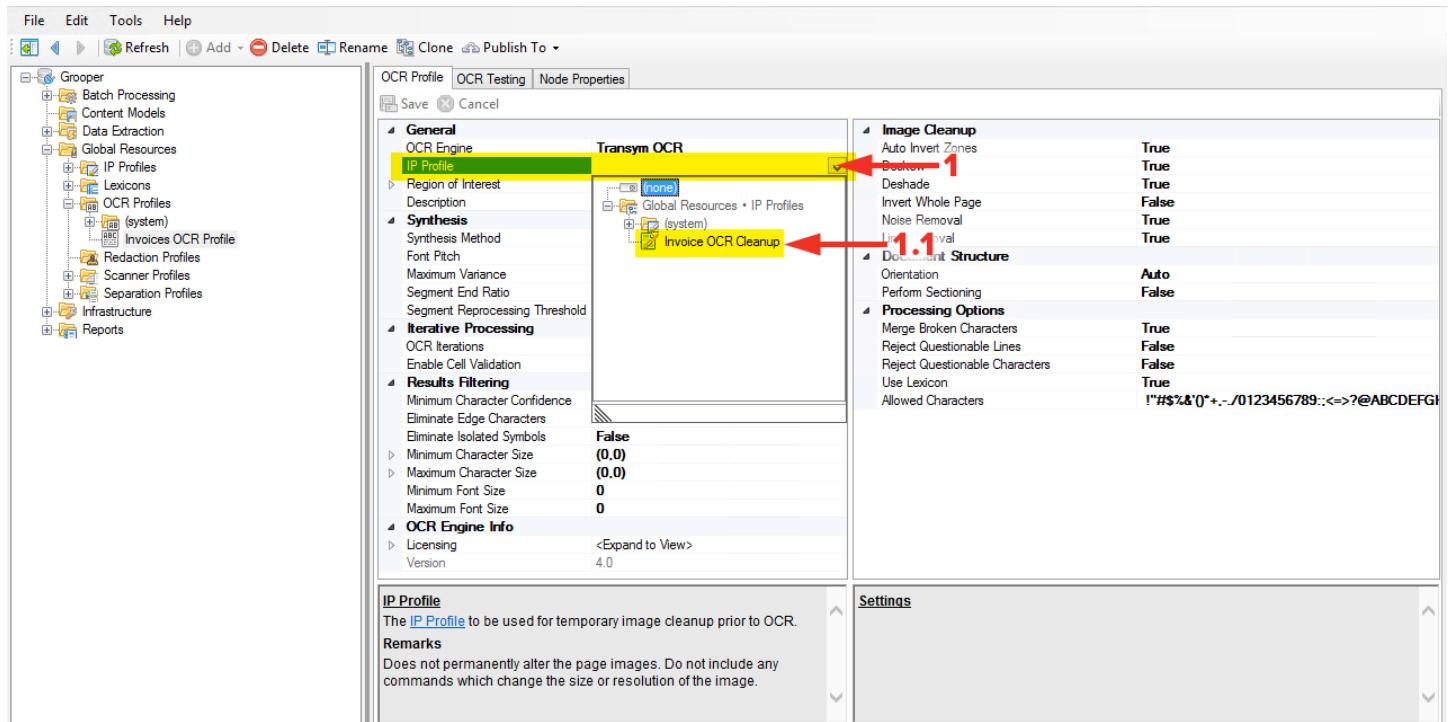
STEP 2 – CREATING A NEW OCR PROFILE

(1) Navigate to Grooper > Global Resources, (2) add a new OCR Profile, and (3) name it **Invoices OCR Profile**.



STEP 3 – SET IP PROFILE

With the newly created **OCR Profile** selected, (1) set the **IP Profile** to the one that was just imported.



STEP 4 – SYNTHESIS SETTINGS • SEGMENT END RATIO

A segment in **OCR'ed** text is a set of synthesized characters on a line that are not separated by too large a space. The space considered “too large” is set by **Segment End Ratio**, in the **Synthesis** section. Set this to **125%**, so a space must be 125% larger than the detected font size for it to be considered the end of a segment.

The screenshot shows the Grooper ACE software interface. On the left is a tree view of project resources under 'Grooper'. The main area is a configuration window for an 'OCR Profile' named 'Invoice OCR Cleanup'. The 'Synthesis' section is expanded, showing the 'Segment End Ratio' field highlighted with a yellow box and a red arrow pointing to its value of '125'. Other settings like 'Synthesis Method' (Full), 'Font Pitch' (Auto), and 'Maximum Variance' (10%) are also visible. A tooltip for 'Segment End Ratio' explains its function: 'When synthesis is enabled, controls how wide a gap must be in relation to the current font size in order to constitute the end of a segment.'

STEP 5 – SYNTHESIS SETTINGS • SEGMENT REPROCESSING THRESHOLD

Now that a segment is understood, (1) set **Segment Reprocessing Threshold** to **90%** and (2) take a moment to read over the tooltip:

Text segments with an average character confidence below this value will be re-processed through OCR.

This screenshot is similar to the previous one but shows the 'Segment Reprocessing Threshold' field highlighted with a yellow box and a red arrow pointing to its value of '90%'. The tooltip for this field provides more detail: 'Text segments with an average character confidence below this value will be re-processed through OCR.' A red number '2' is placed next to the 'Remarks' section at the bottom of the tooltip, which states: 'A value of 0 disables segment reprocessing entirely.'

STEP 6 – ITERATIVE PROCESSING • OCR ITERATIONS AND CELL VALIDATION

In the Iterative Processing section, (1) set **OCR Iterations** to 2. (2) Enable **Cell Validation** (which will present a new set of properties) and set the **Rows** and **Columns** to 1 and 4 respectively. (3) Set **Skip First Column** to True.

The screenshot shows the Grooper interface with the 'OCR Profile' tab selected. In the 'Iterative Processing' section, the 'OCR Iterations' field is set to 2 (1). The 'Enable Cell Validation' checkbox is checked (2). The 'Rows' and 'Columns' fields are set to 1 and 4 respectively. The 'Skip First Column' checkbox is checked (3). The 'Image Cleanup' section contains several options like Auto Invert Zones, Deskew, and Noise Removal, all set to True. The 'Document Structure' section includes Orientation and Perform Sectioning, both set to Auto. The 'Processing Options' section includes Merge Broken Characters, Reject Questionable Lines, and Reject Questionable Characters, all set to False. The 'Results Filtering' section includes Minimum Character Confidence (set to 0%), Eliminate Edge Characters (set to False), and Eliminate Isolated Symbols (set to True). The 'OCR Engine Info' section shows Licensing and Version information. A note below the 'Skip First Column' setting explains its purpose: "When Cell Validation is in use, indicates whether the first column of cells should be skipped." A 'Remarks' section notes that skipping the first column speeds up the validation process.

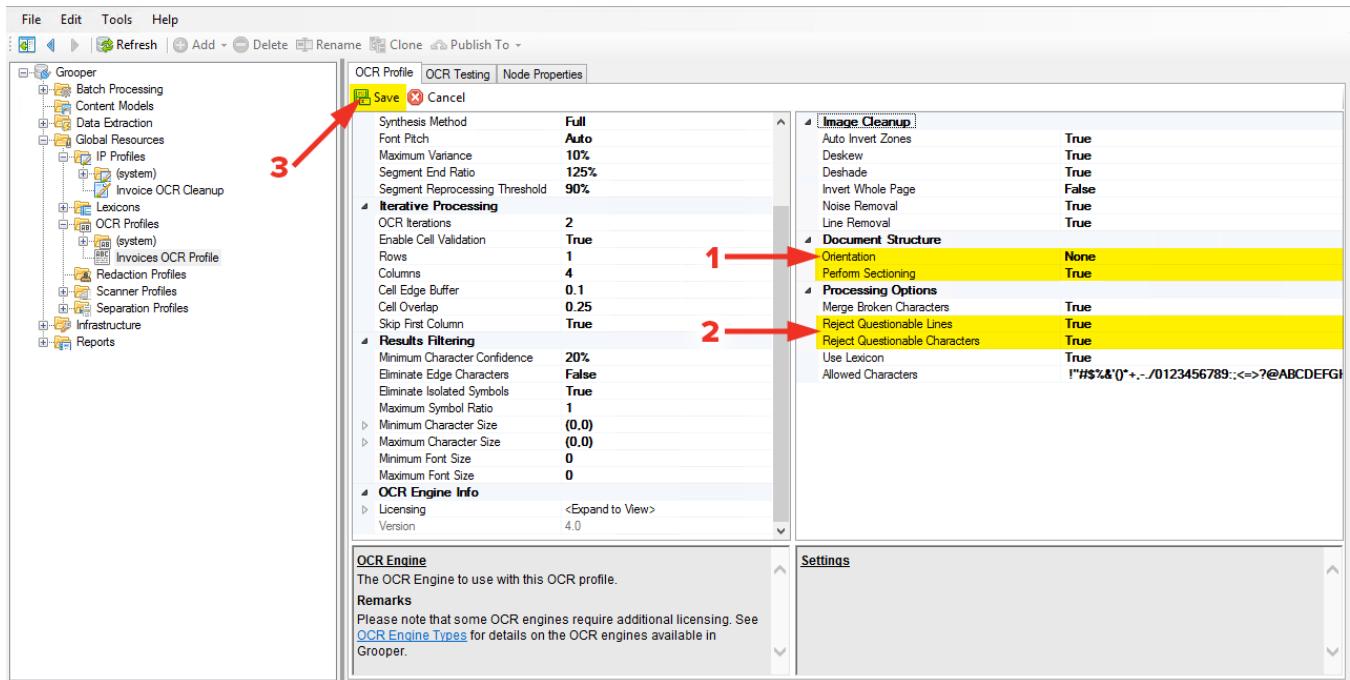
STEP 7 – RESULTS FILTERING

In the **Results Filtering** section, (1) set **Minimum Character Confidence** to 20% and (2) **Eliminate Isolated Symbols** to True.

This screenshot shows the same Grooper interface as the previous one, but with different settings highlighted. The 'Results Filtering' section now has 'Minimum Character Confidence' set to 20% (1) and 'Eliminate Isolated Symbols' set to True (2). The other settings remain the same as in Step 6. The 'Image Cleanup' and 'Processing Options' sections are also visible on the right side of the configuration window.

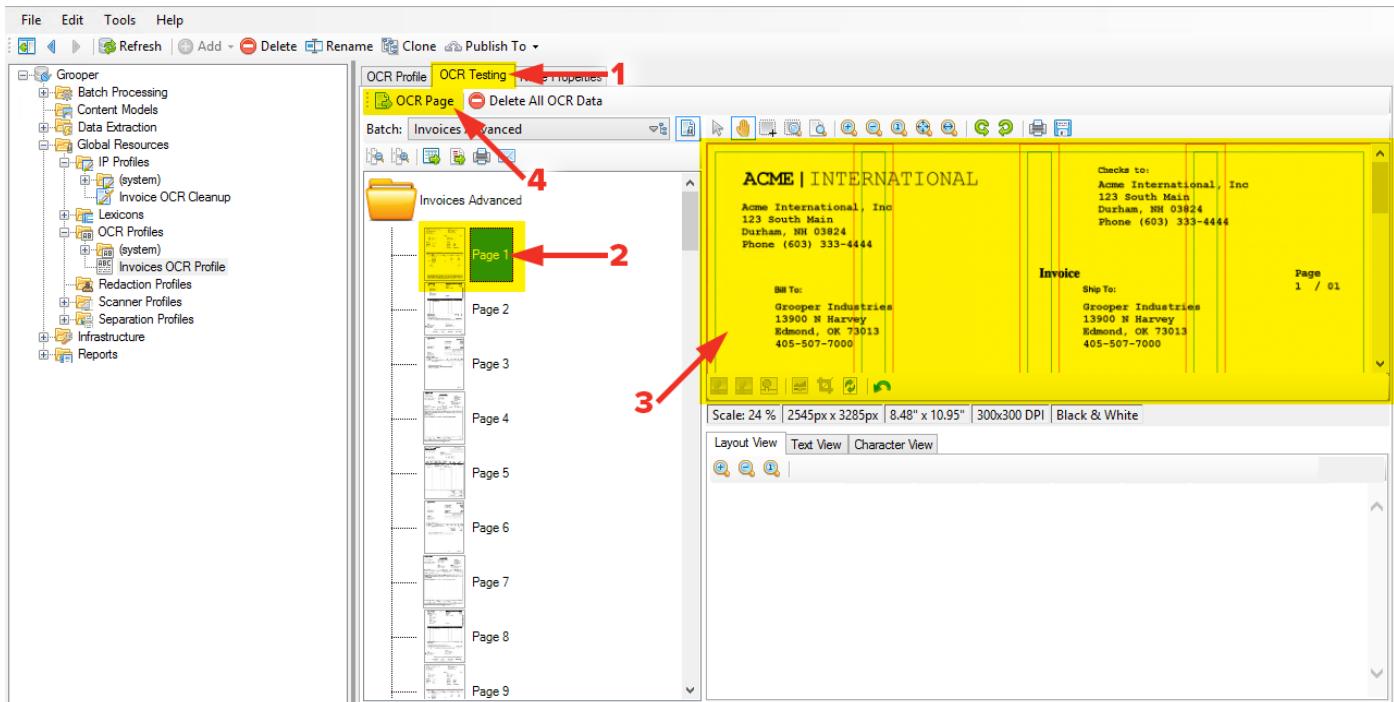
STEP 8 – FINAL ADJUSTMENTS

Finally, (1) set Orientation to None and Perform Sectioning to True, (2) set Reject Questionable Lines and Reject Questionable Characters to True. (3) Click the Save button.



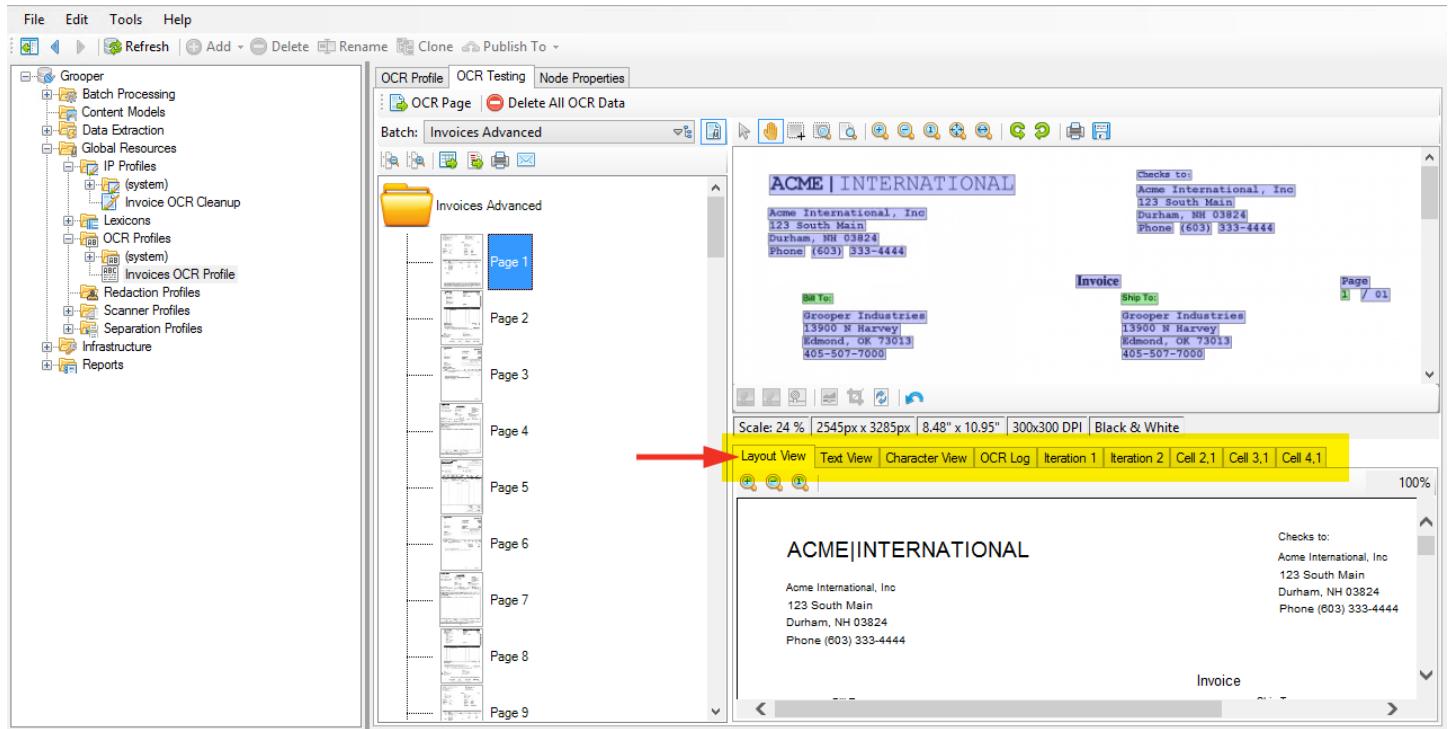
STEP 9 – TEST THE OCR PROFILE

(1) Click on the OCR Testing tab. (2) Select Page 1 from the Batch Viewer. (3) You'll notice in the Page Viewer how the selected page is divided into columns and that those columns overlap due to the column settings from the Iterative Processing section. (4) Click the OCR Page button.



STEP 10 – OBSERVE RESULTS

Look at the different tabs that are available as a result of the [OCR Page](#) test. Take note that this “test” permanently altered the page object as it applied [OCR](#) results.



The [Layout View](#) tab shows a representation of the [OCR](#)'ed characters in a view that resembles the image. The [Text View](#) tab shows the characters in their synthesized layout including carriage returns and line breaks. The [Character View](#) tab shows a table with each row being each individual character [OCR](#)'ed, and columns representing specific results about that character from its XY coordinates on the page to percentage confidence results, etc.

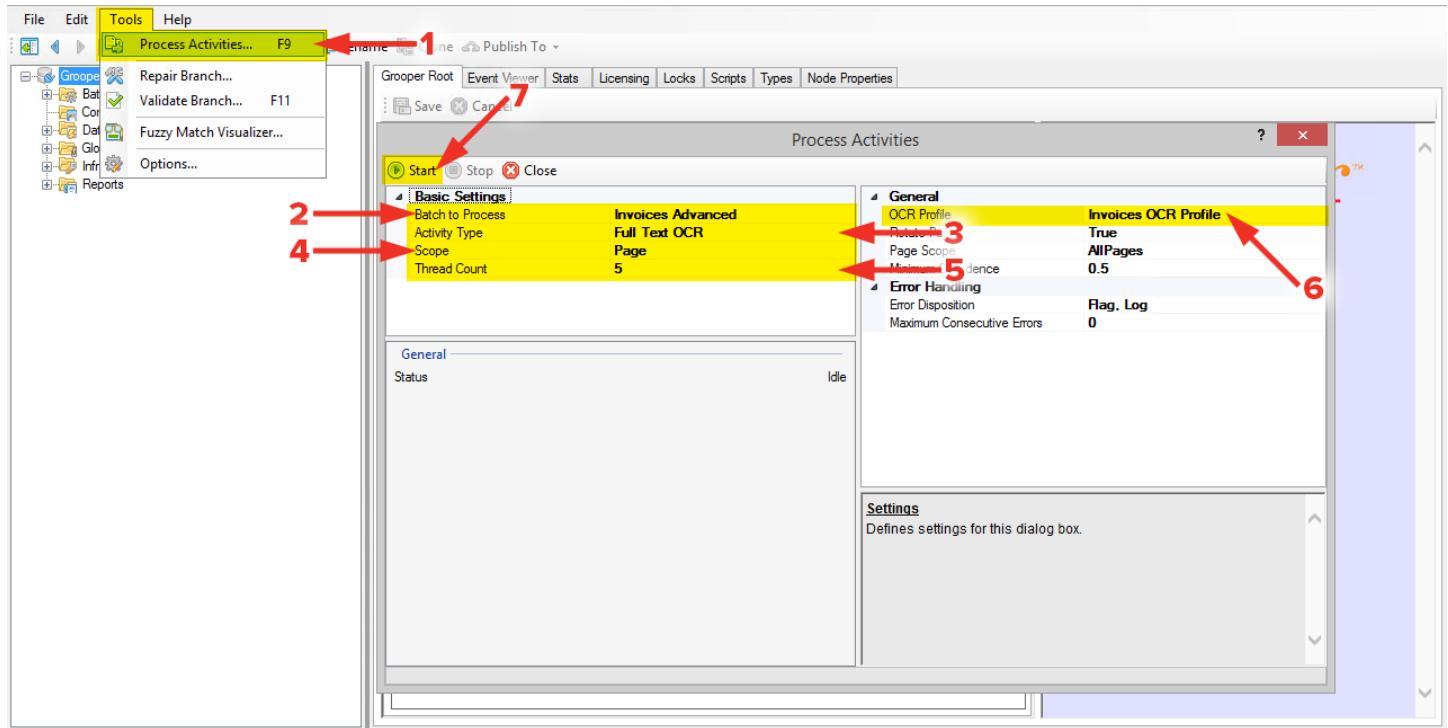
The [OCR Log](#) tab displays information regarding specificities of the [OCR](#) process.

Finally, because iterations and cells were used in the [OCR Profile](#), the remaining tabs show the characters read from the different iterations, as well as the images made from the column breakouts.

STEP 11 – PROCESS OCR ON THE BATCH

With the **OCR Profile** created and successfully exported, select the **Grooper Root Node** and then click the **Refresh** button that is directly above it. After this, **(1)** click on the **Tools** menu and select **Process Activities**. In the **Process Activities** window **(2)** select **Invoices Advanced** for the **Batch to Process**, **(3)** **Full Text OCR** for the **Activity Type**, **(4)** **Page** for **Scope**, **(5)** an amount that doesn't exceed your **Grooper OCR** license count for **Thread Count**, **(6)** **Invoices OCR Profile** for **OCR Profile**, and the other settings can be defaulted. **(7)** Click **Start**, and the **OCR Profile** will be run against all pages in the batch.

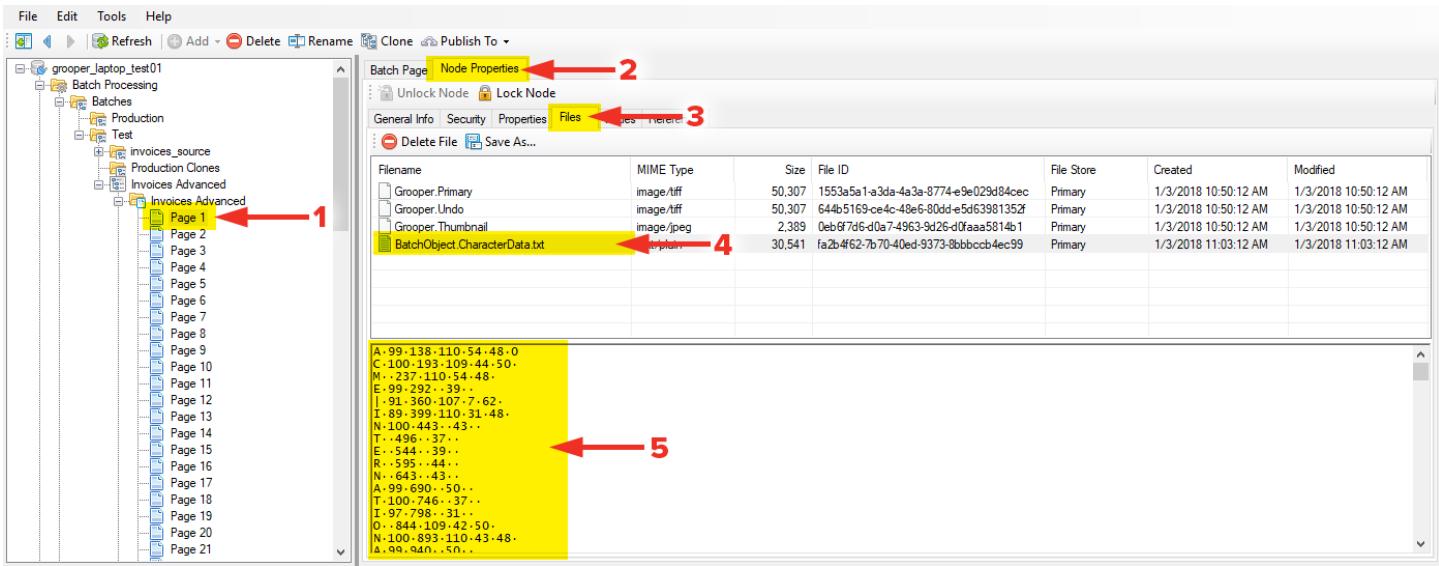
At this point a blue progress bar will begin to fill the bottom of the **Process Activities** window. Once it completes, a **Process Completed** window will pop up and you can close it. Once done, close the **Process Activities** window.



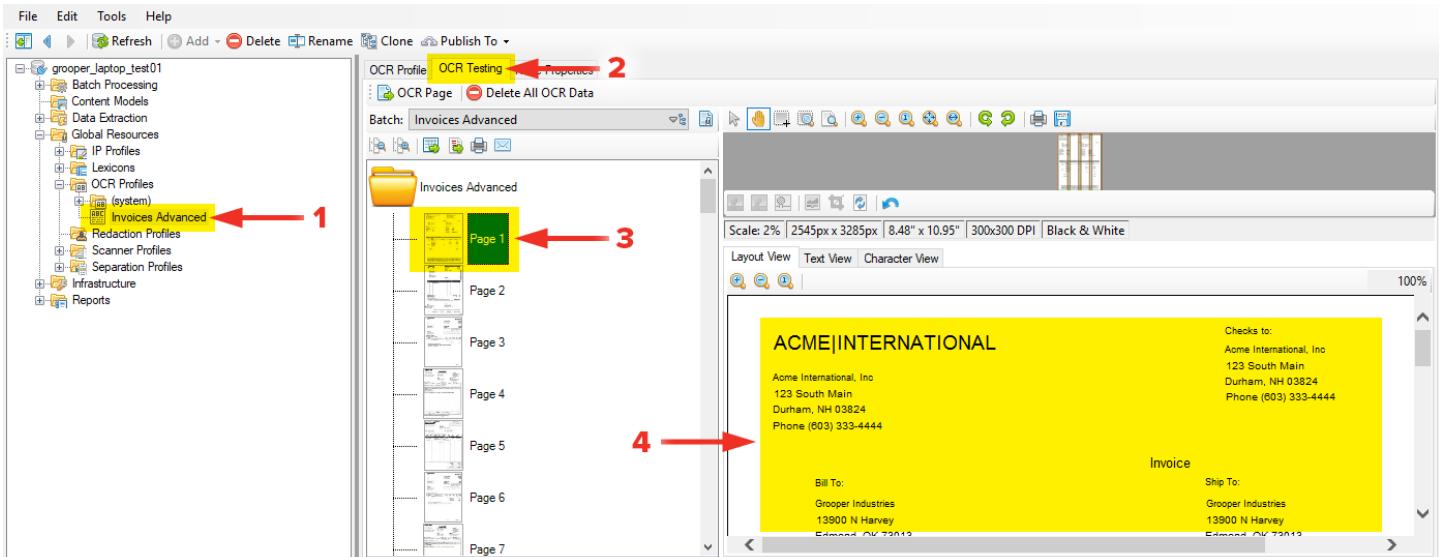
This ability to perform individual activities on the fly like this (via **Process Activities**) is a powerful feature in **Grooper** as it allows testing and results to be reached quickly and dynamically. You don't have to build out an entire **Batch Process** and test from end to end to see immediate results. As is probably obvious as well, this is not limited to Full Text OCR, as you can immediately process just about any activity in Grooper.

STEP 12 – CHECK FOR OCR RESULTS

You may find yourself at some point asking, “has this page/batch been **OCRed?**”. There are a couple of places to check for this. **(1)** The first way is to, navigate all the way to the page object within a batch folder object. With a page object selected **(2)** click on the **Node Properties** tab. From there, **(3)** click on the **Files** tab. **(4)** Select the **BatchObject.CharacterData.txt** file (the existence of this file is enough...) and **(5)** notice all the character data in the Document Viewer panel.



The next place to place to determine if you have **OCR** results is within the **OCR Testing** tab of an **OCR Profile**. **(1)** Select any **OCR Profile** and **(2)** click on the **OCR Testing** tab. **(3)** Select a page in the **Batch Viewer** and **(4)** in the **Layout View**, if you have any characters visible, that page has been **OCRed**.



PHASE 3 – ORGANIZE

Separation and Classification are the next activities to be run against our batch. Control Sheets have been removed from the pages in this batch to show a new method of Separation. ESP Auto Separation could still be used to classify the documents once separated, but again, a different approach using extractors will be taken to show a different means of classifying than has been seen previously.

SEPARATION – CHANGE IN VALUE

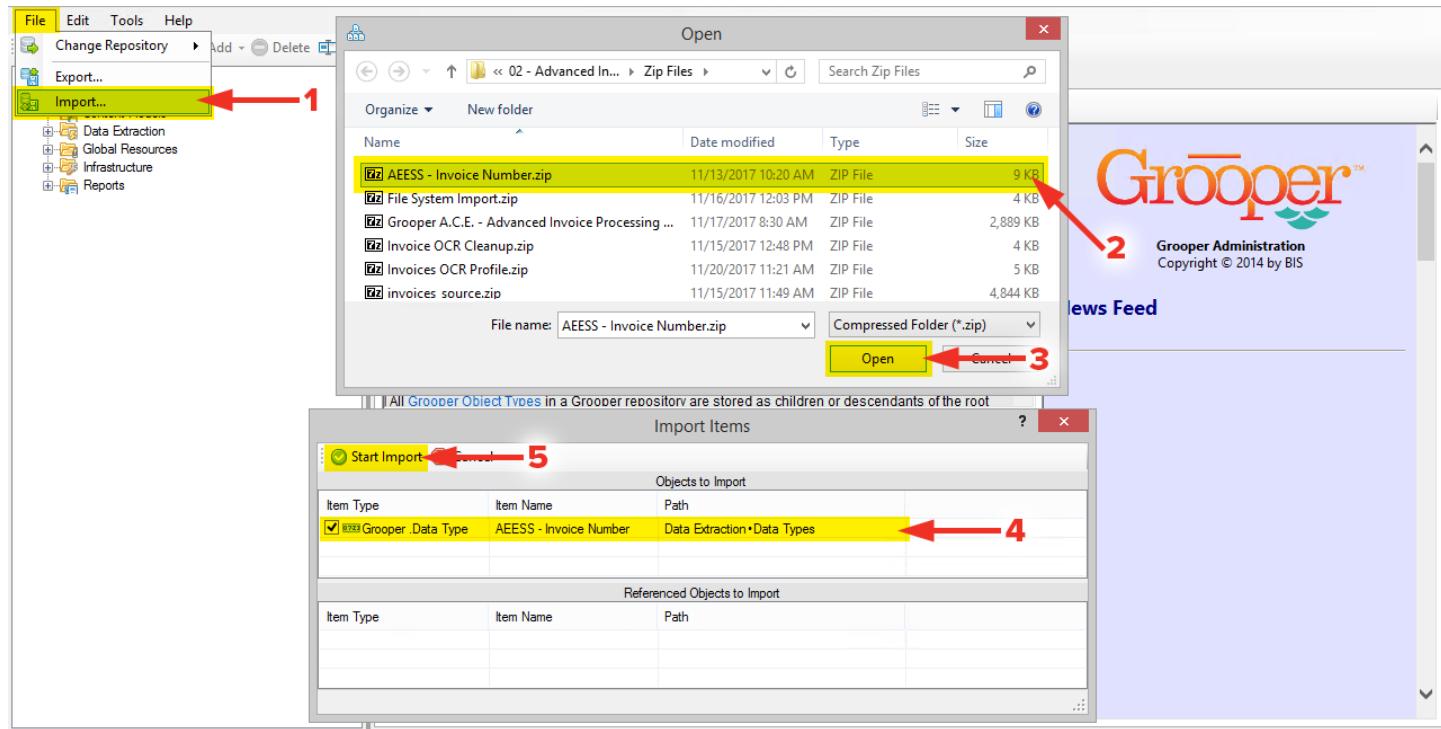
For this method of Separation to work, an extractor will be used in a Separation Profile to find a specific value on the pages. Until the value found changes, pages will be sorted together and treated as a single document. This Change in Value approach works well for invoices.

SETTING UP A SEPARATION PROFILE TO LEVERAGE A CHANGE IN VALUE

STEP 1 – IMPORT DATA TYPE

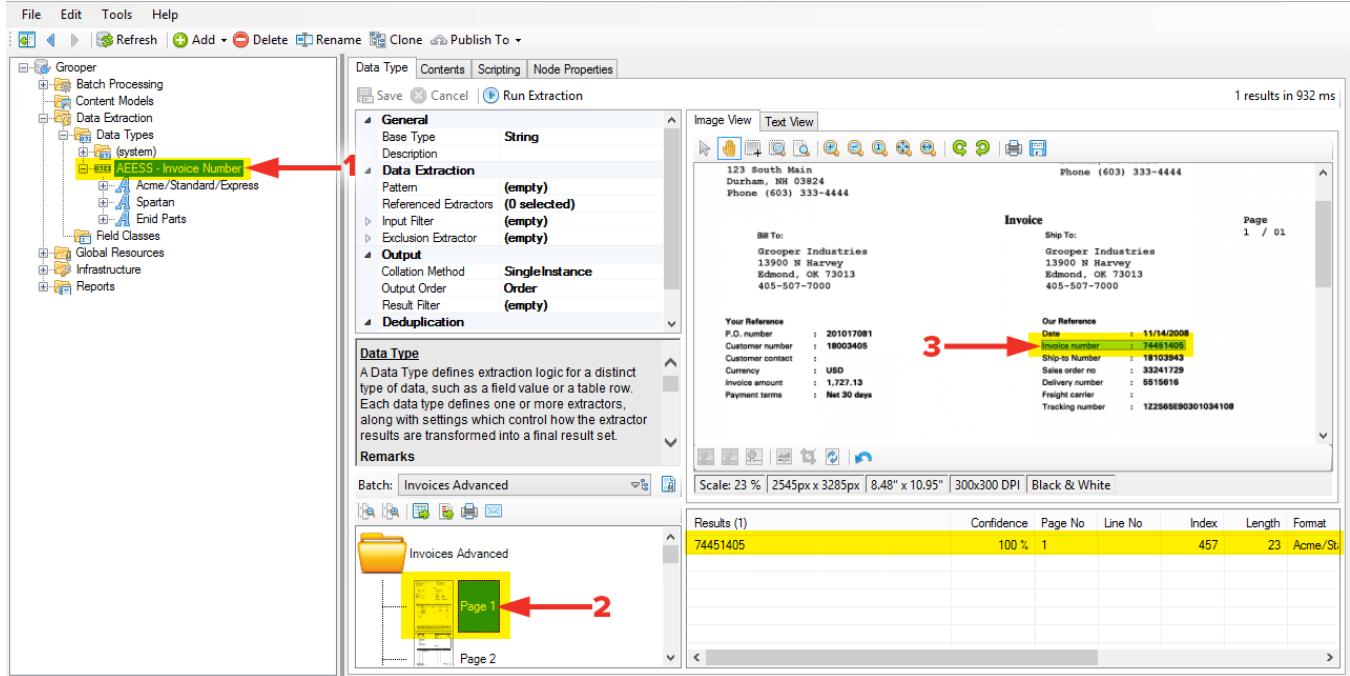
Instead of building the extractor necessary to perform this operation, the Data Type that was built during the Overview and Concepts exercises was exported and will be used here.

Start by (1) clicking File > Import and in the Open window (2) select the AEESS – Invoice Number.zip file (3) then click Open. In the Import Items window (4) select the Data Type and (5) click Start Import.



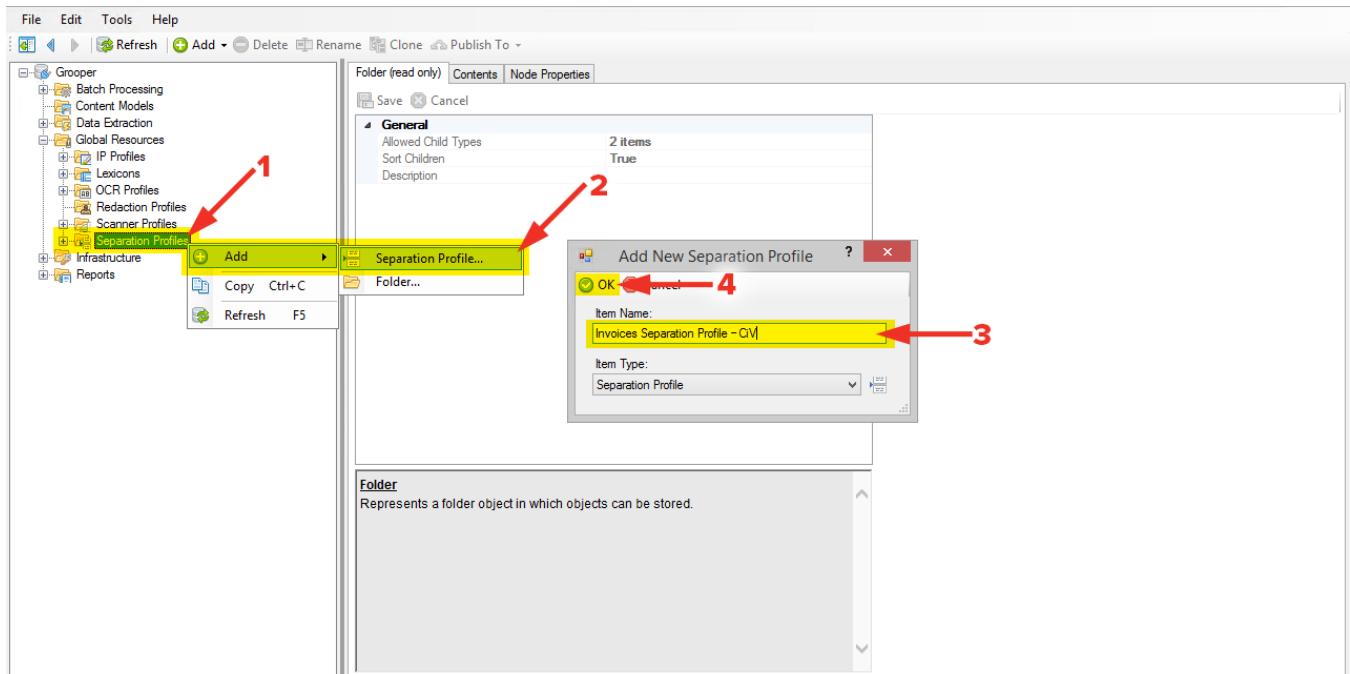
STEP 2 – REVIEWING THE DATA TYPE

To refresh your memory of what this **Data Type** did, (1) navigate to it in the node tree and (2) select a page object of the **Batch Viewer**. (3) The **Invoice Number** for this page will be highlighted, as will that of the other pages if you click through them as well.



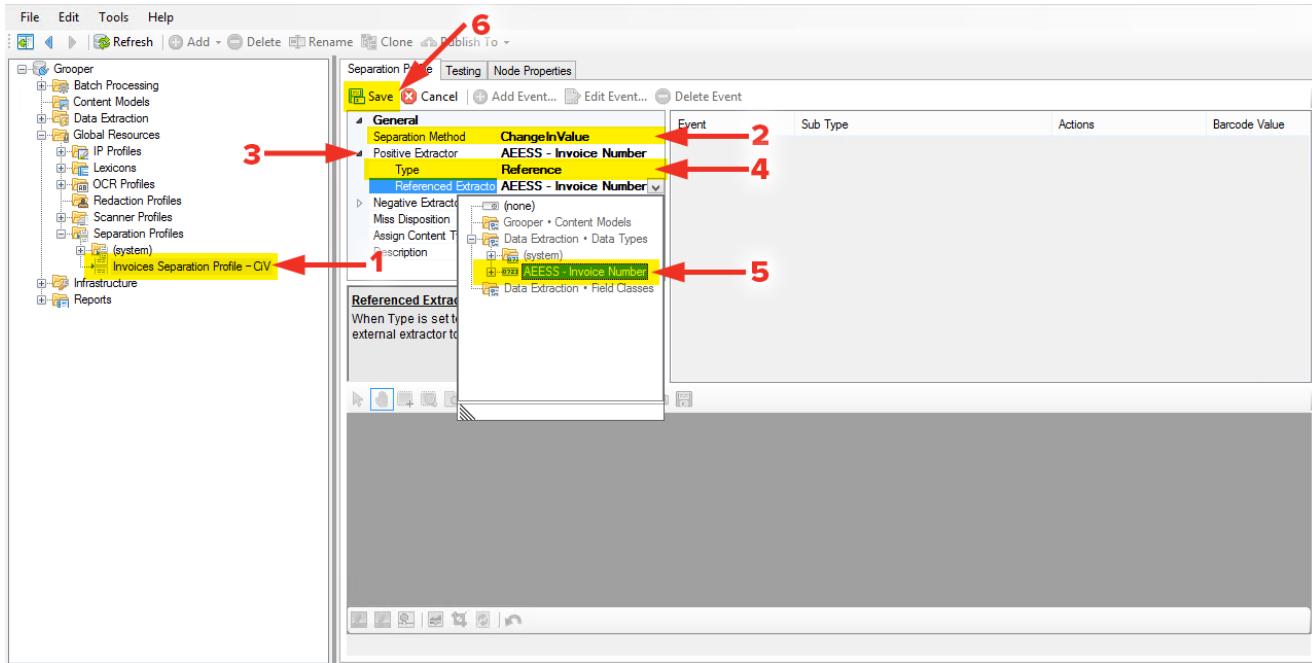
STEP 3 – SETTING UP THE SEPARATION PROFILE

(1) Navigate to **Grooper > Global Resources** and select **Separation Profiles**. (2) Right click and **Add > Separation Profile...** In the **Add New Separation Profile** window (3) name it **Invoices Separation Profile – CiV** and (4) click **Ok**.



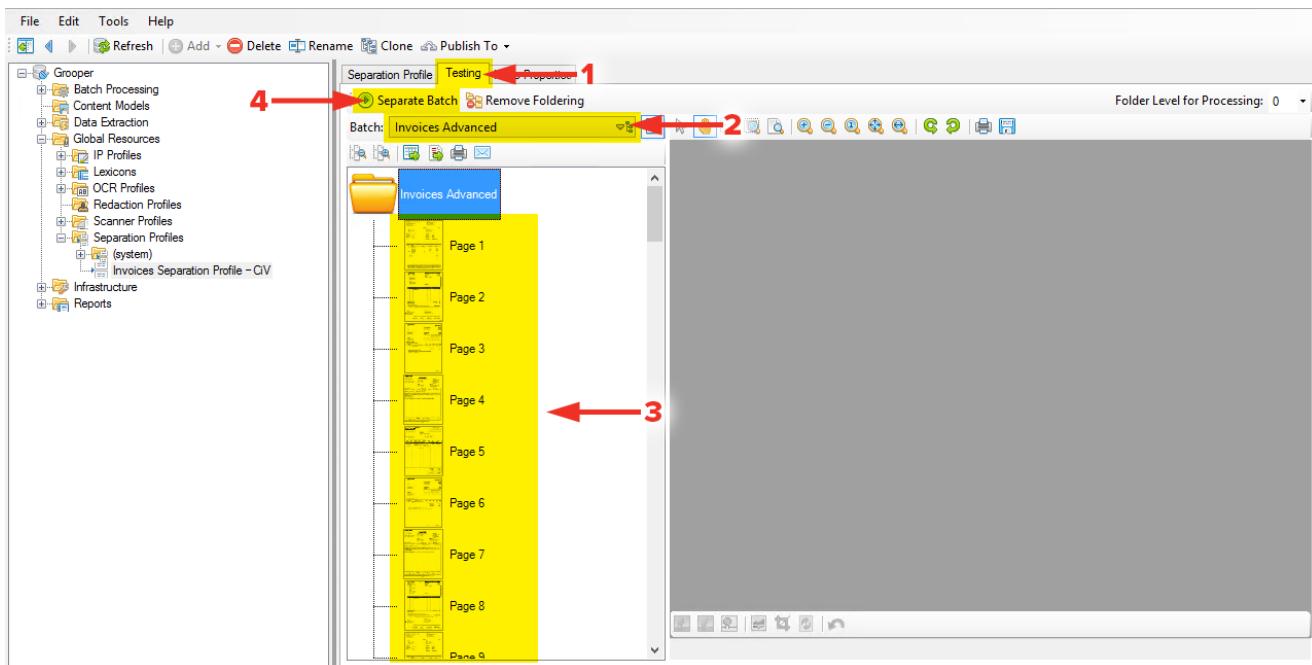
STEP 4 – SETTING PROPERTIES FOR NEW SEPARATION PROFILE

(1) With the newly created Separation Profile selected, (2) set the Separation Method to ChangeInValue. (3) Expand Positive Extractor and (4) set its Type to Reference, and (5) set the Referenced Extractor to the AEESS – Invoice Number extractor that was imported. (6) Click Save.



STEP 5 – TESTING SEPARATION

(1) Click the Testing tab and (2) make sure the Invoices Advanced batch is selected in the Batch: drop-down. (3) Observe the Batch Viewer and notice the pages are loose and not separated. (4) Click the Separate Batch button and separation will execute each time the Data Type that was imported detects a change in the value extracted. As a result, the loose pages will be in folders which can now be classified.



STEP 6 – TWO PAGE EXAMPLE

- (1) In the **Batch Viewer**, scroll down to **Folder (16)** and notice there are two pages in this document.
- (2) Observe the value extracted on **page 1** and on **page 2** and this should help illustrate that since the value didn't change between these pages, they were combined into one document.

The screenshot shows the Grooper ACE software interface. On the left, the 'Batch Processing' tree view includes 'Content Models', 'Global Resources' (with 'IP Profiles', 'OCR Profiles', 'Scanner Profiles', and 'Separation Profiles'), and 'Reports'. A specific 'Invoices Separation Profile - CIV' is selected under 'Global Resources'. The main workspace displays a 'Separation Profile' titled 'Invoices Advanced'. A folder named 'Folder (16)' is highlighted with a green box and contains two sub-folders labeled 'Page 1' and 'Page 2'. To the right, the 'Invoice' tab is active, showing an invoice from 'ACME | INTERNATIONAL' to 'Grooper Industries'. The invoice details include:

ACME | INTERNATIONAL

Acme International, Inc
123 South Main
Durham, NH 03824
Phone (603) 333-4444

Invoice

To: Acme International, Inc
123 South Main
Durham, NH 03824
Phone (603) 333-4444

From: Grooper Industries
13900 N Harvey
Edmond, OK 73013
405-507-7000

Your Reference#

P.O. number	: 010013809
Customer number	: 123456789
Customer contact	: Ken Stork
Currency	: USD
Invoice amount	: 2,432.98
Payment terms	: Net 30 days

Our Reference

Date	: 12/05/2008
Invoice number	: 7445635
Ship to Number	: 18103443
Sales order no	: 33247493
Delivery number	: 5519238
Freight carrier	:

Invoice details

Item	Material Description	Quantity	Unit Price	Value
000010	GB.B145645-00001 SHOCK ABSORBER	1 EA	480.00	480.00

Scale: 24 % | 2533px x 3276px | 8.44" x 10.92" | 300x300 DPI | Black & White

A red arrow labeled '1' points to the 'Page 1' section of 'Folder (16)'. A red arrow labeled '2' points to the 'Invoice' tab where the 'Invoice number' field is highlighted.

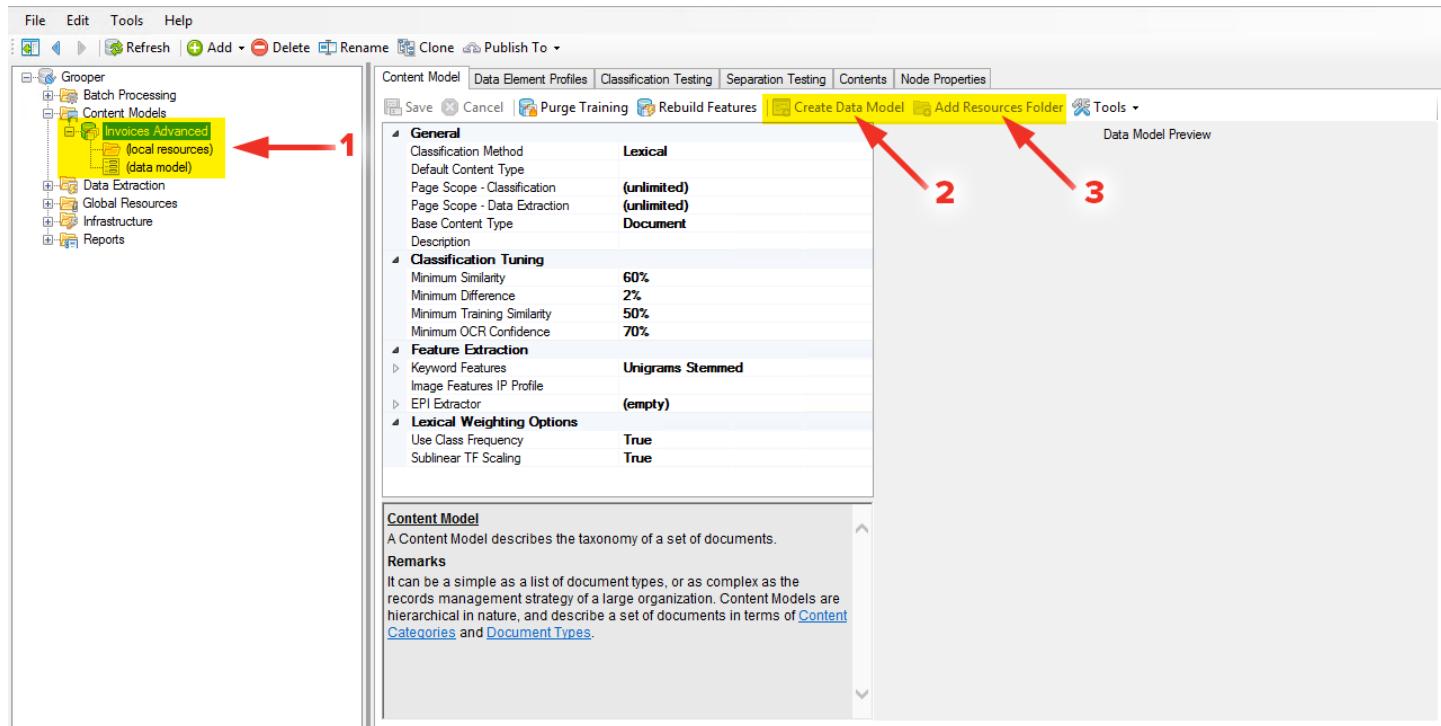
CLASSIFICATION – POSITIVE EXTRACTOR ON DOCUMENT TYPE

In previous exercises, **Classification** was performed because of **Lexical** analysis and training of documents. That approach would still work here, but learning a new method would be prudent. This new means of **Classification** allows a **Document Type** to leverage an extractor to find a value. If/when that value is found, the document is classified as that **Document Type**.

SETTING UP POSITIVE EXTRACTORS FOR DOCUMENT TYPES

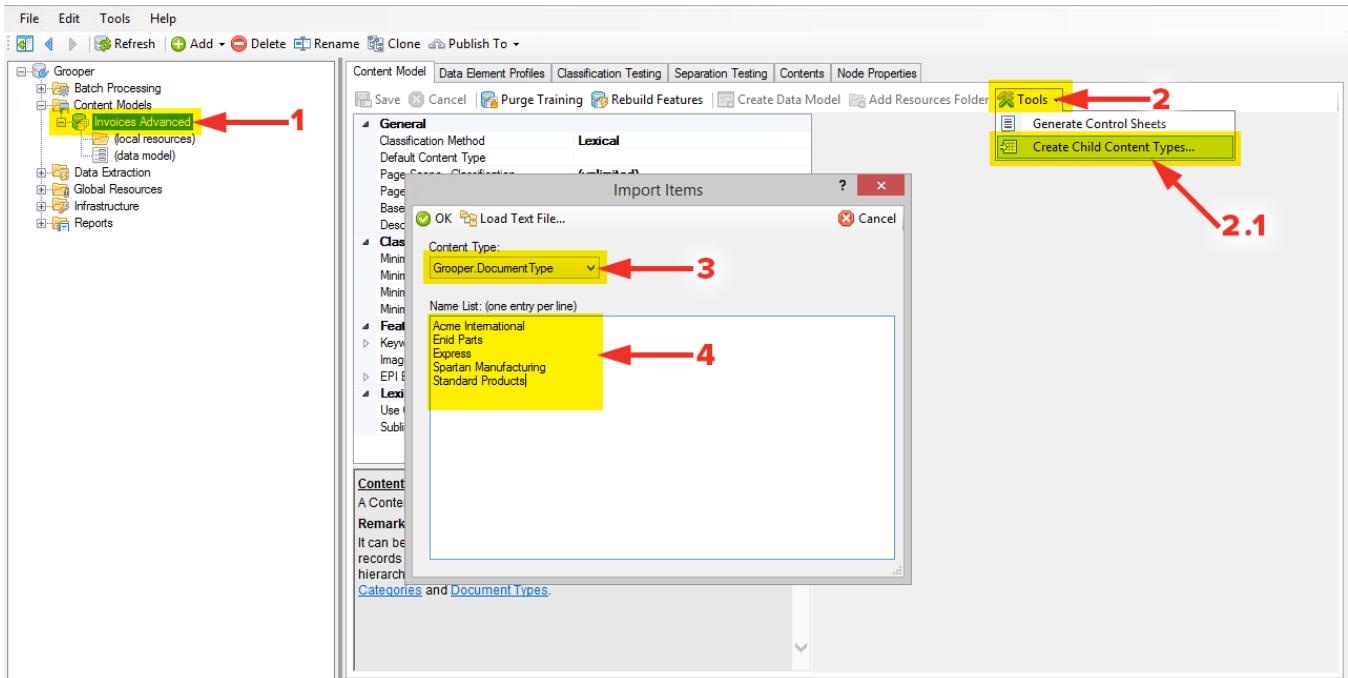
STEP 1 – CREATING A CONTENT MODEL

- (1) Create a new Content Model and name it **Invoices Advanced**. (2) Add a **(local resources)** folder (3) and a **(data model)**.



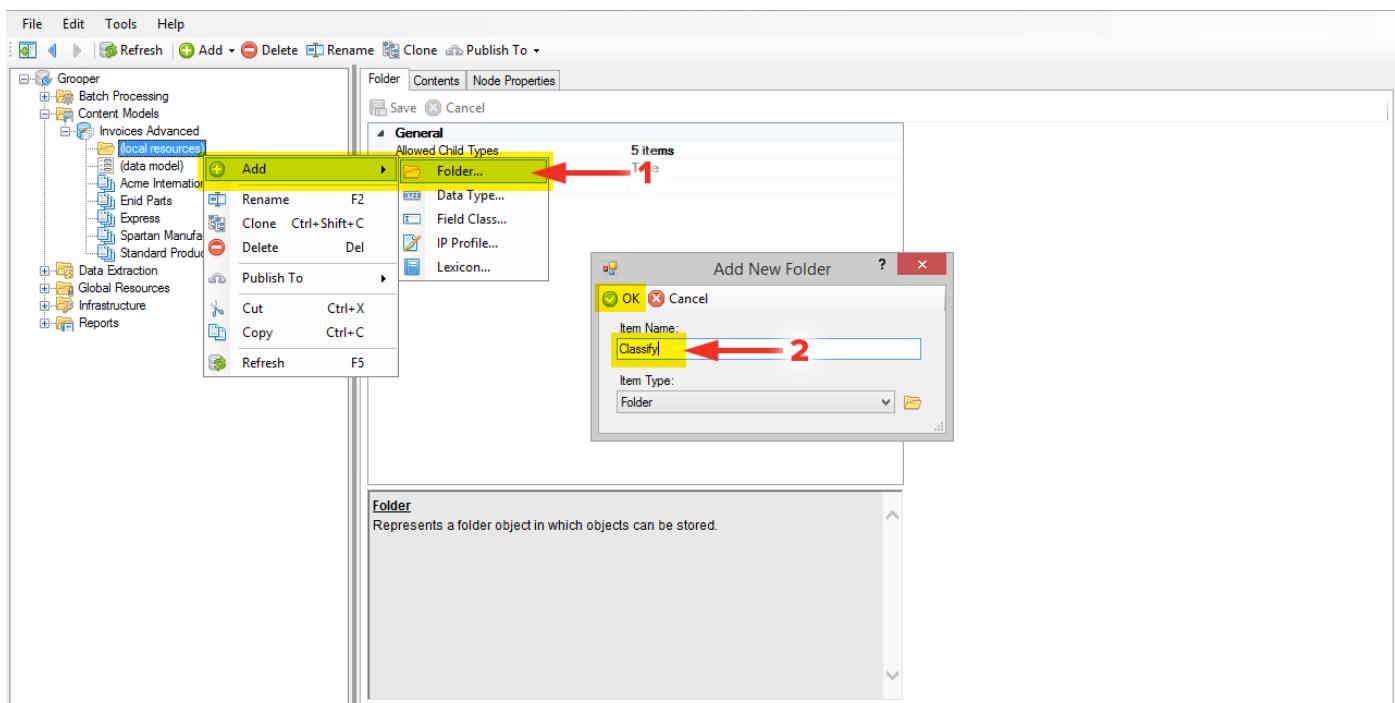
STEP 2 – ADD DOCUMENT TYPES

(1) With the newly created Content Model selected, (2) click on the Tools drop-down menu and select Create Child Content Types... In the Import Items menu, (3) make sure Content Type: is set to Grooper.DocumentType and in the Name List: (4) add the following names: Acme International, Enid Parts, Express, Spartan Manufacturing, and Standard Products. Click Ok.



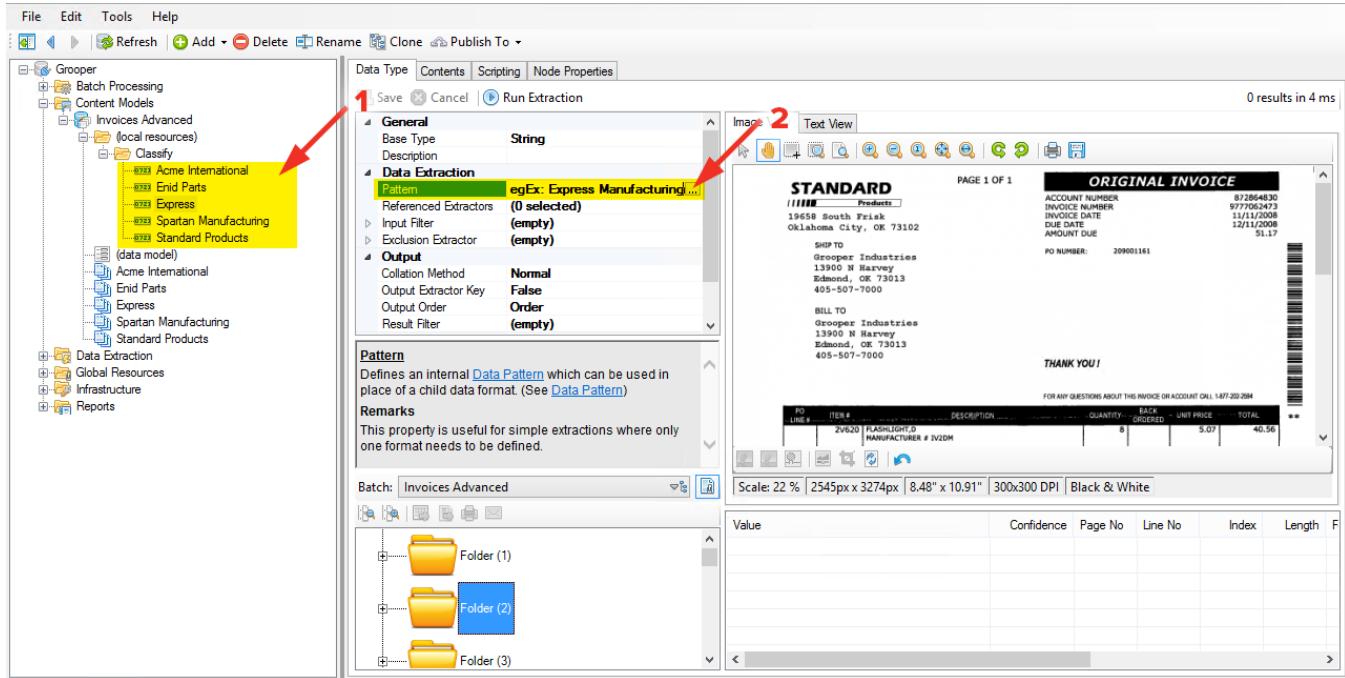
STEP 3 – ADD CLASSIFY FOLDER TO LOCAL RESOURCES

Select the (local resources) folder and (1) add a new folder (2) named Classify.



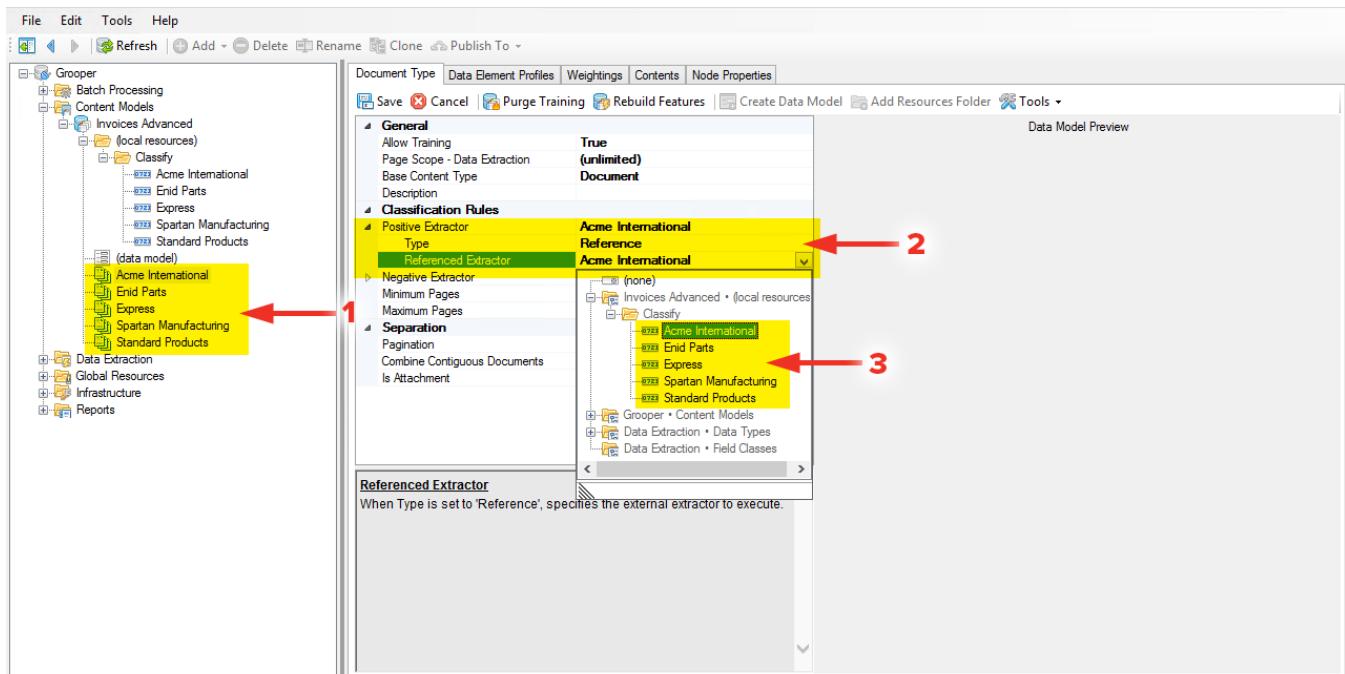
STEP 4 – ADD DATA TYPES

In the newly created **Classify** folder, (1) create 5 **Data Types** and (2) set their **Pattern** to reflect the title of each of the 5 company invoices: **Acme International, Enid Parts, Express Manufacturing, Spartan Manufacturing, Standard Products.**



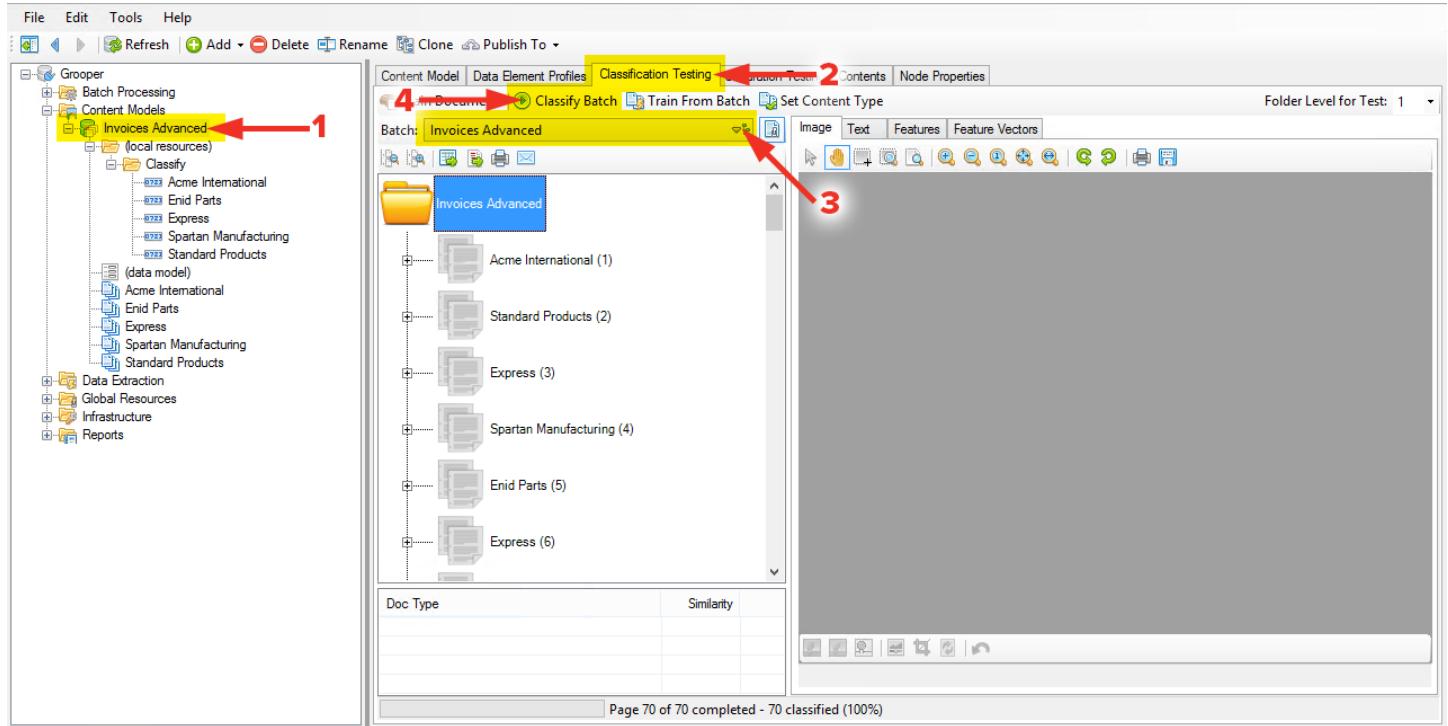
STEP 5 – DOCUMENT TYPE POSITIVE EXTRACTORS

(1) Go into each **Document Type** of the **Content Model** and (2) set their **Positive Extractor Types** to **Reference**, and (3) point the **Referenced Extractor** to their corresponding **Data Type** that was made in the **Classify** folder.



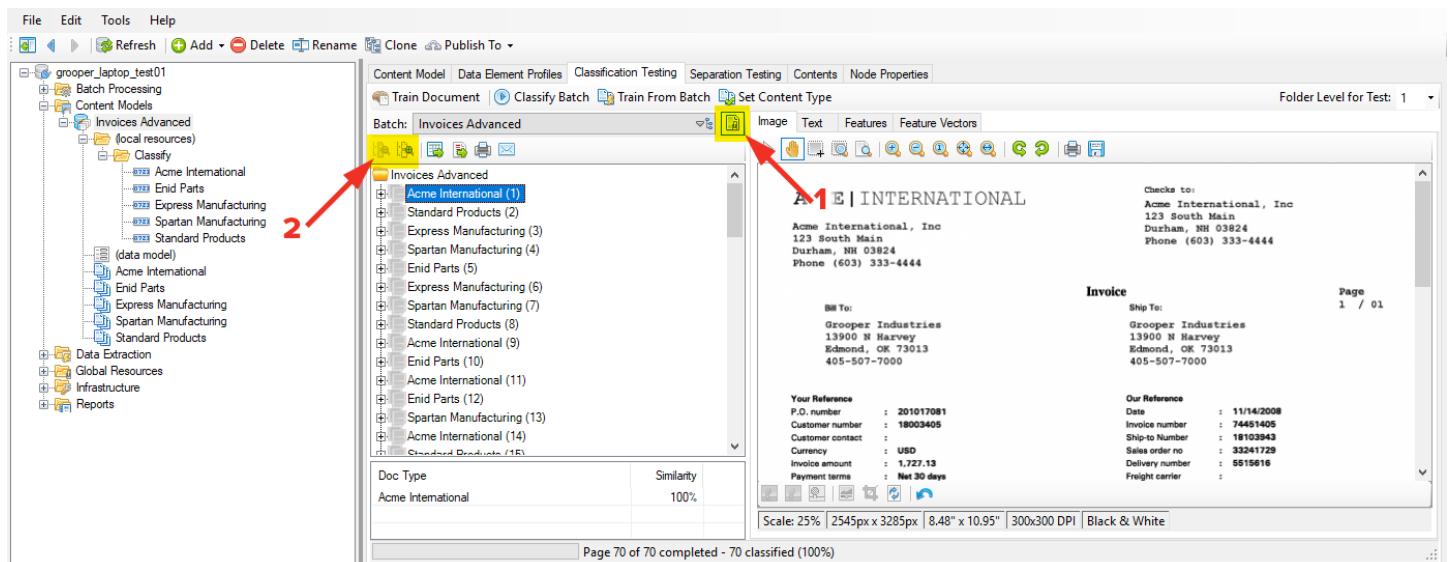
STEP 6 – CLASSIFY THE BATCH

(1) With the Content Model selected, (2) click on the Classification Testing tab. (3) Make sure the Invoices Advanced batch is selected in the Batch: drop-down. (4) Click the Classify Batch button (in the Classification Test Settings window that appears, leave it to DocType and click OK) and observe all the documents get accurately classified.



STEP 7 – BATCH VIEWER TOOLBAR INTERFACE BUTTONS

Finally, I'd like to call attention to some buttons in the Batch Viewer. (1) This button is very handy as you navigate in and out of different Batch Viewers as it will, when toggled, keep the last selected document the active/selected document. (2) While I have hi-lighted the zoom buttons as I use these the most frequently, there are others here for exporting, printing, and emailing selected documents.



PHASE 4 – COLLECT

As was mentioned in the introduction, the bulk of the time spent with this document will be here in **Phase 4 - Collect**. A new extractor, the [Field Class](#), will be introduced, as well as a new [Data Element](#), the [Data Table](#). Properties that were glazed over in previous exercises will be critical in getting desired results now. More, and new ways of practice with [RegularExpression](#) will be inherent and necessary to successful extraction.

ESTABLISHING A DATA MODEL AND WORKING WITH A DATA TABLE

It's best to have a solid understanding of what is required from the documents so that building out the entire Data Model from the beginning can be achieved. This will add to efficiencies down the road.

SETTING UP NEW DATA FIELDS AND DATA TABLE

To begin, all the [Data Fields](#) will be created. Once the desired fields of information are firmly established, how that information can be extracted is better understood, and the approach to building out the extractors will be easier.

STEP 1 – ADDING ALL THE DATA FIELDS

Add each of the following as [Data Field](#) objects to the [Data Model](#):

[Invoice No](#) • [PO Number](#) • [Invoice Date](#) • [Freight](#) • [Sales Tax](#)
[Discount](#) • [Invoice Amount](#) • [Payment Terms](#) • [Ship To](#) • [Remit To](#)

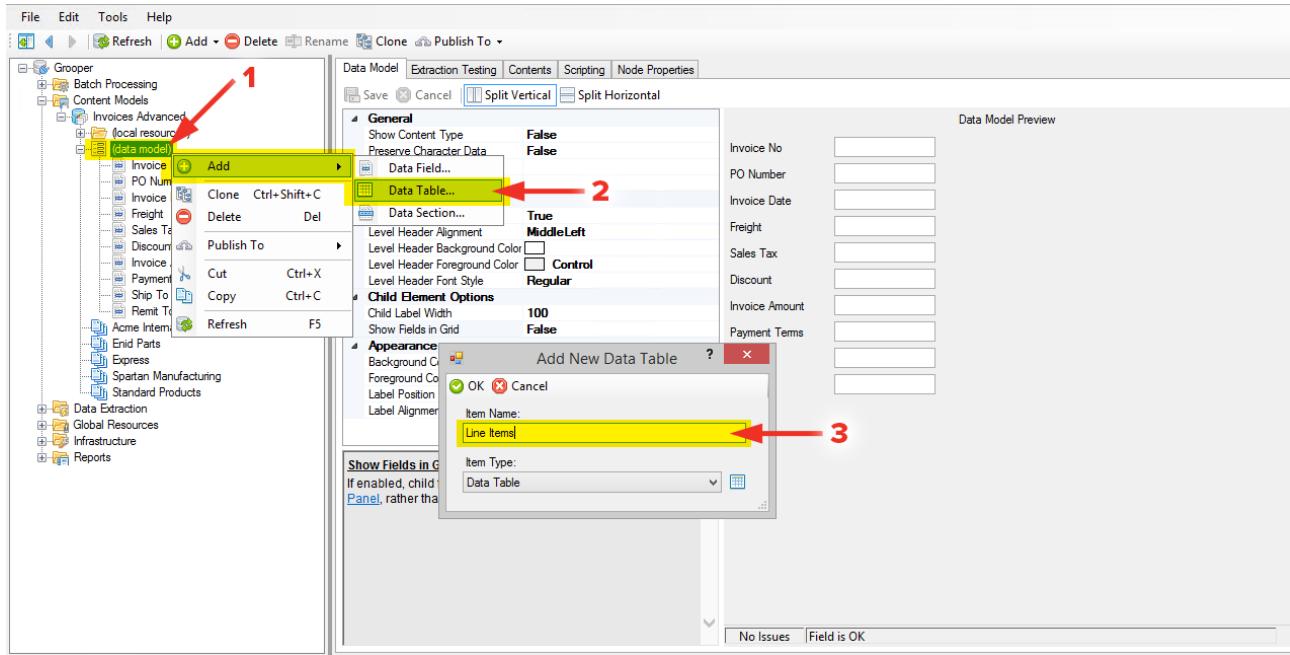
The screenshot shows the Grooper Node Tree on the left and the Data Model Editor on the right. In the Node Tree, a 'data model' node under 'Invoices Advanced' is selected and highlighted in yellow. The Data Model Editor has several tabs at the top: 'Data Model' (selected), 'Extraction Testing', 'Contents', 'Scripting', and 'Node Properties'. The 'Data Model' tab contains a 'General' section with properties like 'Show Content Type' (False) and 'Preserve Character Data' (False). It also includes sections for 'Multi-Level Display', 'Child Element Options', and 'Appearance'. On the right, there is a 'Data Model Preview' window showing a grid of fields: Invoice No, PO Number, Invoice Date, Freight, Sales Tax, Discount, Invoice Amount, Payment Terms, Ship To, and Remit To. Each field has a corresponding input field next to it. At the bottom of the preview window, there are buttons for 'No Issues' and 'Field is OK'.

If at any point you want to change the order of (most, not all) objects in the [Grooper Node Tree](#), simply select one, and hold the [Ctrl](#) button and use either the [↑](#) or [↓](#) arrow keys on the keyboard.

STEP 2 – ADDING NEW DATA ELEMENT – DATA TABLE

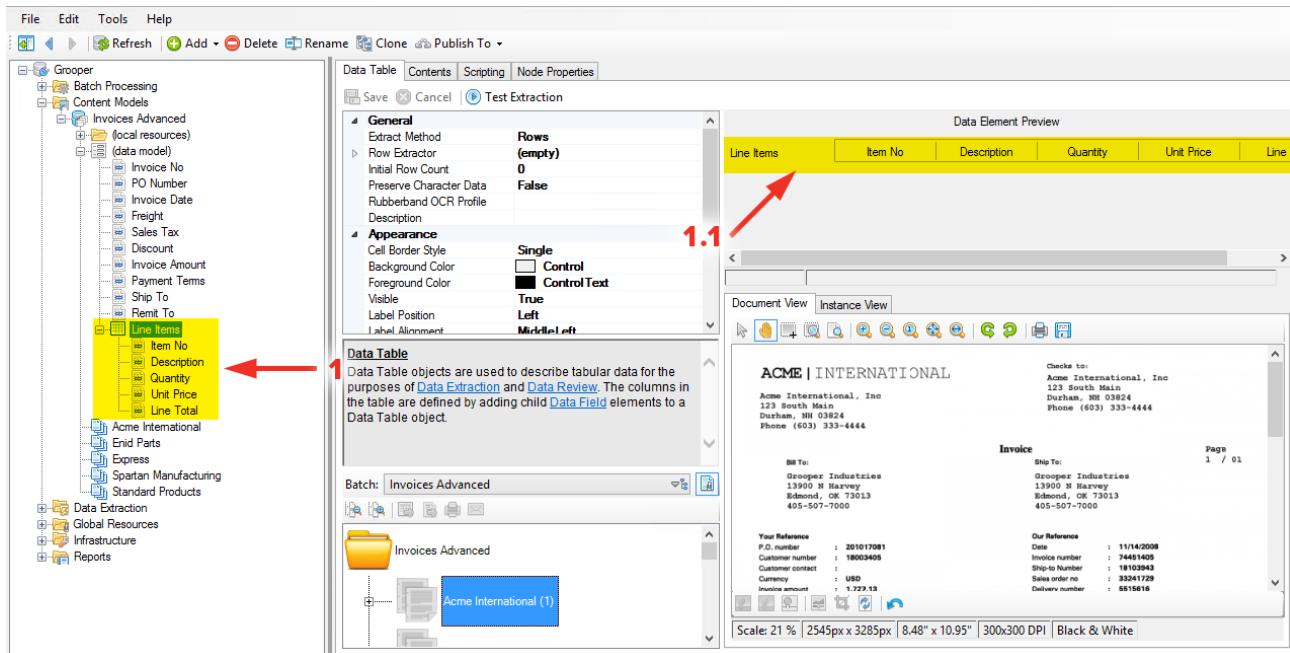
While individual **Data Fields** are purposed for displaying individual elements of data, there are instances where multiple elements represent one overall idea of information. For this, a **Data Table** works well. The **Data Table** is a child to the **Data Model**, but itself contains **Data Fields** that will be represented succinctly by their parent **Data Table**. It also houses some critical control properties that will dictate behaviors of the child **Data Fields**.

- Right Click on the **Data Model** (2) and Add > Data Table... (3) Name it **Line Items**.



STEP 3 – ADDING DATA FIELDS TO THE DATA TABLE

- Add each of the following as **Data Field** objects to the **Data Table** (pay very close attention to spelling):
Item No • **Description** • **Quantity** • **Unit Price** • **Line Total**



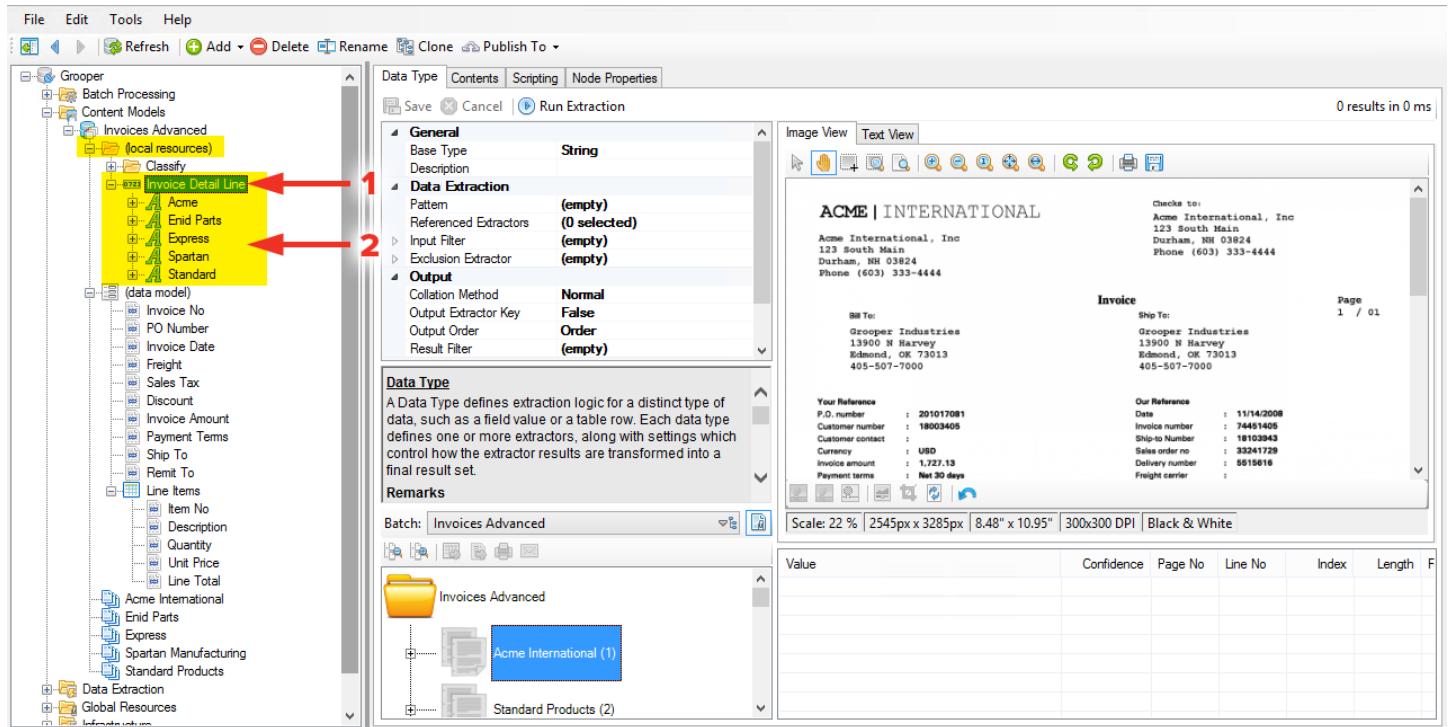
SETTING UP THE TABLE ROW EXTRACTOR

It is possible to create an extractor for each **Data Field** of the **Data Table**, however, there is a better approach. When writing a **RegEx** pattern for an extractor it is possible to label elements of the pattern with groups. Groups in a **RegEx** pattern are referred to as sub-elements. If the sub-elements are named **exactly** the same (but underscores instead of spaces) as the **Data Field** child objects of the **Data Table**, one **Data Format** can populate all the **Data Fields** of the **Data Table**.

STEP 1 – ADD NEW DATA TYPE AND CHILD DATA FORMATS

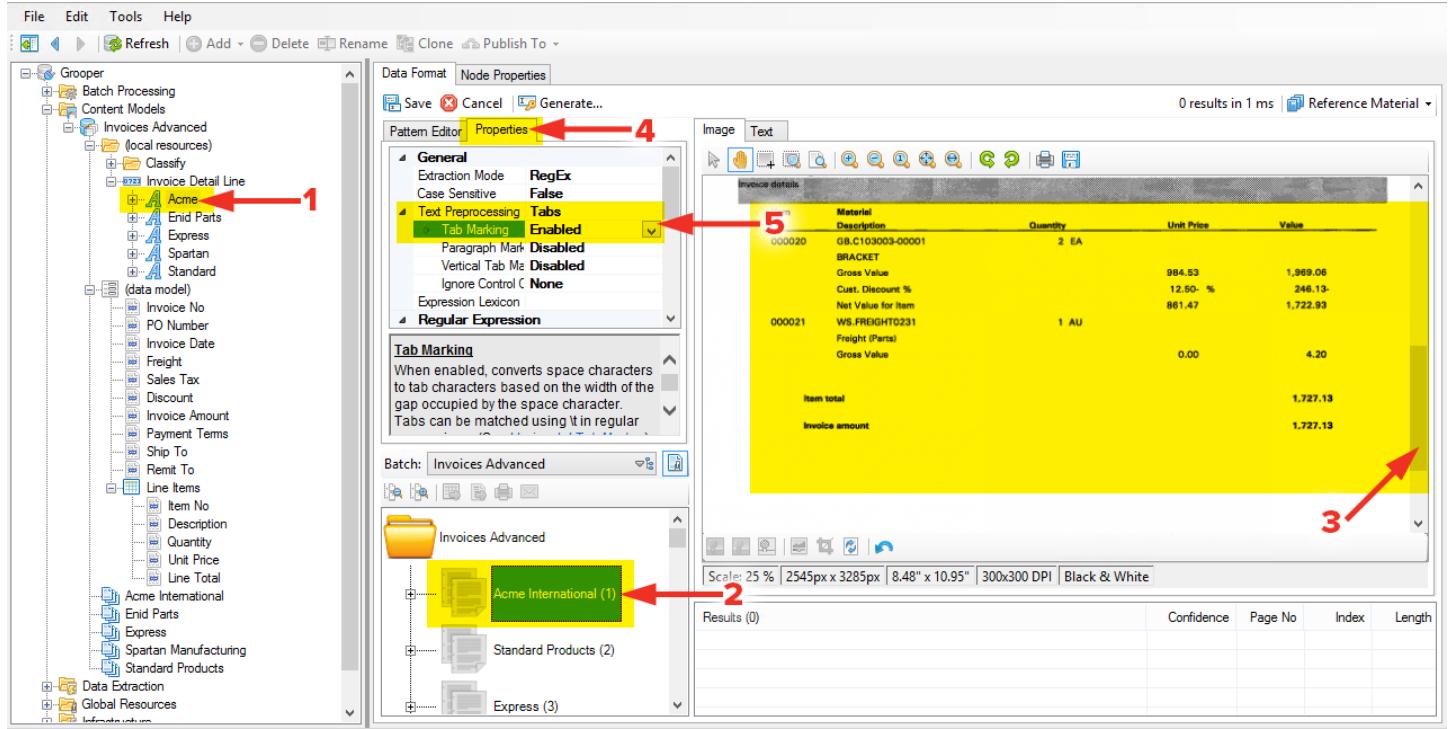
One **Data Type** will be passed as the **Row Extractor** of the **Data Table**, but it will need different formats to handle each of the different types of invoices. These different formats will be found with different **Data Format** objects.

(1) Add a **Data Type** to the **(local resources)** folder and name it **Invoice Detail Line**. **(2)** Add children **Data Format** objects to the **Invoice Detail Line Data Type** naming them after the different invoice providers: **Acme**, **Enid Parts**, **Express**, **Spartan**, **Standard** (the names of the **Data Formats** don't have to be named exactly as the **DocTypes** they're for.)



STEP 2 – ACME PATTERN 01

- (1) Select the **Acme Data Format** and (2) make sure the **Acme International (1)** document is selected in the **Batch Viewer**. (3) In the **Page Viewer**, scroll the page down to view the tabular information on the document. (4) Click on the **Properties Tab** and expand **Text Preprocessing**, then (5) set **Tab Marking** to **Enabled**.



STEP 3 – ACME PATTERN 02

Back in the **Pattern Editor** tab, type the following into the **Value Pattern** area:

```
[@Number]{6}\t(?!WS.FREIGHT)
(?<Item_No>[^|\t]*)\t
```

This is only the beginning of the pattern, but it's a good place to pause for a moment to get a better understanding of what it's doing.

- **[@Number]{6}** – The square brackets **[]** are defining a character set and the **@Number** is an expression variable built into **Grooper** that defines a range of number characters, but accounts for possible **OCR** errors. You may notice as you begin to type that **Grooper**'s built in **IntelliSense** will detect what you are typing. Press Enter to allow the **IntelliSense** to complete the expression for you. The number in curly braces **{6}** following the character set is a quantifier that defines how many of what in the character set in a row you're looking for.

• **\t** – The escape character **** followed by the **t** is defining a tab character. If you look at the **Text** tab you'll see the characters that were **OCR**'ed, including carriage returns **\r** and line new line feeds **\n**. Tab marking is off by default, and simple spaces are instead read. When tab marking is enabled (feel free to go back to the properties tab and look at the settings for tab marking, as there is more to it than simply enabling it) spaces between **OCR**ed characters that are larger than a defined threshold (based on font size) are read as tabs.

• **(?!WS.FREIGHT)** – This defines what is called a negative look ahead. Everything between the **(?!** and the **)** is considered as being unwanted following the previous pattern. So if any combinations of 6 number characters is followed by the string **WS.FREIGHT**, that string of 6 numbers will be considered invalid.

- The pilcrow ¶ character is not actually typed in this case, but in **Grooper** is automatically inserted when you return the line in your pattern. This allows you to write neat patterns by having them exist on more than one line.

- (?<Item_No>[^t]*)** – Strings of characters surrounded by parenthesis (**example**) are considered to be in a group. When the open parenthesis (is followed immediately by a question mark ? and then immediately by a less than < the group is being named. The string of characters following the less than < are naming the group, up until the greater than >. Therefore **(?<Item_No>)** is a group named **Item_No**.

Naming the group isn't useful if you're not defining what the group contains however. Everything between the greater than > and the close parenthesis) will be what is contained in the named group. In this case, **[^t]***. Starting a character set [] with a caret ^ tells the system what to NOT get. In this case, any non-tab character. The quantifier * outside the character set says get zero or more not-tab **[^t]** characters. After the capture group, a tab \t character is captured to continue the pattern.

- To recap: 6 number characters **@Number** {6} are followed by a tab \t, but not where the string **WS.FREIGHT** exists **(?!WS.FREIGHT)**. A named group called **Item_No** contains zero or more non-tab characters **(?<Item_No>[^t]*)** followed by a tab character \t.

The screenshot shows the Grooper interface with the following components:

- Left Sidebar:** Shows the project structure under "Grooper". It includes sections like "Batch Processing", "Content Models", "Invoices Advanced", "Invoice Detail Line", and various company models (Acme, Enid Parts, Express, Spartan, Standard).
- Pattern Editor (Top Right):** Displays the "Value Pattern" section with the following content:


```
1. [@Number]{6}\t(?:!WS.FREIGHT)\\t
2. (?<Item_No>[^t]*)\\t
```
- Properties (Top Right):** Shows "3 results in 6 ms" and a "Reference Material" link.
- Results (Bottom Right):** Contains three tabs: "Image", "Text", and "Table".
 - Text Tab:** Shows invoice details:

Currency	:	USD
Invoice amount	:	1,727.13
Payment terms	:	Net 30 days
Sales order no : 33241729		
Delivery number : 5515616		
Freight carrier :		
Tracking number : 122565E90301034108		
 - Table Tab:** Shows invoice details:

Item	Material Description	Quantity	Unit Price	Value
600026	08.C103003-00001	2 EA		
	BRACKET		984.53	1,969.06
	Gross Value			
	Cust. Discount %		12.50- %	246.13-
	Net Value for item		861.47	1,722.93
000021	WS.FREIGHT0231	1 AU		
	Freight (Partial)			
 - Results Tab:** Shows search results:

Results (3)	Confidence	Page No	Index	Length
017081Date	100 %	1	408	12
003405Invoice number	100 %	1	450	22
000020GB.C103003-00001	100 %	1	741	24

STEP 04 - ACME PATTERN 03

Return the line and enter the following pattern:

(?<Quantity>[@Number]{1,3})\s

This creates another named group called **Quantity** that looks for 1 to 3 number characters followed by a space.

STEP 05 – ACME PATTERN 04

Return the line and enter the following pattern:

[A-Z]{2}\r\n

This will capture any letter in the alphabet twice, followed by a carriage return and newline feed.

STEP 06 – ACME PATTERN 05

Return the line and enter the following pattern:

(?<Description>[^r]*)\r\n

This will create another named group called **Description** that will capture all non-carriage return characters, followed by a carriage return, and a new line feed.

STEP 07 – ACME PATTERN 06

Return the line and enter the following pattern:

([^t]*t)?

This creates a non-named group that will capture all non-tab characters, followed by a tab. The question mark quantifier (zero or one) after the group will make it optional.

STEP 08 – ACME PATTERN 07

Return the line and enter the following pattern:

(?<Unit_Price>[@Number.,]*)\t

This creates a group named **Unit_Price** that will capture zero or more number characters, periods, and commas, followed by a tab.

STEP 09 – ACME PATTERN 08

Return the line and enter the following pattern:

```
(?<Line_Total>[@Number.,]*)
```

This will create the final named group for this pattern called **Line_Total**. It will capture zero or more number characters, periods, and commas, then stop.

The screenshot shows the Grooper ACE software interface. On the left, there is a navigation tree with categories like Grooper, Batch Processing, Content Models, Invoices Advanced, Data Extraction, Global Resources, Infrastructure, and Reports. Under Invoices Advanced, there are sub-folders for Classify, Invoice Detail Line, and several Acme-related models (Acme, End Parts, Express, Spartan, Standard). The main workspace is divided into several panes:

- Pattern Editor:** Shows a yellow-highlighted "Value Pattern" block containing the regular expression: `1 [@Number] {6} \t (? WS.FREIGHT) \t 2 { > Item_No > [^\t] * } \t 3 { > Quantity > [@Number] {1,3} } \s \t 4 [A-Z] {2} \r \n \t 5 { > Description > [^\r] * } \r \n \t 6 { [^\t] * \t } ? \t 7 { > Unit_Price > [@Number.,]* } \t \t 8 { > Line_Total > [@Number.,]* }`
- Image/Text:** Shows a preview of an invoice document with various fields like Customer contact, Currency, Invoice amount, Payment terms, and tracking numbers.
- Invoice details:** A table showing invoice items with columns: Item, Material Description, Quantity, Unit Price, and Value. One item is highlighted in green: "000026 GB.C103003-00001 BRACKET 2 EA".
- Results:** A table titled "Results (1)" showing the output of the search. It includes columns: Confidence, Page No, Index, and Length. The single result is: "000020GB C103003-00012 EABRACKETGross Value984.531,969.06".

STEP 10 – REMAINING PATTERNS

The remaining **Data Formats** that are child objects of the **Invoice Detail Line Data Type** will all use **tab marking** and have the following patterns:

Enid Parts:

```
(?!020-0027)
(?<Item_No>[A-Z0-9.\-]+)\t
([A-Z]{1,2}\t)?
(?<Description>[^\\r]+)\\r\\n
[@Number.,]+\\t
(?<Quantity>[@Number.,\\s]+)\\t
[A-Z]{1,4}\\t
(?<Unit_Price>[@Number.,]+)\\s
[A-Z\\s/]+\\t
(?<Line_Total>[@Number.]+)
```

For this pattern use a look ahead of:

\n

Express:

```
[0-9]{1,3}[\s\-]
(?<Item_No>[0-9]{4}[0-9A-Z]{3,4})\\t
(?<Description>[^\\t]+)\\t[0-9]{1,4}\\t
(?<Quantity>[0-9]{1,4})\\t[0-9]{1,4}\\t
(?<Unit_Price>[0-9.,]+)\\t
(?<Line_Total>[0-9.,]+)
```

Spartan:

```
\\d{5}\\s[A-Z]\\s
(?<Item_No>\\d{5})\\t
\\d{1,4}\\t\\d{1,4}\\t
(?<Quantity>\\d{1,4})\\t
(?<Unit_Price>[\\d.,]+)\\t
([A-Z]{1,5}\\t)?
(?<Line_Total>[\\d.,]+)
[^\\n]+\\n
(?<Description>[^\\t\\r]+)
```

Standard:

```
([0-9]{6}\\s)?
(?<Item_No>[0-9A-Z]{5})\\s
(?<Description>[^\\t\\r]*))\\t
(?<Quantity>[0-9]{1,5})\\s
(?<Unit_Price>[0-9.,]+)\\s
(?<Line_Total>[0-9.,]+)
```

For this pattern use a look ahead of:

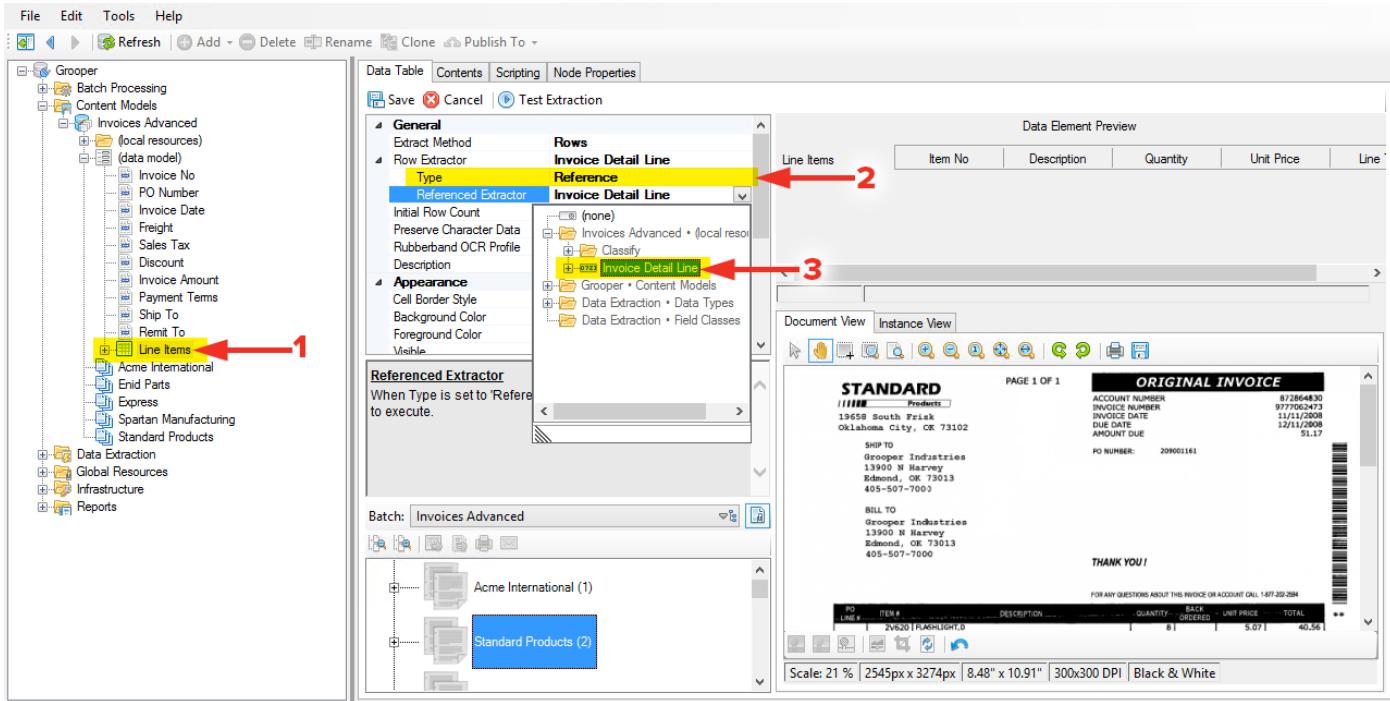
\n

And a look behind of:

\r

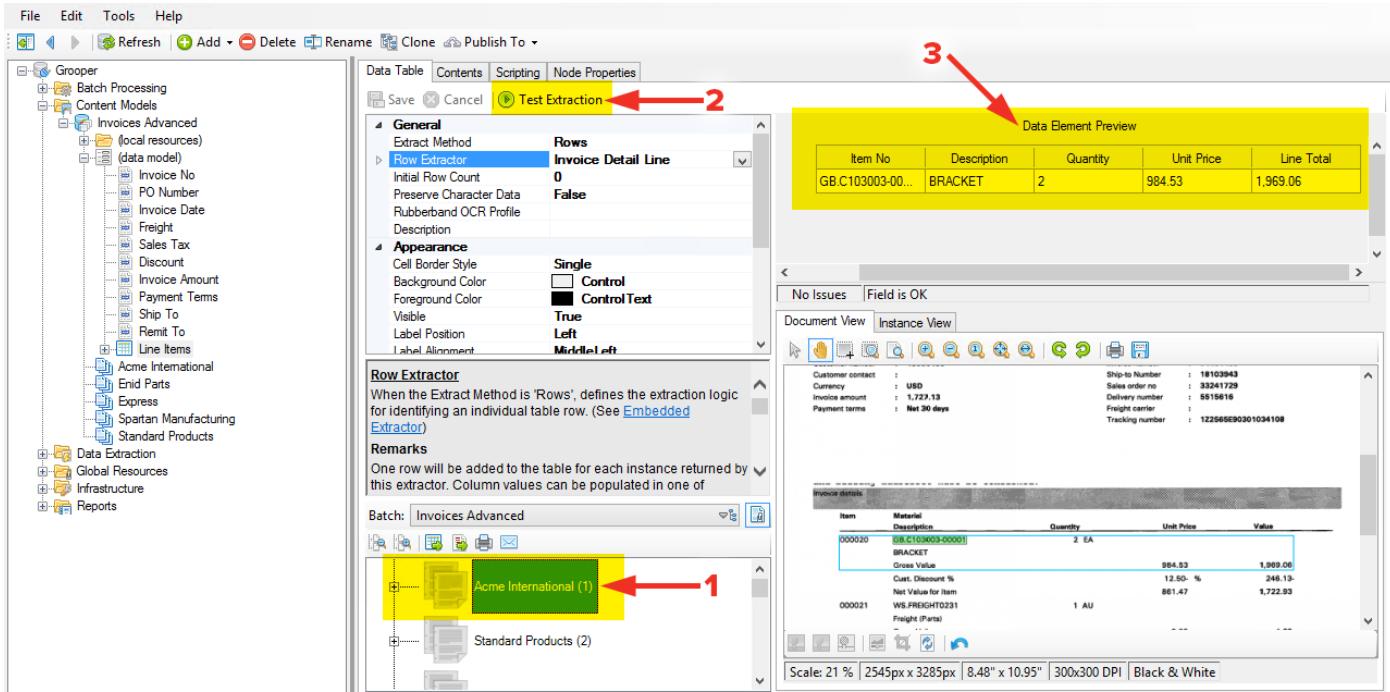
STEP 11 – SETTING THE ROW EXTRACTOR

With the Data Formats for our **Invoice Detail Line Data Type** complete, it's time to reference this **Data Type** within the **Data Table**. Expand the **(data model)** and **(1)** select the **Line Items Data Table**. **(2)** Set the Row Extractor as a **Type > Reference** and **(3)** select the **Invoice Detail Line** as the **Data Type** for the **Referenced Extractor**.



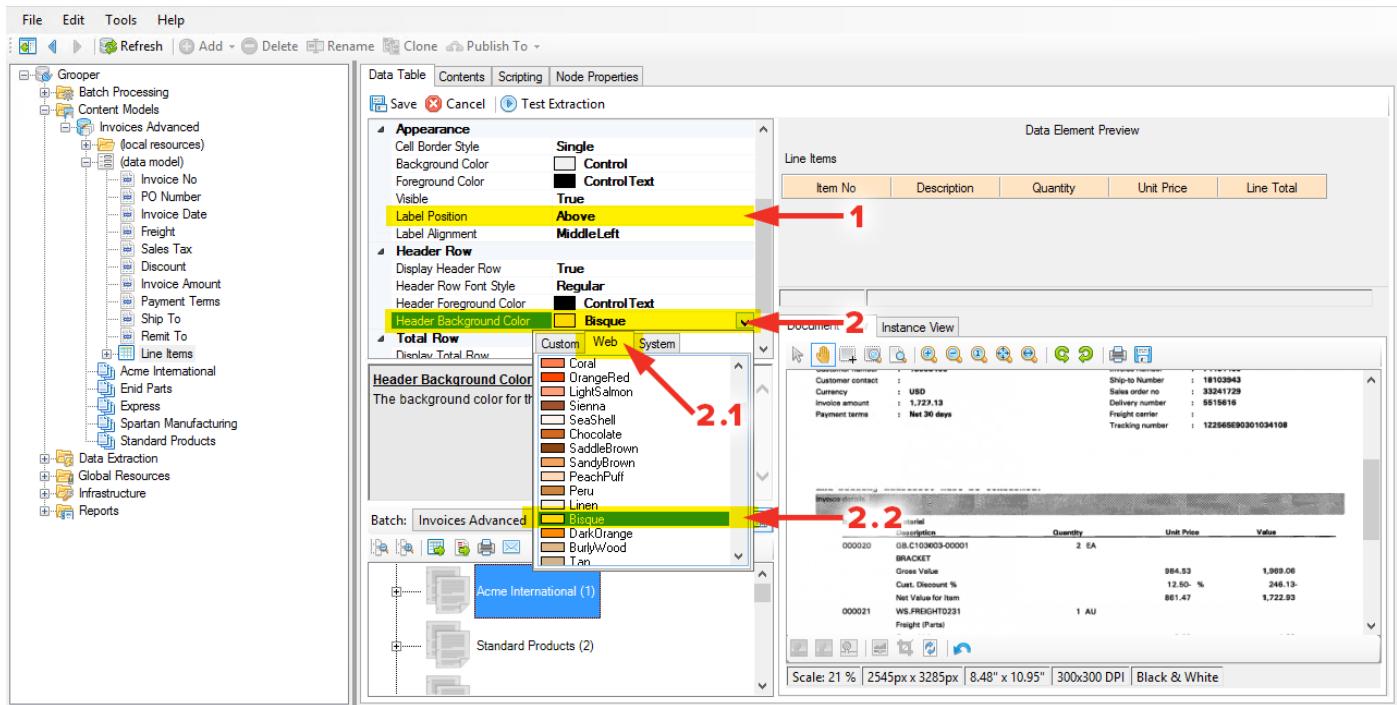
STEP 12 – TESTING EXTRACTION

(1) Select any document from the **Batch Viewer**, and **(2)** click the **Test Extraction** button. **(3)** In the **Data Element Preview**, the results of the test extraction should be visible and displayed in a table.



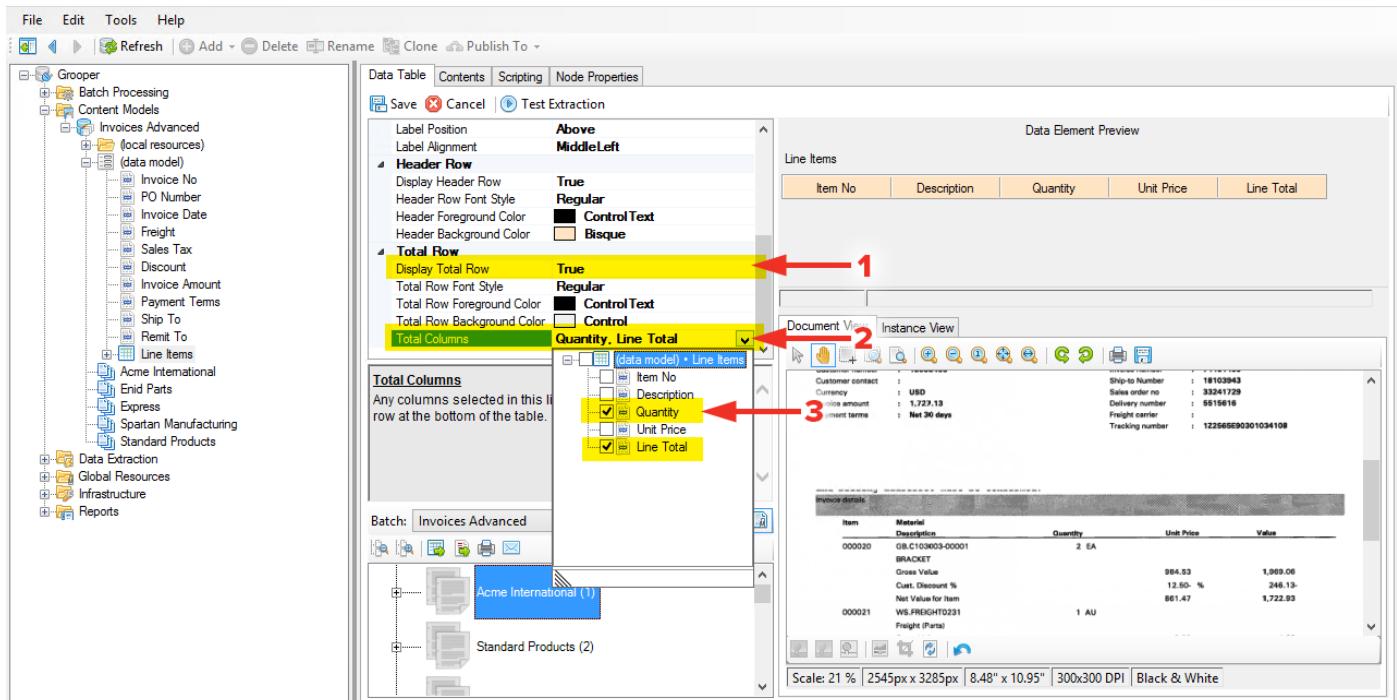
STEP 13 – ADJUSTING PROPERTIES OF THE DATA TABLE

in the Appearance properties section, (1) set the Label Position to Above. In the Header Row section, (2) set Header Background Color to Bisque.



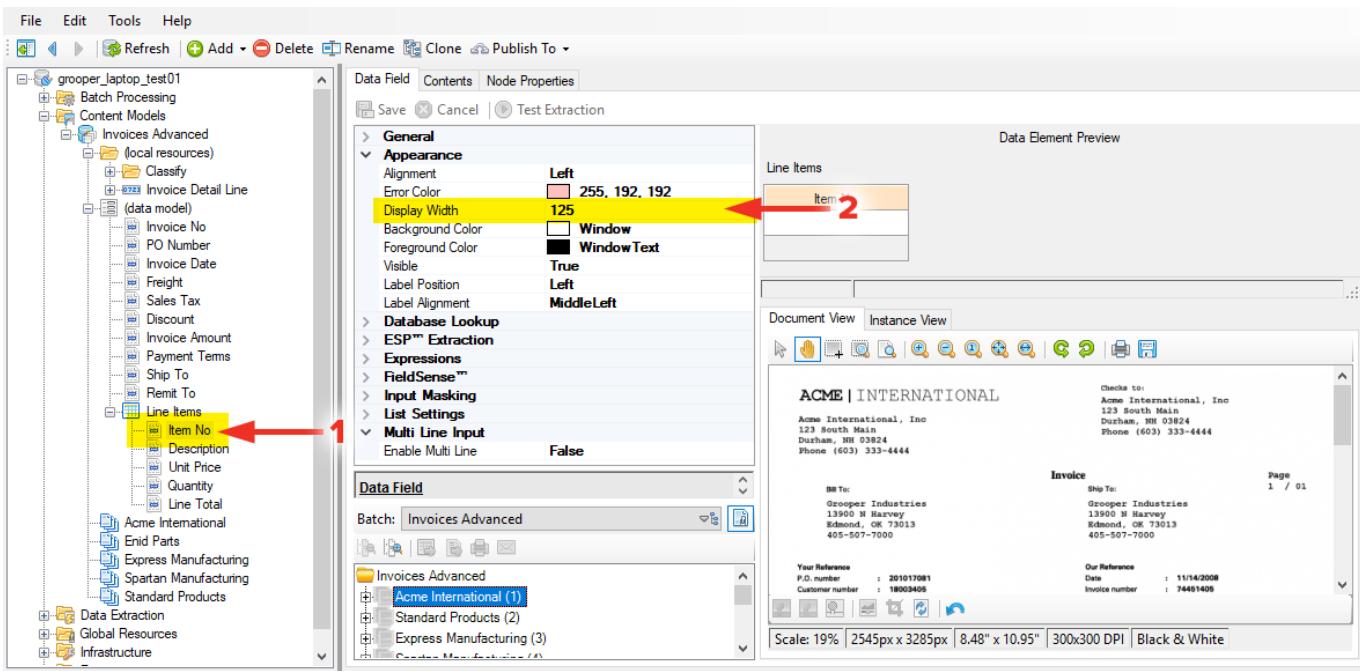
STEP 14 – ENABLE TOTAL ROW

The final section of the Data Table, Total Row, houses settings for a dynamically generated row that will create sum totals of selected columns. (1) Set Display Total Row to True, and (2) select Quantity and Line Total as the Total Columns. Feel free to Test Extraction again to observe the changes made.



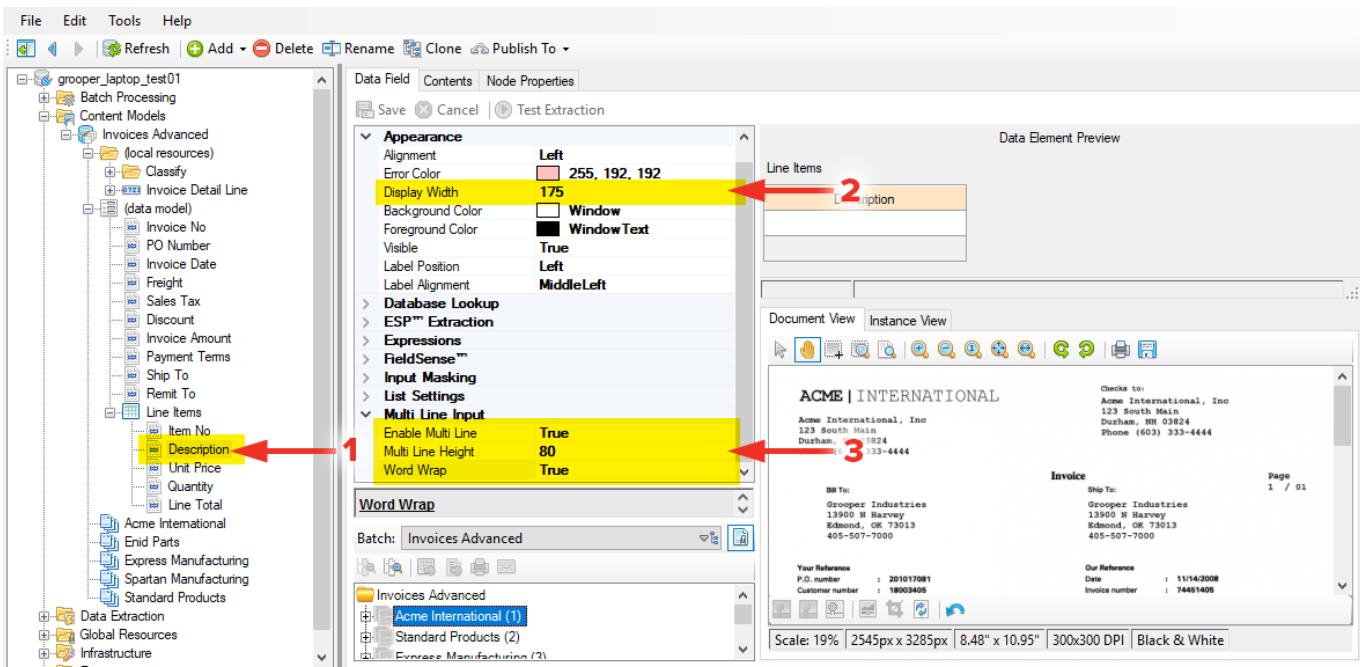
STEP 15 – TABLE APPEARANCE ADJUSTMENTS

The fields of the table are uniform in their appearance, without respect to what they're capturing, so some information has too much room, while others (like **Description**) don't have enough. **(1)** Click the **Item No Data Field** and **(2)** set its **Display Width** to **125**. Do the same for **Unit Price**, **Quantity**, and **Line Total** with values of **75**, **100**, and **85** respectively.



STEP 16 – APPEARANCE ADJUSTMENTS CONTINUED

(1) Select the **Description Data Field** and **(2)** set its **Display Width** to **175**. **(3)** Set **Enable Multi Line Height** to **True**, **Multi Line Height** to **80**, and **Word Wrap** to **True**. With these adjustments made, feel free to **Test Extraction** on the **Line Items Data Table** to see the results.



THE GROOPER FIELD CLASS EXTRACTOR

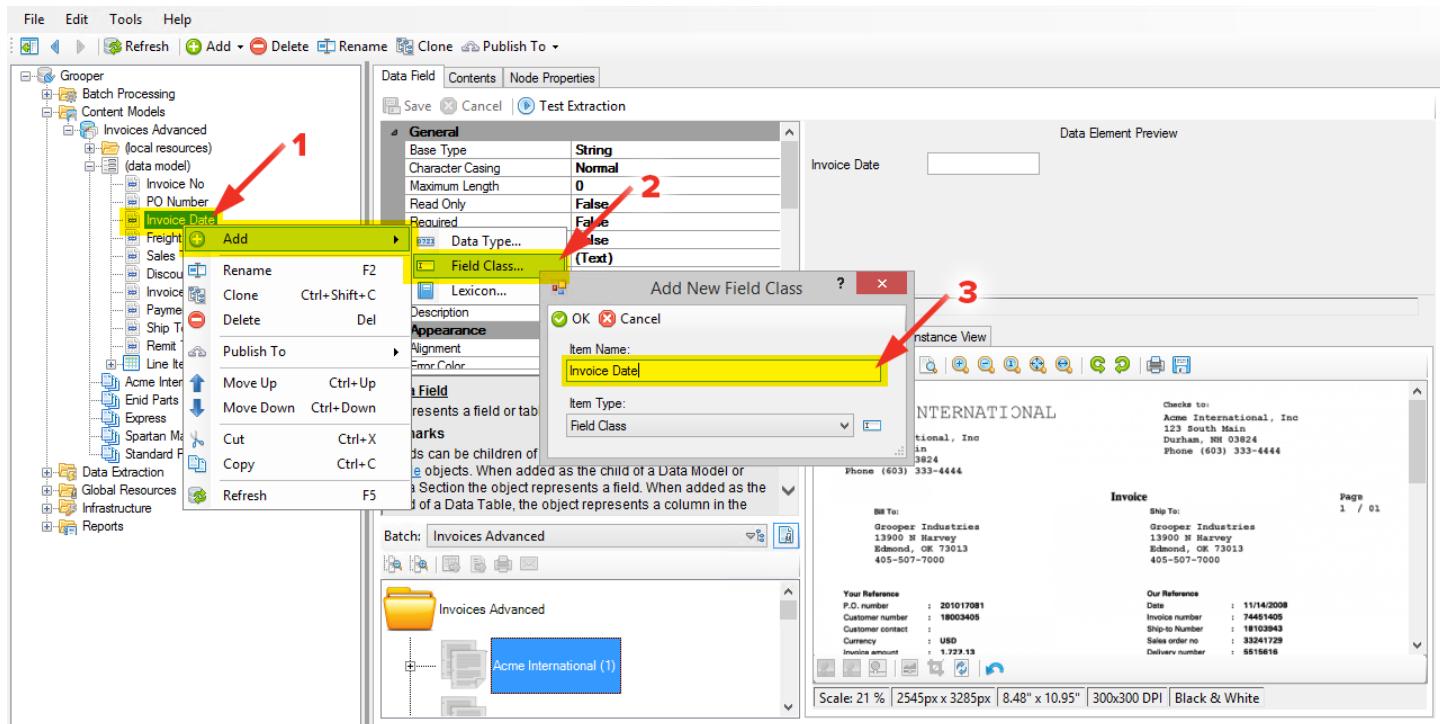
Throughout all the training that's been done so far, only two of the three extractors in **Grooper** have been leveraged: the **Data Type** and the **Data Format**. The final extractor to learn about is the **Field Class**. The **Field Class** is one of **Grooper**'s most powerful tools and steps up where simple patterns can fail. The reason for this is it is not reliant on uniform structure of a document to perform its extraction. It instead leverages the context around a desired value to understand what is to be extracted. This context is understood because of an end user training the **Field Class** to understand what features around a desired value it considers positively, or negatively.

SETTING UP A FIELD CLASS FOR THE FIRST TIME

The best way to learn about the **Field Class** is to use one. This process will begin with the **Invoice Date Data Field**, and step by step walk through creating (or referencing) all the requisite components to get accurate, consistent extraction.

STEP 1 – ADDING A FIELD CLASS TO THE INVOICE DATE DATA FIELD

- (1) Select the **Invoice Date Data Field** then (2) right-click and select **Add > Field Class...** (3) Name it **Invoice Date**.



STEP 2 – SETTING THE VALUE EXTRACTOR

The first thing to understand about a **Field Class** is that it leverages two extractors to accomplish its end goal. To keep things simple in the beginning extractors will be referenced, instead of built from scratch.

The first property to set for a **Field Class** is the **Value Extractor**. This is the extractor that will define what information is to be found, and in the case of this **Data Field**, a date.

- (1) Set the **Value Extractor Type** to **Reference**, and the **Referenced Extractor (2)** to **Data Extraction • Data Types > (system) > Date (OCR)**.

As a side point, feel free to explore this **Data Type** on your own to discover how it is returning the values it does.

The screenshot shows the Grooper ACE application interface. On the left, there's a sidebar with tabs like 'File', 'Edit', 'Tools', and 'Help'. Below these are buttons for 'Refresh', 'Add', 'Delete', 'Rename', 'Clone', and 'Publish To'. The main area has tabs for 'Field Class', 'Weightings', 'Contents', and 'Node Properties'. Under 'Field Class', there are sections for 'General', 'Value Extractor' (set to 'Date (OCR)'), 'Type' (set to 'Reference'), and 'Referenced Extractor' (set to 'Date (OCR)'). A red arrow labeled '1' points to the 'Run Extraction' button in the toolbar. Another red arrow labeled '2' points to the 'Date (OCR)' entry in the list of extractors. The central pane displays a document titled 'ACME | INTERNATIONAL' with contact information for Acme International, Inc. The right pane shows an 'Invoice' section with bill-to and ship-to details for Grooper Industries. At the bottom, there are tables for 'Feature Occurrences' and 'Feature' with columns like Count, CWF, CTC, ID, CF, TF, IDF, and Weight.

STEP 3 – UNDERSTANDING THE FEATURE EXTRACTOR

The second extractor that is set for a **Field Class** defines what around the desired value defines it. If there are multiple date patterns on an invoice, it is clear which date is the desired one due to contextual information surrounding that value.

(1) Field Classes default properties have the system **Data Type – nGrams 1-3** set as the **Feature Extractor**. This will work fine for now. Save the changes made and click the **Run Extraction** button on the **Acme International (1)** document.

The screenshot shows the Grooper ACE software interface. On the left, the 'Field Class' configuration window is open, specifically the 'General' tab. Under 'Value Extractor', 'Type' is set to 'Date (OCR) Reference'. Under 'Feature Extractor', 'Type' is set to 'nGrams 1-3 Reference'. A red arrow points from the text '1.1' to the 'nGrams 1-3' option in the dropdown menu. In the center, the 'Image View' shows an invoice from 'ACME | INTERNATIONAL'. The invoice details include 'Acme International, Inc' with address '123 South Main, Durham, NH 03824' and phone '(603) 333-4444'. Below this, the 'Invoice' section shows 'Bill To: Grooper Industries' with address '13900 N Harvey, Edmond, OK 73013' and phone '405-507-7000'. The 'Ship To:' section is identical. At the bottom of the central pane, it says 'Scale: 33 % | 2545px x 3285px | 8.48" x 10.95" | 300x300 DPI | Black & White'. On the right, the 'Text View' pane displays the extracted data in a table format:

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight

STEP 4 – UNDERSTANDING CONTEXT SCOPE AND ZONES

With extraction run there will be new information presented that needs to be explained. (1) First off, the **Value Extractor** will return a value and that will be displayed in the **Value List View**, as well as draw a green box on the value in the **Page Viewer**. (2) The default **Zonal Context Scope** will define (3) 2 items for the **Context Zones** which are rectangles drawn around the value. **Grooper** will look within the boundaries of these rectangles to find features returned from the **Feature Extractor**. (4) Features it finds will have a blue box drawn around them.

The screenshot shows the Grooper ACE software interface with several windows open:

- Value List View (Left):** Shows a table with one row:

Value	Page	Confidence
11/14/2008	1	0.0000 %

 A red arrow labeled "1" points to the "Value" column of the first row.
- Page Viewer (Center):** Displays an invoice document with various fields like Bill To, Your Reference, and Our Reference. A yellow box highlights the date "11/14/2008" under "Our Reference Date". A red arrow labeled "1.1" points to this highlighted date.
- Feature Occurrences (Bottom Right):** Shows a table of extracted features:

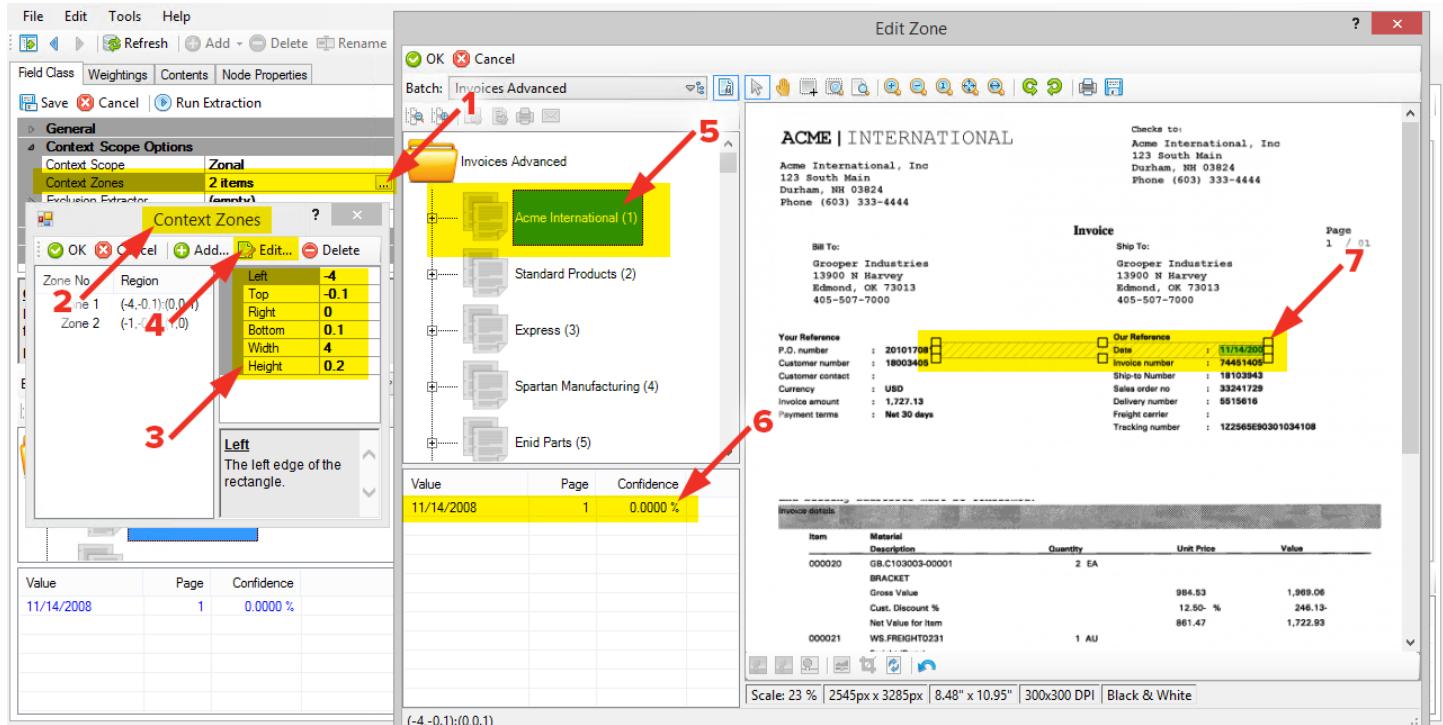
Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
date	1	0	0	1.000000	1.000000	1.000000	1.000000	1.000000

 A red arrow labeled "4.1" points to the "Weight" column of the "date" row.
- General Settings (Top Left):** Shows context scope options set to "Zonal" and "2 items". A red arrow labeled "3" points to the "Context Zones" setting.
- Image View (Top Center):** Shows a toolbar with various icons for image processing.

STEP 5 – EDITING A CONTEXT ZONE

With the **Context Zones** property selected, (1) click the ellipsis button (2) to bring up the **Context Zones** window. (3) From here you can select a zone and manually adjust the boundary numeric values, or click (4) **Edit...** which will bring up the **Edit Zone** window. In this window, (5) select a document (6) and a value, and (7) use the handles of the marquee to adjust the shape.

There is no specific need to adjust the **Context Zones** for now, so you can leave them defaulted. This was merely an exercise in understanding how one can adjust the **Context Zones**.



STEP 6 – TRAINING A FEATURE INSTANCE

The **Field Class** now must be trained as to what feature it will consider positively so it can identify an appropriate value. For the **Acme International (1)** document, there is only one date value returned, therefore only the feature, **date**, will be considered. (1) In the **Value List View**, select and right click the only listed value and select **Train As Positive**. Keep in mind this positive training happened for (2) the feature **date**, not for the value **11/14/2008**.

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
date	1	0	0	1.000000	1.000000	1.000000	1.000000	1.000000

STEP 7 – CHECKING WEIGHTINGS

Click on the **Weightings** tab. This will display the features that have been trained. When a feature is highlighted in a blue box, and the corresponding value trained against it, the value(s) within boxes will be stored in the **Positive Instances**. These positive instances will create a weight by which a confidence is built to understand which value to consider on future documents. If other values are present, and as a result, other features highlighted by those values, but those are not selected to be trained positively, those features will be stored as **Negative Instances**. For the sake of this simple example with the **Acme International (1)** document, only one word was stored positively.

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
date	1	1	0	1.000000	1.000000	1.000000	1.000000	1.000000

STEP 8 – CHECKING THE NEXT DOCUMENT

Click the **Standard Products (2)** document in the Batch Viewer. (1) Notice that several date values are being returned by the **Value Extractor**, but only one is listed with confidence (100% in this case.) (2) The feature **date** was found within a context zone around the extracted value, and since this one word has been trained positively, **Grooper** is 100% confident this value is what we want.

The screenshot shows the Grooper interface with the following details:

- Left Panel (Batch Viewer):**
 - Field Class: General
 - Batch: Invoices Advanced
 - Selected Document: Standard Products (2)
 - Table: Value, Page, Confidence

11/11/2008	1	100.0000 %
11/11/2008	1	0.0000 %
12/11/2008	1	0.0000 %
11/11/2008	1	0.0000 %
- Right Panel (Image View):**
 - Document Preview: PLEASE DETACH THIS PORTION AND RETURN WITH YOUR PAYMENT.
 - Text View: BILL TO: Grooper Industries, 13900 N Harvey, Edmond, OK 73013, 405-507-7000; REMIT TO: Standard Products, 19658 South Frisk, Oklahoma City, OK 73102, 800-555-2121.
 - Invoice Details: ACCOUNT NUMBER 872864830, DATE 11/11/2008, FILE NUMBER 7062473, AMOUNT DUE 51.17.
 - Annotations: A red arrow labeled '1' points to the first row in the table where the confidence is 100%. Another red arrow labeled '2' points to the 'date' column in the Feature Occurrences table.

STEP 9 – FINDING ANOTHER CORRECT VALUE

While this value is the desired value, there is another instance of a correct value on the document. (1) Select the second **11/11/2008** value, which will be an instance at the top of the document. This is also a value we want, but the feature (in this case a two-word phrase) **invoice date** hasn't been trained positively.

The screenshot shows the Grooper interface with the following details:

- Left Panel (Batch Viewer):**
 - Field Class: General
 - Batch: Invoices Advanced
 - Selected Document: Standard Products (2)
 - Table: Value, Page, Confidence

11/11/2008	1	100.0000 %
11/11/2008	1	0.0000 %
12/11/2008	1	0.0000 %
11/11/2008	1	0.0000 %
- Right Panel (Image View):**
 - Document Preview: STANDARD Products, 19658 South Frisk, Oklahoma City, OK 73102, PAGE 1 OF 1, ORIGINAL INVOICE.
 - Text View: SHIP TO: Grooper Industries, 13900 N Harvey, Edmond, OK 73013, 405-507-7000; BILL TO: Grooper Industries, 13900 N Harvey, Edmond, OK 73013, 405-507-7000.
 - Annotations: A red arrow labeled '1.1' points to the first row in the table where the confidence is 100%.

STEP 10 – TRAINING TWO FEATURE INSTANCES

Both features, **date** and **invoice date** need to be positively reinforced, so both should be selected and trained positively. Keep in mind, features found on the document but not selected at the time of positive training will be considered negatively, and at this point, training the feature **date** negatively would throw off our weightings.

Value	Page	Confidence	Count	CWF	CTC	ID	CF	TF	IDF	Weight
11/11/2008	1	100.0000 %								
11/11/2008	1	0.0000 %								
12/11/2008	1	0.0000 %								
11/11/2008	1	0.0000 %								

STEP 11 – CHECKING WEIGHTINGS ONCE AGAIN

Click the **Weightings** tab again. Notice now there are two **Positive Instances**: **date** and **invoice date**. Notice also that the feature **date** has a count of two, while **invoice date** only has a count of one. See also that there are now **Negative Instances**. These were features that were captured within the **Context Zones** of the other listed values for this document, but weren't selected when the positive training was invoked.

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
date	2	2	0	1.000000	1.000000	0.666667	0.397940	0.265293
invoice date	1	1	0	1.000000	1.000000	0.333333	0.698970	0.232990

Item	Count	Avg Distance	Instance Count
due date	1	0	1
number of pkgs	1	0	1
date shipped	1	0	1

STEP 12 – REINFORCING TRAINING

(1) Select the Express (3) document then (2) select the 12/2/2008 value in the List View with ~65% and right-click Train As Positive. This will reinforce the feature invoice date as a Positive Instance, and add features around the other listed values as Negative Instances.

EXPRESS

Invoice Number: 16862865
Invoice Date: 12/2/2008
Purchase Order: 710015038

Bill To: Grooper Industries
13900 N Harvey
Edmond, OK 73013
405-507-7000

Invoice Total: \$216.80
Date Due: 1/1/2009
Mail Payment To: Express Manufacturing
12333 N Dallas Tollway
Dallas, TX 75022

Value	Page	Confidence
12/2/2008		Train As Positive F3
1/1/2009		Train As Negative
12/12/2008		Inspect Instance...
12/2/2008		Combine

STEP 13 – ZONAL PROBLEMS

(1) Click the Spartan Manufacturing (4) document. (2) Notice the low confidence of the correct value in the List View. (3) This low confidence is due to the erroneous features being captured by the context zones surrounding the desired value. Considering this, the Context Zones need to be adjusted to mitigate capturing unwanted features.

Spartan

Manufacturing
12 West Laguna Dr
Irvine, CA 92612

Phone: (800) 111-22222
Fax: (308) 333-1182

INVOICE

SHIP TO (NAME AS SOLD TO UNLESS SHOWN)
Grooper Industries
13900 N Harvey
Edmond, OK 73013

INVOICE DATE
11/10/08 ORIGINAL
INVOICE NUMBER
MA20-552100

PO / RELEASE NUMBER
201011309

REMIT TO:
Spartan Manufacturing
12 West Laguna Dr
Irvine, Ca, 92612

Value	Page	Confidence
11/10/2008	1	21.9533 %
11/10/2008	1	0.0000 %
11/10/2008	1	0.0000 %

STEP 14 – ZONAL ADJUSTMENTS

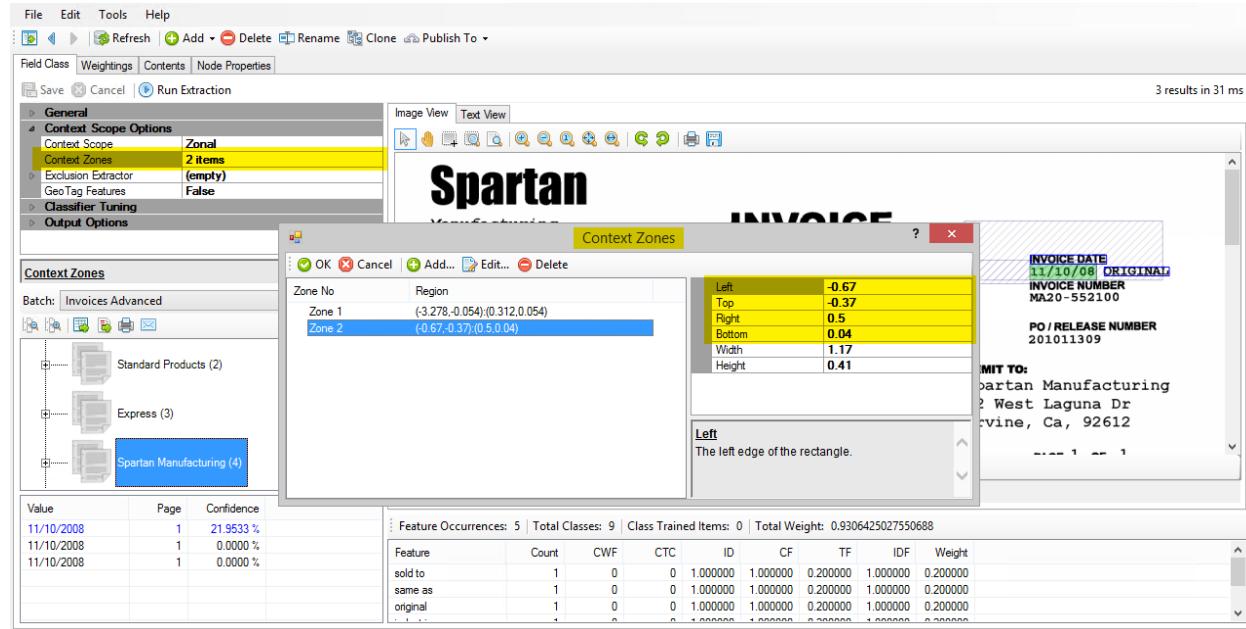
Bring up the **Context Zones** window and enter the following values:

Zone 1 – Left: **-3.278** Top: **-0.054** Right: **0.312** Bottom: **0.054**

Zone 2 – Left: **-0.67** Top: **-0.37** Right: **0.5** Bottom: **0.04**

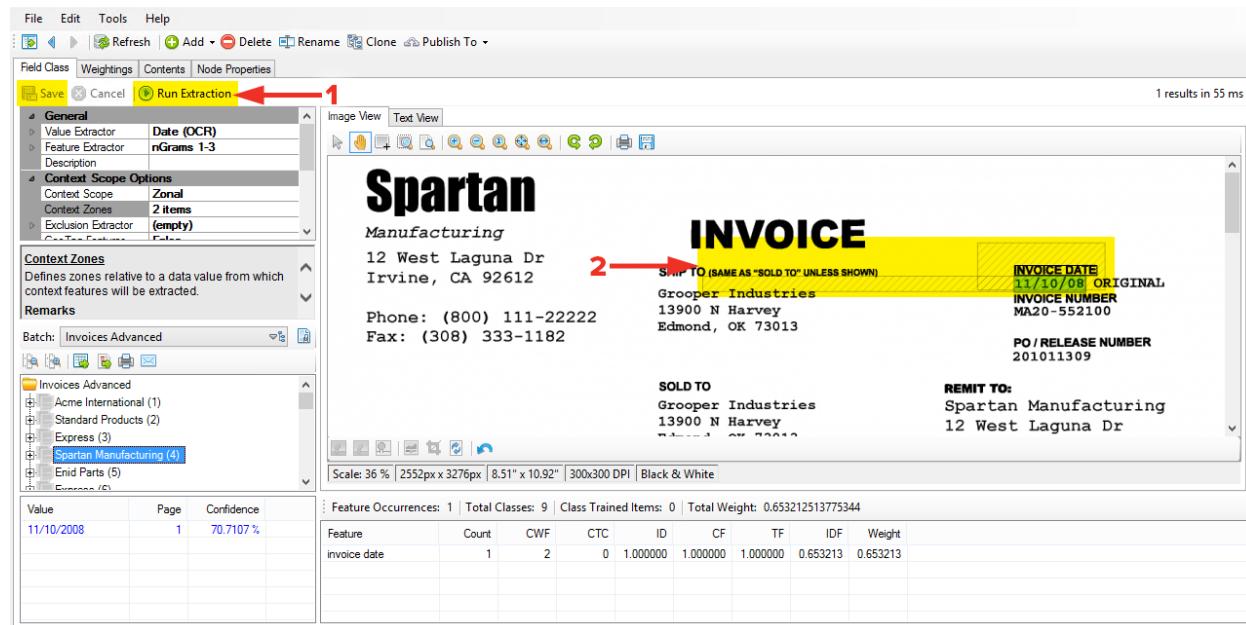
Click **OK** to accept the changes and close the window.

These values will adjust the **Context Zones** such that the outer most boundaries of the boxes are not touching the center point of the blue boxes that are drawn around the unwanted features. Keep this in mind when adjust **Context Zones**. You only need your zones to touch (or NOT in this case) the mid-points, not the entire word.



STEP 15 – CHECKING ZONAL ADJUSTMENTS

(1) Click **Save** and **Run Extraction**. (2) Notice the **Context Zones** are no longer highlighting unwanted features, and the **Feature List View** is returning a desired feature. This instance will not need to be trained.



STEP 16 – UNDERSTANDING WEIGHTINGS FURTHER

(1) Click the [Enid Parts \(5\)](#) document. (2) The correct value will be listed at the top of the [Value List View](#), but you should notice a couple of other date values listed with non-zero values for their confidence. (3) Select one of the ~38% values, and check the [Feature List View](#) and observe that a positively trained feature, in this case `date`, is being found within a [Context Zone](#), but also an untrained feature, `order`. Due to this, **Grooper** is less confident this is a correct value, which is correct for this case.

Value	Page	Confidence
12/5/2008	1	70.7107 %
11/11/2008	1	38.6701 %
12/3/2008	1	38.6701 %

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
order	1	0	0	1.000000	1.000000	0.500000	-1.000000	0.500000
date	1	2	0	1.000000	1.000000	0.500000	0.653213	0.326606

STEP 17 – SETTING MINIMUM CONFIDENCE

With a well-established confidence of ~70%, unwanted returned values can be eliminated by setting an acceptance threshold. Set the [Minimum Confidence](#) property to **70%**. Any value results returned that have a confidence less than this will be dropped. Feel free to scroll through and select documents from the [Batch Viewer](#) and see the results of this trained [Field Class](#).

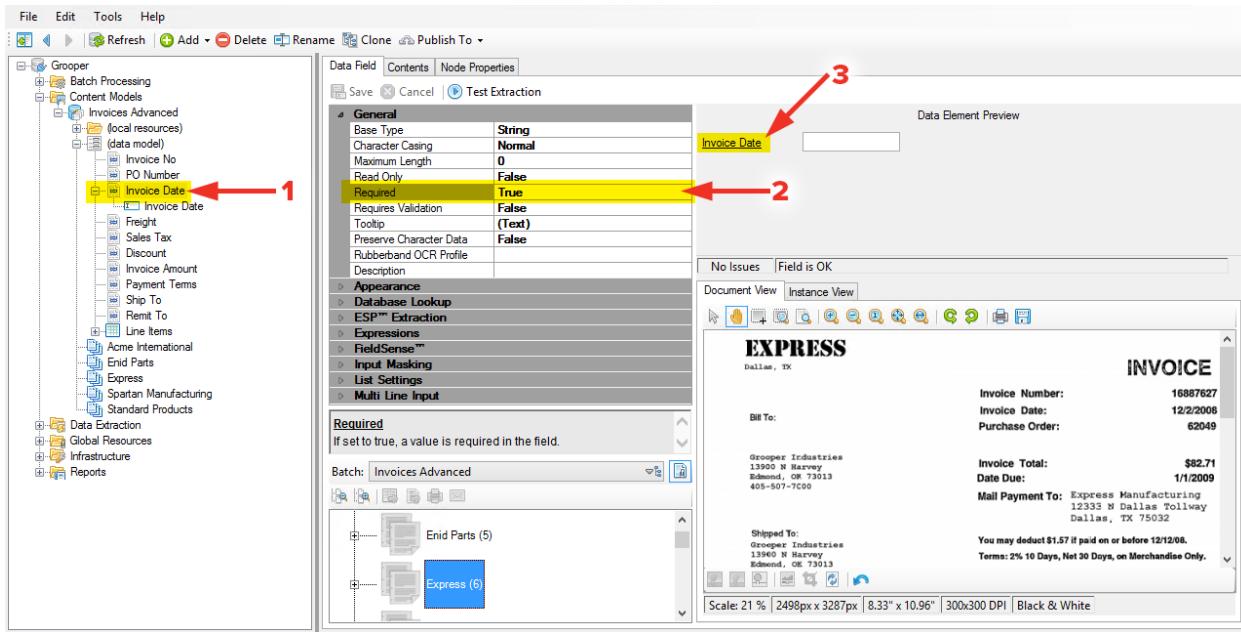
Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
invoice date	1	2	0	1.000000	1.000000	1.000000	0.653213	0.653213

SETTING UP THE INVOICE DATE DATA FIELD

The extractor for the **Invoice Date Data Field** has been built, but it needs to be applied to the **Data Field** and its properties adjusted.

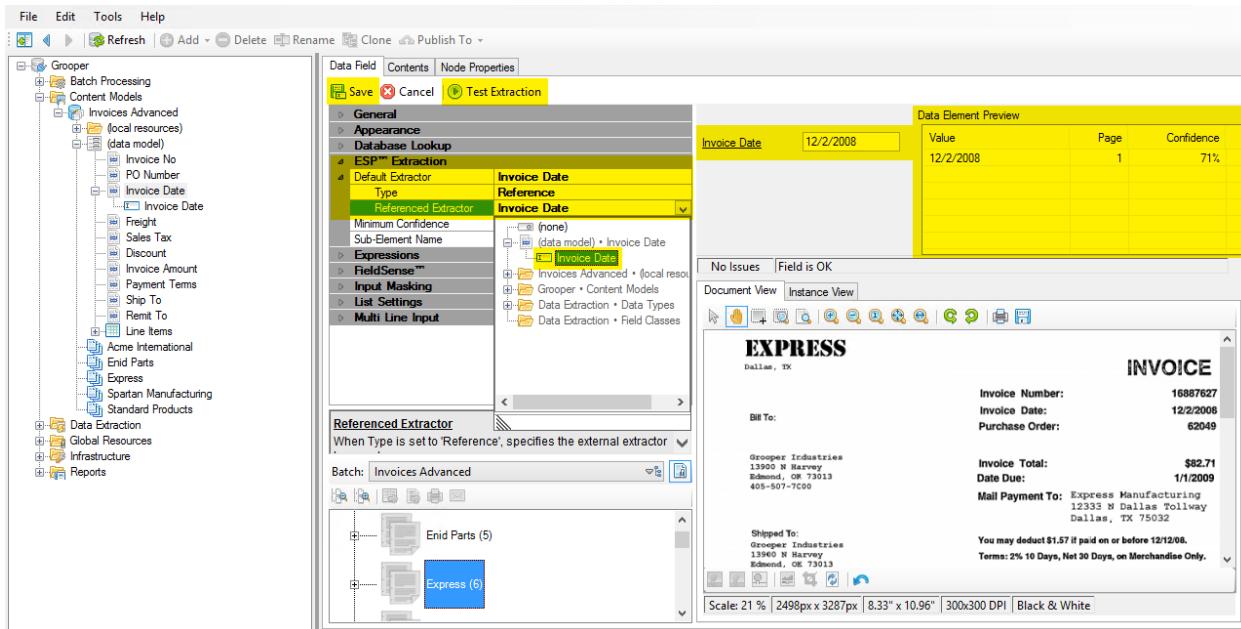
STEP 1 – SETTING THE DATA FIELD TO BE REQUIRED

(1) Select the **Invoice Date Data Field** and in the **General** section, (2) set its **Required** property to **True**. (3) Notice the name of the field become underlined.



STEP 2 – APPLYING THE EXTRACTOR

In the **ESP Extraction** section, set the **Default Extractor Type** to **Reference**. Point the **Referenced Extractor** property to the **Invoice Date Field Class**. Save and **Test Extraction** to see results.

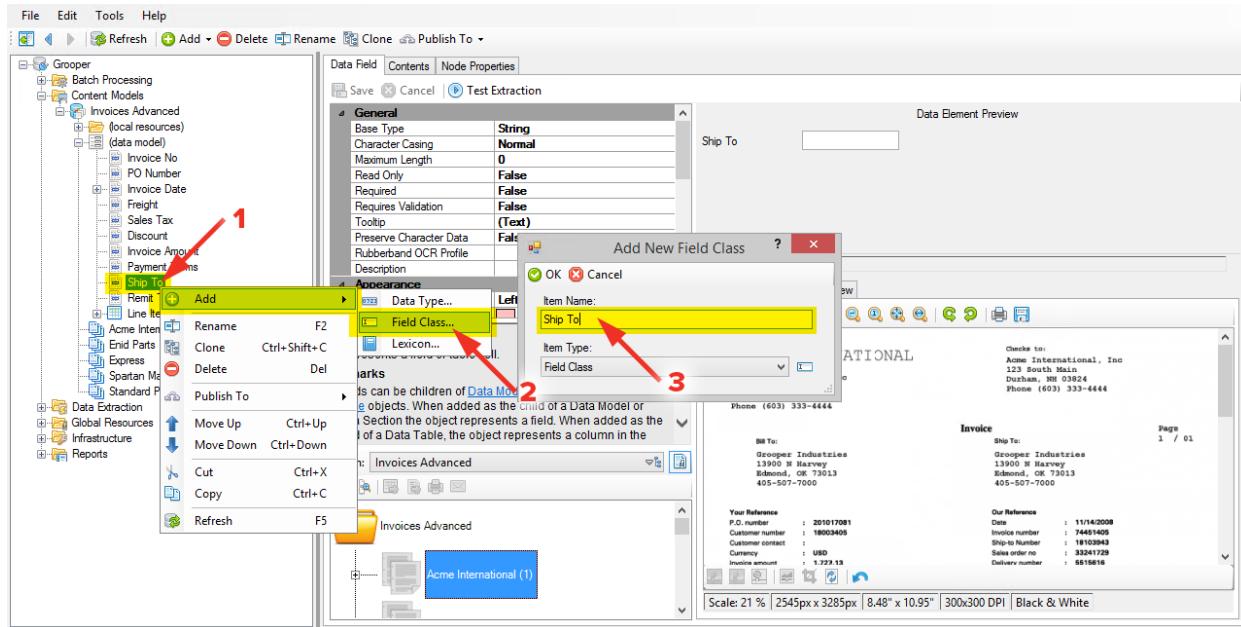


THE NEXT FIELD CLASS – BUILDING A FEATURE EXTRACTOR

Let's continue working with the [Field Class](#), but this time build out one of the two extractors it leverages. This [Field Class](#) will be looking for the [Ship To](#) address, and will use a [\(system\) Data Type](#) for the address pattern (the [Value Extractor](#)). The [Feature Extractor](#) will use a simple [Data Type](#) that will look for specific word patterns, but return a single value each time. The usefulness of this will be recognized soon.

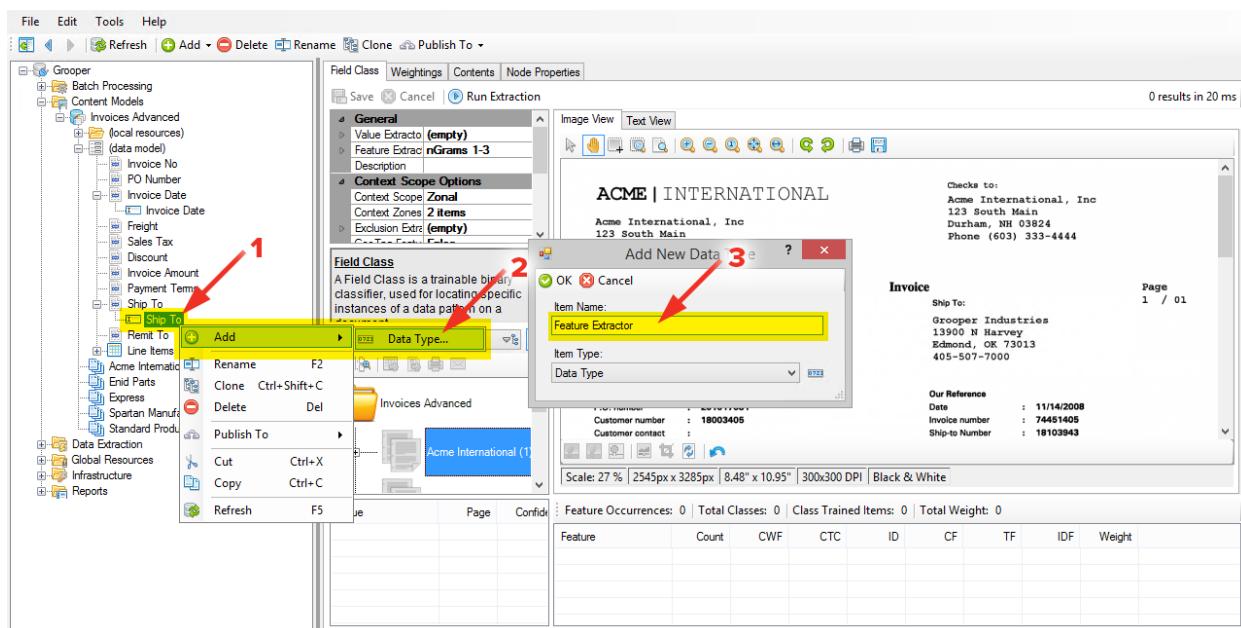
STEP 1 – ADDING A FIELD CLASS TO THE SHIP TO DATA FIELD

Right-click the [Ship To Data Field](#) and [Add > Field Class...](#) Name it [Ship To](#).



STEP 2 – ADDING A DATA TYPE TO THE SHIP TO FIELD CLASS

(1) Right-click the [Ship To Field Class](#) (2) and [Add > Data Type...](#) (3) Name it [Feature Extractor](#).



STEP 3 – ADDING AN INTERNAL PATTERN TO A DATA TYPE

Previously, the patterns added to **Data Types** that have been worked with were via **Data Format** objects. This pattern will be written internally on the **Data Type** by selecting the **Pattern** property in the **Data Extraction** section and clicking the ellipsis button.

General

- Data Extraction** (empty) **...**
- Referenced Extractors (0 selected)
- Input Filter (empty)
- Exclusion Extractor (empty)
- Output**
- Deduplication

Pattern
Defines an internal [Data Pattern](#) which can be used in place of a child data format. (See [Data Pattern](#))

Remarks
This property is useful for simple extractions where only one format needs to be defined.

Batch: Invoices Advanced

Image View **Text View**

ACME | INTERNATIONAL

Acme International, Inc
123 South Main
Durham, NH 03824
Phone (603) 333-4444

Checks to:
Acme International, Inc
123 South Main
Durham, NH 03824
Phone (603) 333-4444

Invoice **Page**
1 / 01

Bill To:
Grooper Industries
13900 N Harvey
Edmond, OK 73013
405-507-7000

Ship To:
Grooper Industries
13900 N Harvey
Edmond, OK 73013
405-507-7000

Scale: 32 % 2545px x 3285px 8.48" x 10.95" 300x300 DPI Black & White

Value	Confidence	Page No	Line No	Index	Length	Format	Pattern

STEP 4 – WRITING THE PATTERN / FORCING A SPECIFIC OUTPUT

- (1) In the **Value Pattern** use the following pattern:

```
ship to|shipped to|ship
```

The vertical pipe characters in **RegEx** determine **OR**, so in this case find **ship to** or **shipped to** or **ship**.

- (2) In the **Output Format** use the following pattern:

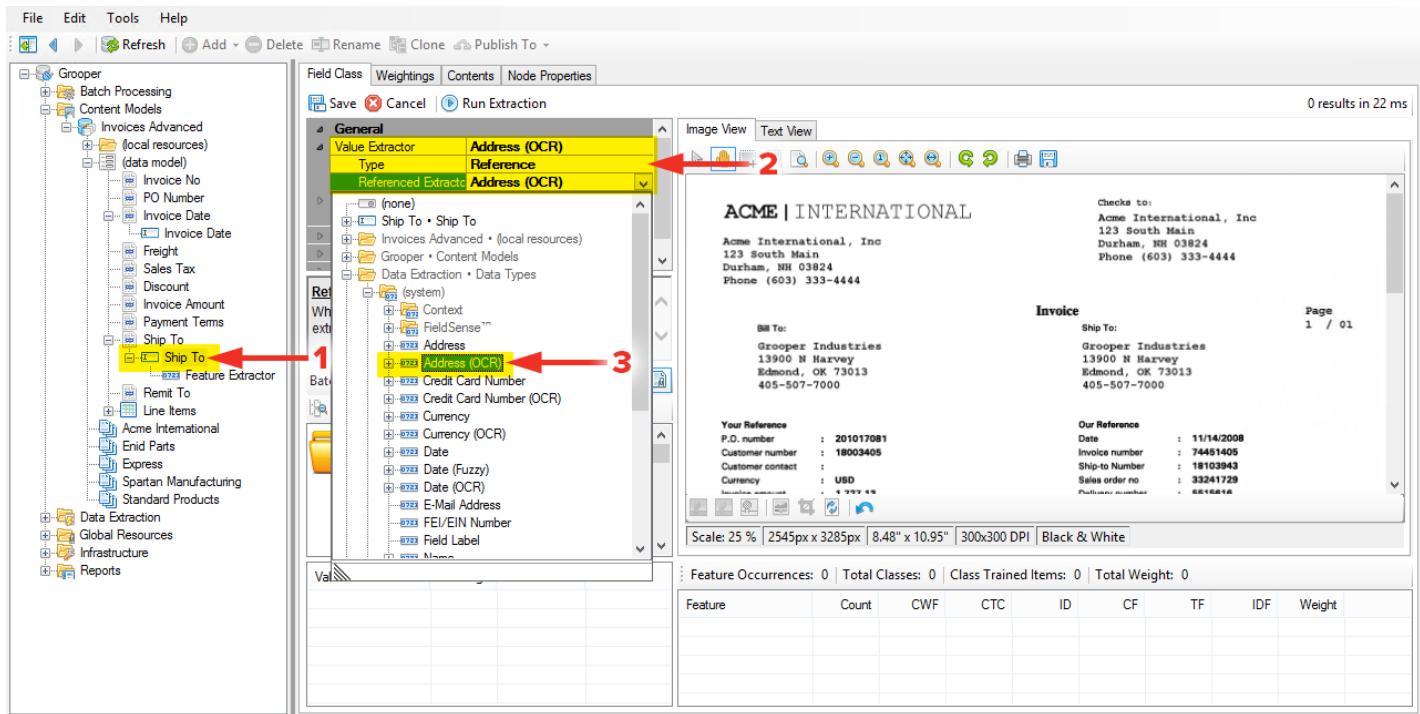
Feature-SHIP To

- (3) This will force all instances of returned results found by the **RegEx** pattern to be returned as the string **Feature-SHIP To**. Click through some documents in the **Batch Viewer** and see the results that are returned. When the **Field Class** features are trained, the usefulness of this technique will become apparent.

The screenshot shows the Grooper ACE software interface. On the left, the **Pattern Editor** window is open, displaying the **Value Pattern** field with the value **1 ship to|shipped to|ship**. A red arrow labeled **1** points to this field. Below it are the **Look Ahead Pattern** and **Look Behind Pattern** fields, both set to **1**. On the right, the **Batch Viewer** displays a document titled **ACME | INTERNATIONAL**. The document contains shipping information for Acme International, Inc. The **Invoice** section shows the recipient as **Grooper Industries** with address details. The **Results** table in the Batch Viewer shows two entries for "Feature-SHIP To". Red arrows labeled **2** point to the **Output Format** field in the Pattern Editor and the first entry in the Results table. Red arrow **3** points to the second entry in the Results table.

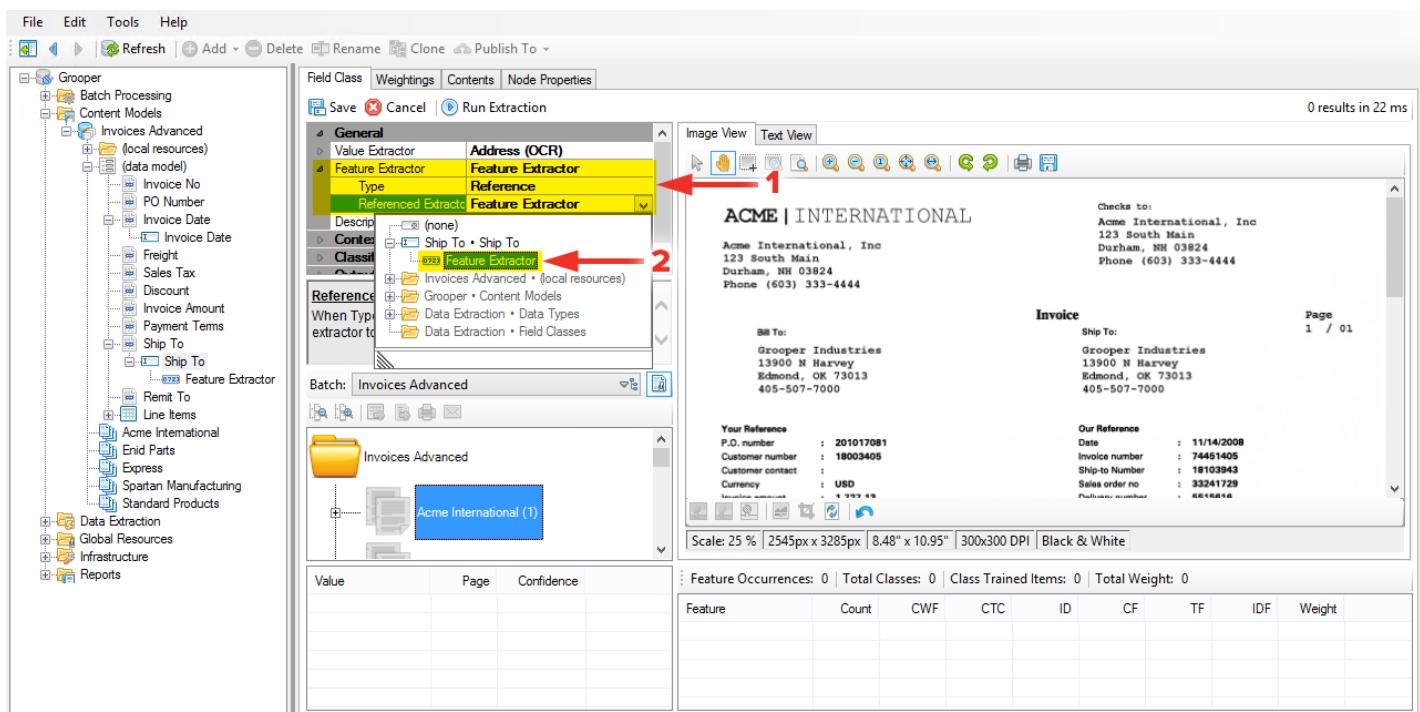
STEP 5 – SETTING THE VALUE EXTRACTOR

(1) Click back on the **Ship To** Field Class. (2) Set the **Value Extractor** type to **Reference** (3) and the **Referenced Extractor** to the Data Extraction • Data Types > (system) > Address (OCR) Data Type.



STEP 6 – SETTING THE FEATURE EXTRACTOR

(1) Set the **Feature Extractor** Type to **Reference** and (2) set the **Referenced Extractor** to the **Feature Extractor** Data Type that is a child object of the **Ship To** Field Class.

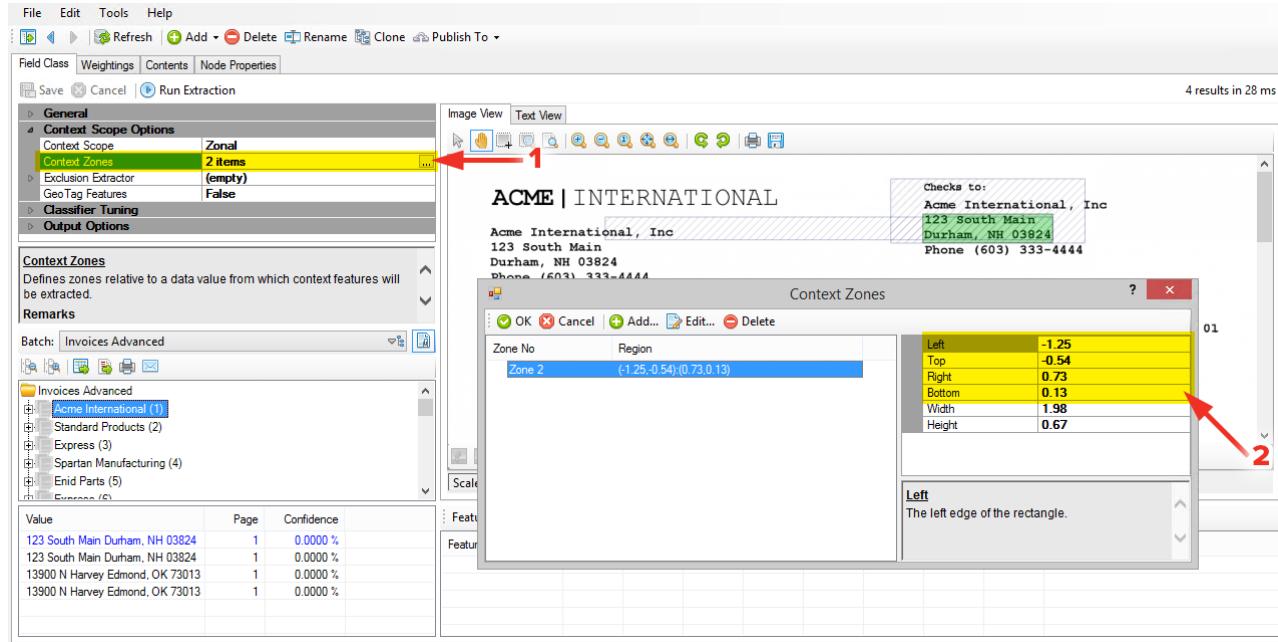


STEP 7 – ADJUSTING THE CONTEXT ZONES

In the **Context Scope Options** section, (1) select the **Context Zones** property and click the ellipsis button to bring up the **Context Zones** properties window. (2) Delete **Zone 1** and set the **Zone 2** dimensions to:

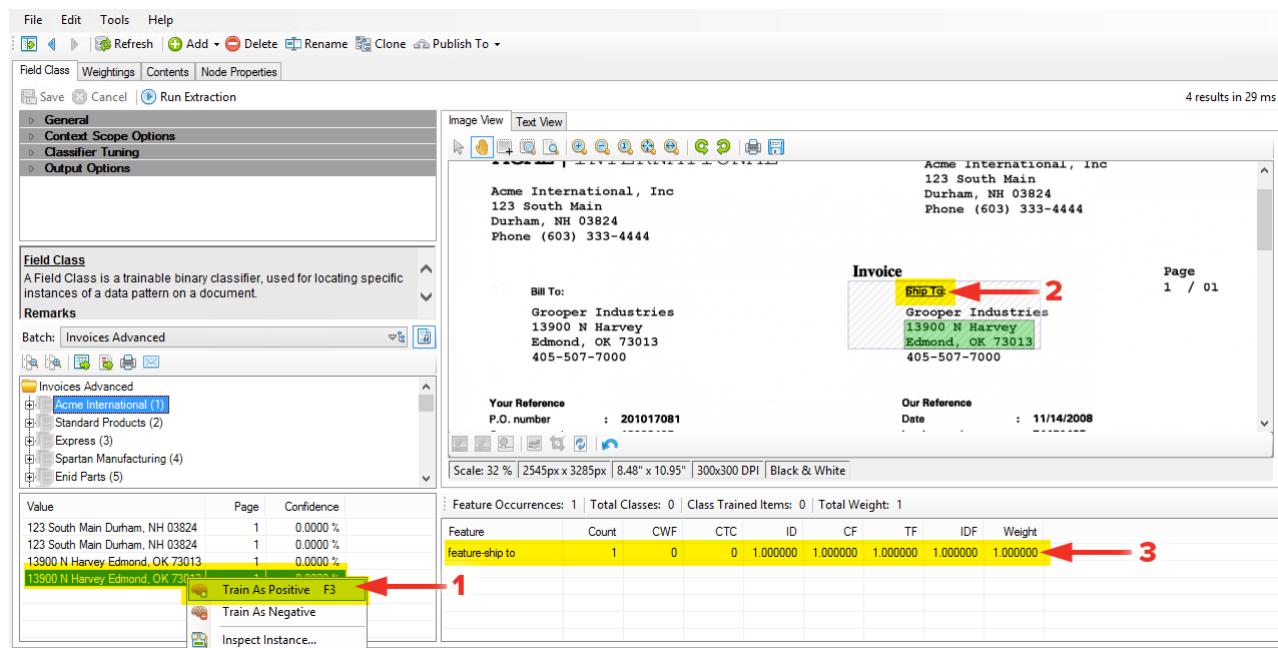
Zone 2 – Left: -1.25 Top: -0.54 Right: 0.73 Bottom: 0.13

Click **OK** to close the **Context Zones** properties window, then **Save** and **Run Extraction** on the **Field Class** against the **Acme International (1)** document.



STEP 8 – TRAINING A FEATURE INSTANCE

Four values should be returned, but only one of them will capture a feature. (1) Select the value that returns a feature, and (2) notice that **Ship To** in the image has the blue box drawn around it, but (3) the **Feature Occurrence** that's returned is **feature-ship to**. Train this instance positively.



STEP 9 – CHECK WEIGHTINGS

Check the **Weightings** tab to verify that only one feature has been trained, and it is **feature-ship to**. No negative instances will be listed because the **Context Zone** around the other values from the **Acme International (1)** did not capture a feature, and as such, did not return negative results during the positive training of the correct value.

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
feature-ship to	1	1	0	1.000000	1.000000	1.000000	0.602060	0.602060

STEP 10 – TESTING OTHER DOCUMENTS

(1) Select the **Standard Products (2)** document from the **Batch Viewer** and (2) notice the top value return the **feature-ship to** and a confidence of 100%.

Value	Page	Confidence
13900 N Harvey Edmond, OK 73013	1	100.0000 %
19658 South Frisk Oklahoma City, ...	1	0.0000 %
13900 N Harvey Edmond, OK 73013	1	0.0000 %
19658 South Frisk Oklahoma City, ...	1	0.0000 %

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
feature-ship to	1	1	0	1.000000	1.000000	1.000000	0.602060	0.602060

Now test either **Express (3)** or **Enid Parts (5)** and notice that the top value result is also returning the feature **feature-ship to**, (3) but the highlighted feature in the **Page Viewer** is **Shipped to** and **Ship** (respectively.)

If you recall, the **Data Type** that was built to extract features is looking for: **ship**, **shipped to**, and **ship to**, but results are always returned as: **feature-ship to**. This feature, as a result, only needed to be trained once, and the **Field Class** will operate at 100% confidence. If the **Data Type** were built to find those words, but only return what was found, each instance of **ship**, **shipped to**, or **ship to** would have to be trained separately.

STEP 11 – SETTING MINIMUM CONFIDENCE

Because there is only one feature trained, and it is being returned at **100%** confidence, the **Minimum Confidence** can be set to anything above **0%** and all unwanted results will be trimmed.

STEP 12 – SETTING PROPERTIES FOR THE SHIP TO DATA FIELD

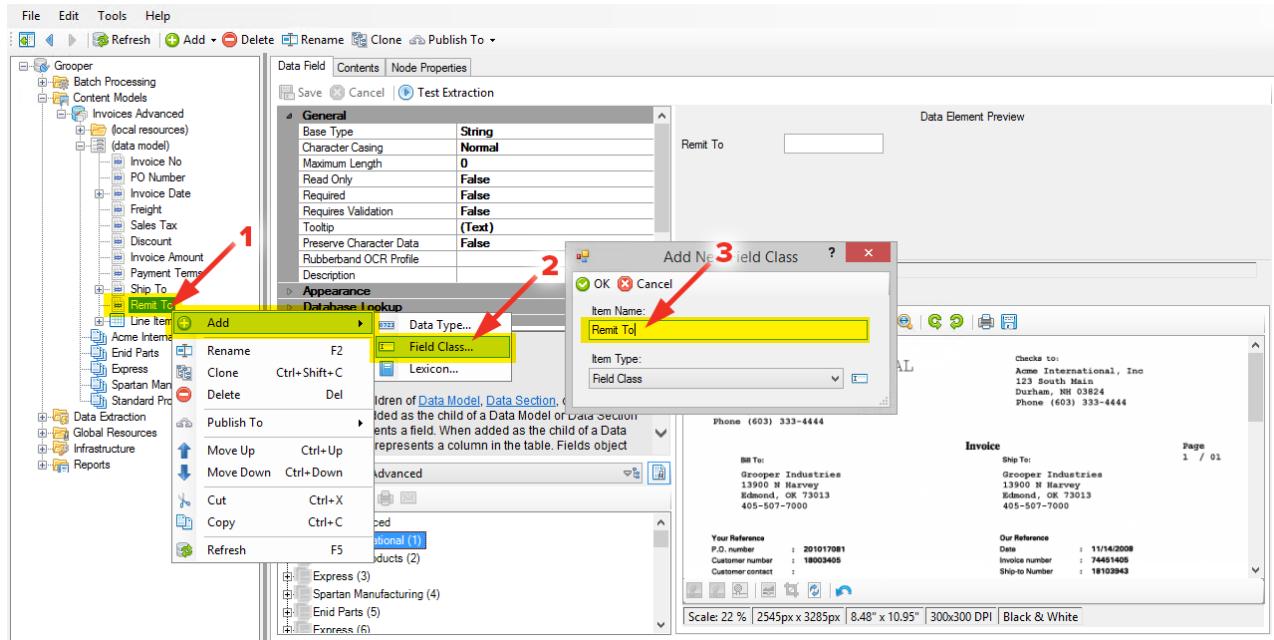
(1) Select the **Ship To Data Field** and **(2)** set its **Required** property to **True**. **(3)** Set the **Default Extractor Type** to **Reference** and set the **Referenced Extractor** to the **Ship To Field Class**.

THE NEXT FIELD CLASS – BUILDING A FEATURE EXTRACTOR WITH FUZZYREGEX

The **Field Class** for the **Ship To Data Field** used a simple pattern in a **Data Type** to find specific sets of words, but return them as a single output. That same idea will be carried on for the next **Field Class**, but the **Feature Extractor** will now compensate for **OCR** errors via a method in **Grooper** called **FuzzyRegEx**.

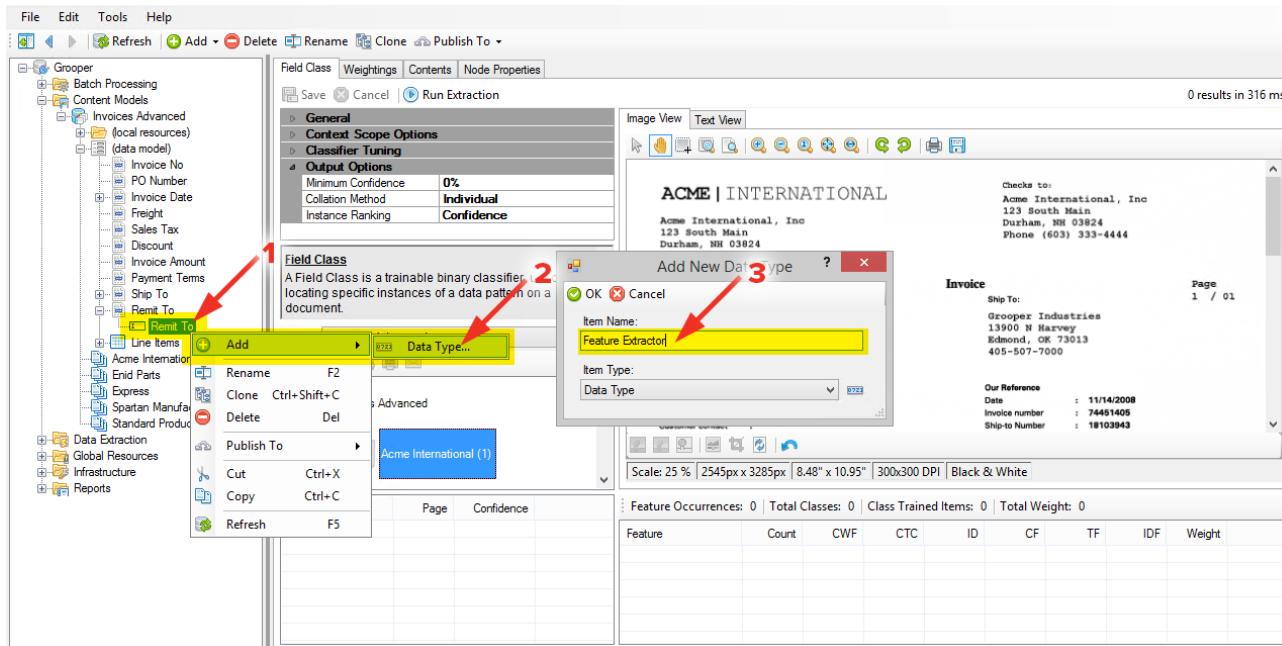
STEP 1 – ADDING A FIELD CLASS TO THE REMIT TO DATA FIELD

Right-click the **Remit To Data Field** and **Add > Field Class...** Name it **Remit To**.



STEP 2 – ADDING A DATA TYPE TO THE SHIP TO FIELD CLASS

Right-click the **Remit To Field Class** and **Add > Data Type...** Name it **Feature Extractor**.



STEP 3 – ADDING AN INTERNAL PATTERN TO THE FEATURE EXTRACTOR DATA TYPE

Like the previous [Feature Extractor Data Type](#) that was built, this [Feature Extractor Data Type](#) will also use an internal pattern. Click the [Pattern](#) property within the [Data Extraction](#) section and then click the ellipsis button to bring up the pattern editor.

The screenshot shows the Grooper ACE interface with the 'Data Type' editor open for the 'Invoices Advanced' data type. The 'General' tab is selected, and the 'Data Extraction' section is expanded. The 'Pattern' field is highlighted with a yellow box, and a red arrow points to the ellipsis button in the toolbar above the preview area. The preview area displays a scanned document titled 'ACME | INTERNATIONAL' with fields such as 'Bill To:' and 'Ship To:'. The bottom right corner of the preview area shows 'Page 1 / 01'.

STEP 4 – WRITING THE PATTERN / FORCING A SPECIFIC OUTPUT

In the **Value Pattern** area using the following pattern:

```
remit ?to|mail ?payment ?to|checks ?to
```

This pattern is using vertical pipes again, and also making the spaces optional by using the 0-1 quantifier **?**, therefore looking for **remit to**, **mail payment to**, and **checks to**, with optional spaces.

In the **Output Format** area using the following pattern:

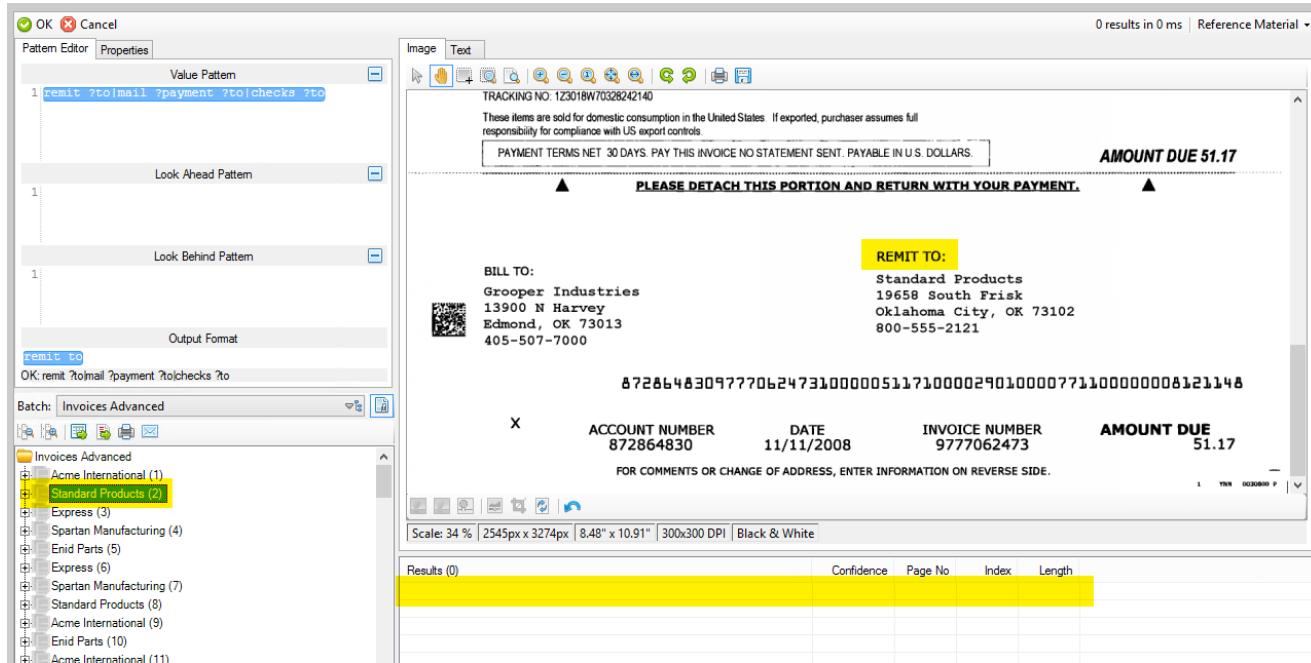
```
remit to
```

This will force all returned values to be **remit to**, instead of the values returned by the **RegEx** pattern. So notice on the **Acme International (1)** document that **Checks to** is found, but returned as **remit to**.

The screenshot shows the Grooper ACE Pattern Editor interface. On the left, the **Value Pattern** field contains the regular expression `remit ?to|mail ?payment ?to|checks ?to`. Below it, the **Output Format** field contains the pattern `remit to`. The **Batch:** dropdown is set to "Invoices Advanced". The file tree on the left shows a folder named "Acme International (1)" which is selected. The main pane displays a PDF document from "ACME | INTERNATIONAL". The document includes shipping information for "Grooper Industries" and an invoice section. In the PDF, the word "Checks to:" is highlighted in yellow, and the value "Acme International, Inc" is also highlighted, demonstrating that the output format was applied correctly. The bottom right corner of the PDF shows the page number "1 / 01".

STEP 5 – DISCOVERING AN OCR ERROR

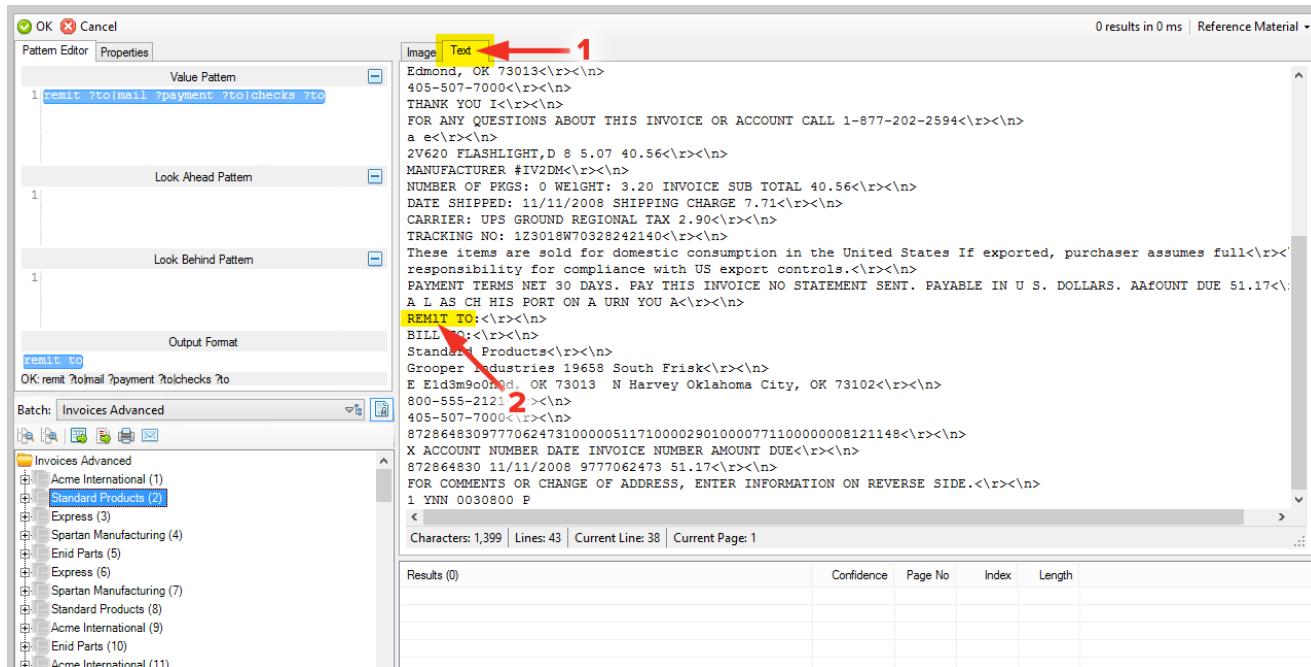
Click on the **Standard Products (2)** document in the **Batch Viewer** and notice our pattern is not returning any results. Scroll the image down in the **Page Viewer** and notice that **REMIT TO** is on this page, so the **RegEx** pattern used should find it, right?



STEP 6 – DIAGNOSING THE PROBLEM

The first thing to understand about **OCR** is that it is imperfect, and it will never return results with 100% accuracy.

(1) Click on the **Text** tab and scroll down to (2) see that the capital **I** in **REMIT TO** was mis-**OCR**ed as a lower case **i**. While complicating the RegEx pattern to accommodate for this is possible, there's a more elegant solution available in **Grooper**.



STEP 7 – ENABLING FUZZYREGEX

(1) Click on the **Properties** tab and in the **General** section, (2) change the **Extraction Mode** from **RegEx** to **FuzzyRegEx**. Notice when doing this a new section of options become available: **Fuzzy Matching Options**.

The screenshot shows the Grooper ACE Pattern Editor interface. On the left, there's a sidebar with a tree view showing a folder named 'Invoices Advanced' containing three sub-folders: 'Acme International (1)', 'Standard Products (2)', and 'Express (3)'. The main area has tabs for 'Image', 'Text', and 'Fuzzy Extraction Visualizer'. The 'Text' tab is active, displaying a large block of text representing an invoice. At the top of the main area, there are buttons for 'OK', 'Cancel', and 'Properties'. The 'Properties' button is highlighted with a yellow box and a red arrow labeled '1'. In the 'General' section of the properties, the 'Extraction Mode' is set to 'FuzzyRegEx', which is also highlighted with a yellow box and a red arrow labeled '2'. Below this, under 'Fuzzy Matching Options', the 'Minimum Similarity' is set to '90%', also highlighted with a yellow box and a red arrow labeled '3'. The text area contains various invoice details like account number, date, and items shipped.

The first property to understand with **FuzzyRegEx** is (3) the **Minimum Similarity**. This is a threshold that is set to disallow translations of our words that fall below this set amount. At **90%** we still won't get a match on this document.

STEP 8 – ADJUSTING THE MINIMUM SIMILARITY TO FIND A MATCH

(1) Change the Minimum Similarity to **85%** and observe what happens. (2) The mis-OCR'd **REMIT TO** will be found and translated to the specific string we told our **RegEx** pattern to find, which is **remit to** (keep in mind the **Case Sensitive** property is **False** by default.) (3) It's also worth noting that our **Output Format** is forcing the returned value to be **remit to**, which means any result will be returned as the string defined by the **Output Format**.

Notice also some of the information being displayed in the **Results List View**, mainly the **Confidence** of **88%**. Dropping the **Minimum Similarity** from the **90%** threshold to this now **85%** allowed this match and translation to occur.

So why is the confidence of this returned result **88%**? Consider that the length of the string our **RegEx** pattern is looking for happens to be 8 characters long. One of the characters in this string was mis-OCR'd as a lower case **l** but returned from the **FuzzyRegEx** as an **i**. If we consider this one missed character as a percentage of the 8 being sought, it is **12%**. Therefore, logically, having translated that one character, the percentage similarity drops from **100%** to **88%**.

The screenshot shows the Grooper ACE software interface. On the left, the 'Pattern Editor' pane is open, showing configuration for 'FuzzyRegEx' mode. The 'Fuzzy Matching Options' section has 'Minimum Similarity' set to **85%**, highlighted with a yellow box and a red arrow labeled '1'. In the main 'Text' pane, the invoice content is displayed, including the 'REMIT TO:' field. The 'Results List View' pane at the bottom shows a single result: 'remit to' with a confidence of **88 %**, index **1**, and length **8**. A red arrow labeled '2' points to the 'remit to' entry in the results table. Another red arrow labeled '3' points to the confidence value in the results table.

Results (1)	Confidence	Age No	Index	Length
remit to	88 %	1	998	8

STEP 9 – FURTHER UNDERSTANDING OF FUZZYREGEX

While 85% isn't necessarily a low threshold to set, we want the **Minimum Similarity** to be as high as possible. The reasoning behind this is to avoid false positives. (1) Lower the **Minimum Similarity** to 50% and (2) notice all the garbage that starts to be considered a match based on that low similarity (set it back to 85% when you're done experimenting.)

Pattern Editor Properties

Fuzzy Matching Options

- Minimum Similarity: **50%** (highlighted with a yellow background and a red arrow)
- Fuzzy Match Weightings: (empty)

Results Table:

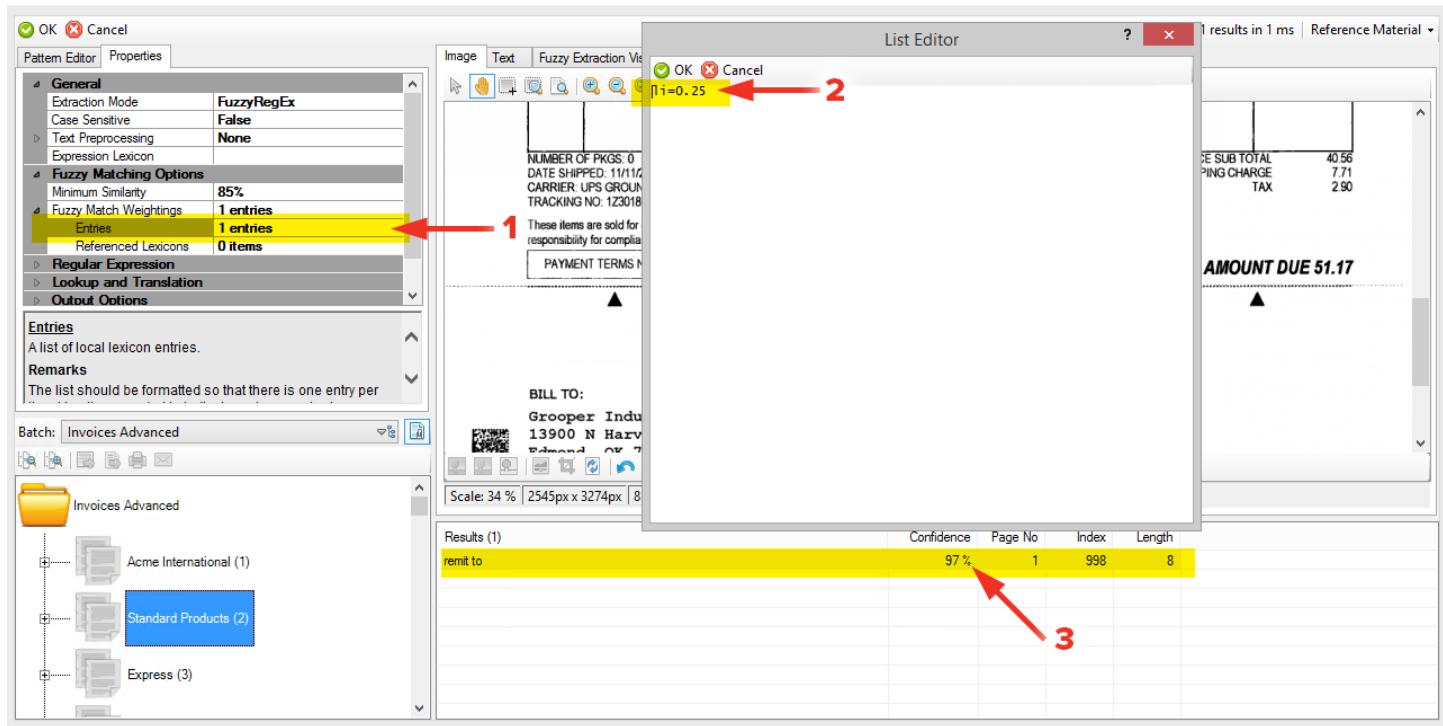
Text	Confidence	Page No	Index	Length
remit to	50 %	1	203	8
remit to	57 %	1	644	5
remit to	50 %	1	709	5
remit to	50 %	1	842	5
remit to	64 %	1	852	14
remit to	56 %	1	970	9
remit to	50 %	1	979	4
remit to	88 %	1	998	8
remit to	57 %	1	1358	6

STEP 10 – FUZZY MATCH WEIGHTINGS

To keep confidence in desired results high, and as a result, keep the [Minimum Similarity](#) as high as possible to avoid false positives, **Grooper** has an excellent solution called [Fuzzy Match Weightings](#). **(1)** Expand [Fuzzy Match Weightings](#) to reveal its properties: [Entries](#) and [Referenced Lexicons](#). We'll focus on [Entries](#) for now. Click on the [Entries](#) property then click the ellipsis button to open a [List Editor](#) window. **(2)** Type the following in the [List Editor](#), then click [OK](#):

1*i=0.25*

This [Entries](#) list uses a very specific syntax. The first character in this string is the incorrect character ([keep in mind the syntax of this entries list is case sensitive.](#)) The second is what you want the incorrect character to be translated to. Finally, the equals sign followed by the decimal number is saying what percentage of a whole cost this specific translation will be. So, in the [remit to](#) pattern the cost of the translation for this one character has been [12%](#) (putting the confidence at [88%](#)), but we've now defined the translation of the [1](#) to the [i](#) to be [25%](#) of [12%](#), so now the cost will now only be [3%](#) (therefore, **(3)** the confidence of this result will now be [97%.](#))



STEP 11 – SETTING THE VALUE EXTRACTOR

- (1) Click back on the **Remit To Field Class**. (2) Set the **Value Extractor** type to **Reference** (3) and the **Referenced Extractor** to the **Data Extraction • Data Types > Address (OCR)** Data Type.

The screenshot shows the Grooper interface with the 'Field Class' tab selected. In the left navigation pane, 'Invoices Advanced' is expanded, and 'Remit To' is selected. The main panel displays the 'General' settings for the 'Value Extractor' and 'Feature Extractor'. The 'Value Extractor' is set to 'Address (OCR)' (1), and the 'Referenced Extractor' is also set to 'Address (OCR)' (2). A red arrow points from step 3 to the 'Address (OCR)' option in the 'Referenced Extractor' dropdown (3). The right panel shows a sample invoice with address extraction results, including 'ACME | INTERNATIONAL' and 'Invoice' details. The bottom right shows a feature occurrence table.

STEP 12 – SETTING THE FEATURE EXTRACTOR

- (1) Set the **Feature Extractor Type** to **Reference** and (2) set the **Referenced Extractor** to the **Feature Extractor Data Type** that is a child object of the **Remit To Field Class**.

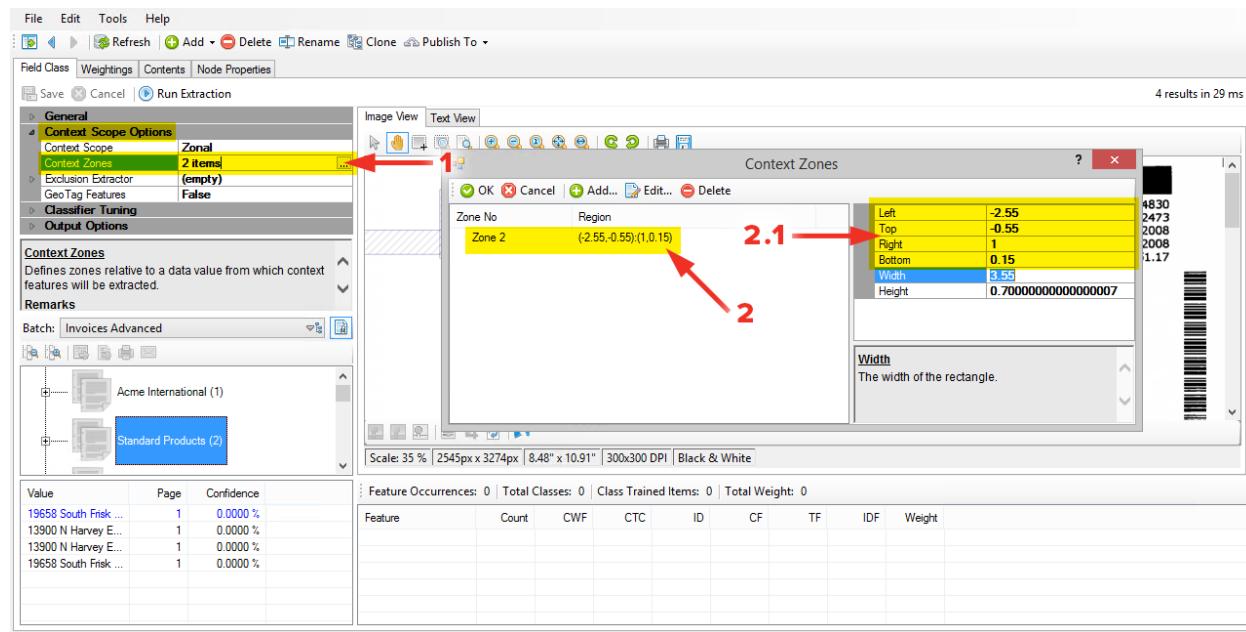
The screenshot shows the Grooper interface with the 'Field Class' tab selected. In the left navigation pane, 'Invoices Advanced' is expanded, and 'Remit To' is selected. The main panel displays the 'General' settings for the 'Value Extractor' and 'Feature Extractor'. The 'Value Extractor' is set to 'Address (OCR)', and the 'Feature Extractor' is set to 'Feature Extractor' (1). A red arrow points from step 2 to the 'Feature Extractor' option in the 'Referenced Extractor' dropdown (2). The right panel shows a sample invoice with address extraction results, including 'ACME | INTERNATIONAL' and 'Invoice' details. The bottom right shows a feature occurrence table.

STEP 13 – ADJUSTING THE CONTEXT ZONES

In the **Context Scope Options** section, (1) select the **Context Zones** property and click the ellipsis button to bring up the **Context Zones** window. Delete **Zone 1** and (2) set the **Zone 2** dimensions to the following:

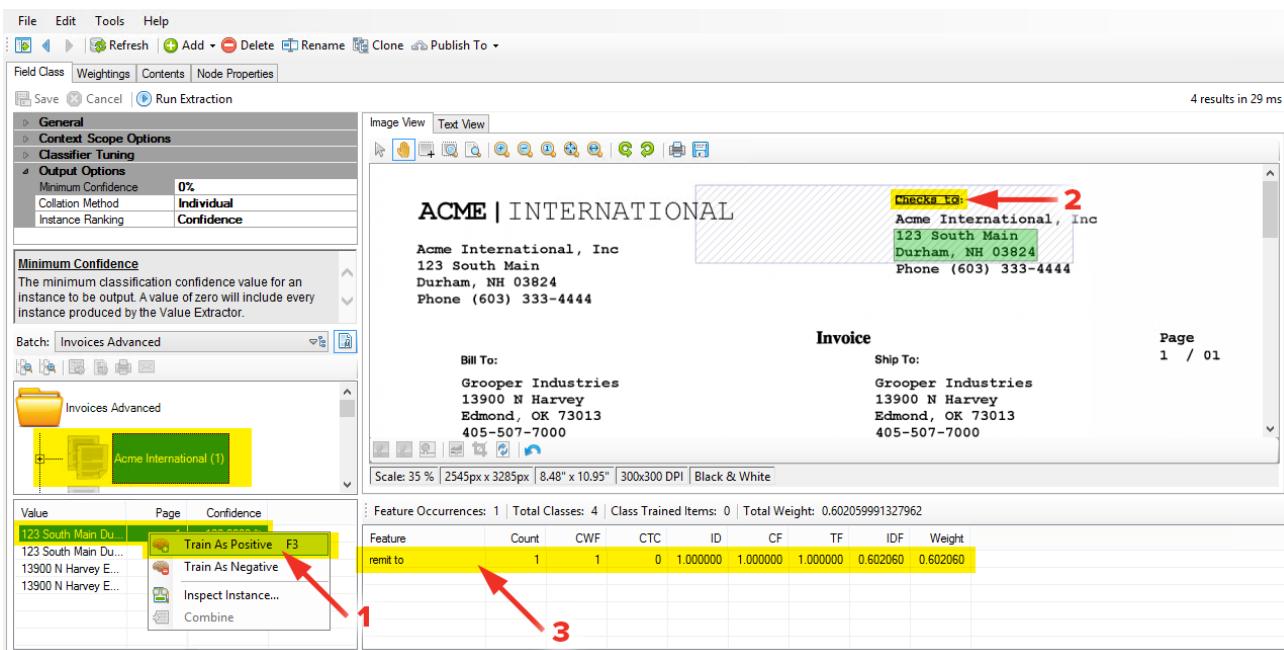
Zone 2 – Left: -2.55 Top: -0.55 Right: 1 Bottom: 0.15

Click **OK** to close the **Context Zones** properties window, then **Save** and **Run Extraction** on the **Field Class** against the **Acme International (1)** document.



STEP 14 – TRAINING A FEATURE INSTANCE

Four values should be returned, but only one of them will capture a feature. (1) Select the value that returns a feature, and (2) notice that **Checks To** in the image has the blue box drawn around it, (3) but the **Feature Occurrence** that's returned is **remit to**. Train this instance positively.



STEP 15 – SETTING MINIMUM CONFIDENCE

Because there is only one feature trained, and (1) it is being returned at 100% confidence, (2) the Minimum Confidence can be set to anything above 0% and all unwanted results will be trimmed.

The screenshot shows the Grooper ACE interface. On the left, the 'Field Class' configuration pane is open, showing the 'Output Options' section with 'Minimum Confidence' set to 1%. A red arrow labeled '2' points to this setting. Below this, the 'Field Class' description states: 'A Field Class is a trainable binary classifier, used for locating specific instances of a data pattern on a document.' The 'Batch' dropdown is set to 'Invoices Advanced'. The main workspace displays an 'Image View' of an invoice from 'ACME | INTERNATIONAL' to 'Grooper Industries'. The 'Invoice' and 'Ship To' sections are visible. At the bottom, a table shows 'Feature Occurrences' and 'Weights' for the trained item 'remit to'. A red arrow labeled '1' points to the first row of this table.

STEP 16 – SETTING PROPERTIES FOR THE REMIT TO DATA FIELD

(1) Select the Remit To Data Field and (2) set its Required property to True. (3) Set the Default Extractor Type to Reference and set the Referenced Extractor to the Remit To Field Class.

The screenshot shows the Grooper ACE interface. On the left, the 'Data Field' configuration pane is open, showing the 'Required' property set to 'True' (highlighted by a red arrow labeled '2'). The 'Default Extractor' section shows 'Type' set to 'Reference' (highlighted by a red arrow labeled '3.1') and 'Referenced Extractor' set to 'Remit To' (highlighted by a red arrow labeled '3.2'). The 'Batch' dropdown is set to 'Invoices Advanced'. The main workspace displays an 'Instance View' of an invoice from 'ACME | INTERNATIONAL' to 'Grooper Industries'. The 'Bill To' and 'Ship To' sections are visible. At the bottom, a table shows 'Your Reference' and 'Our Reference' details.

THE REMAINING FIELD CLASSES – LEXICONS, CHEAT CODES, AND ARRAYS

There are a few fields left whose extraction is best suited by a [Field Class](#). Having built out a few to gain an understanding of how they function, it's now a good time to introduce a new concept of creating a [Feature Extractor](#) that can work for many, if not all, future [Field Classes](#).

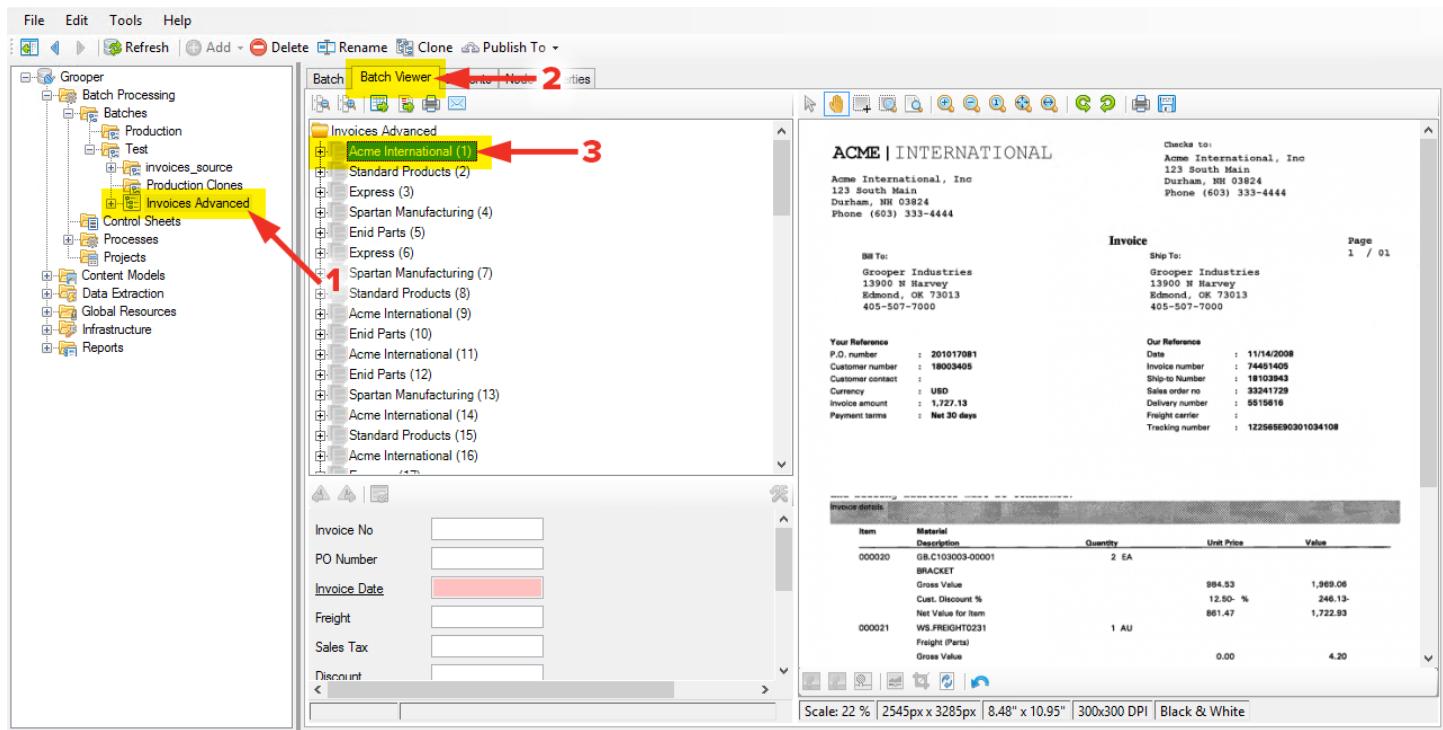
A [Data Type](#) will be built that will use several new concepts including [OrderedArrays](#), a [Lexicon](#) with translations, and a very simple but highly effective [RegEx](#) pattern we lovingly referred to as the [Cheat Code](#).

BUILDING A FEATURE LIST LEXICON

If you take a moment to look through the 5 main [DocTypes](#) and analyze the information on them, you can start to understand that basically every bit of relevant information on these documents has a label near it that defines what that bit of information is. If one were to take a moment and make a list of these labels he/she would have a master source of all the relevant [Features](#) that a [Field Class](#) could be trained against.

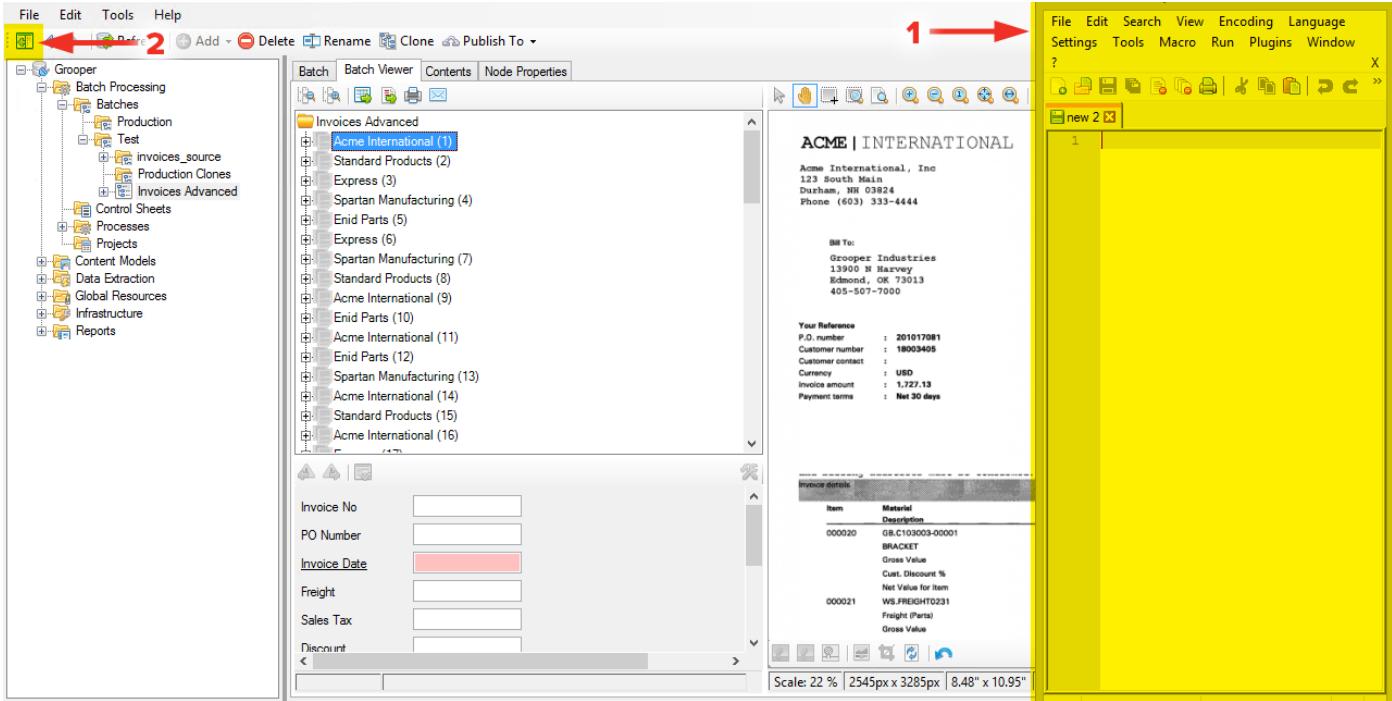
STEP 1 – VIEWING THE DOCUMENTS IN GROOPER

- (1) Navigate to [Grooper > Batch Processing > Batches > Test](#) and select the [Invoices Advanced](#) batch.
- (2) Click on the [Batch Viewer](#) tab and (3) select a document from the [Batch Viewer](#).



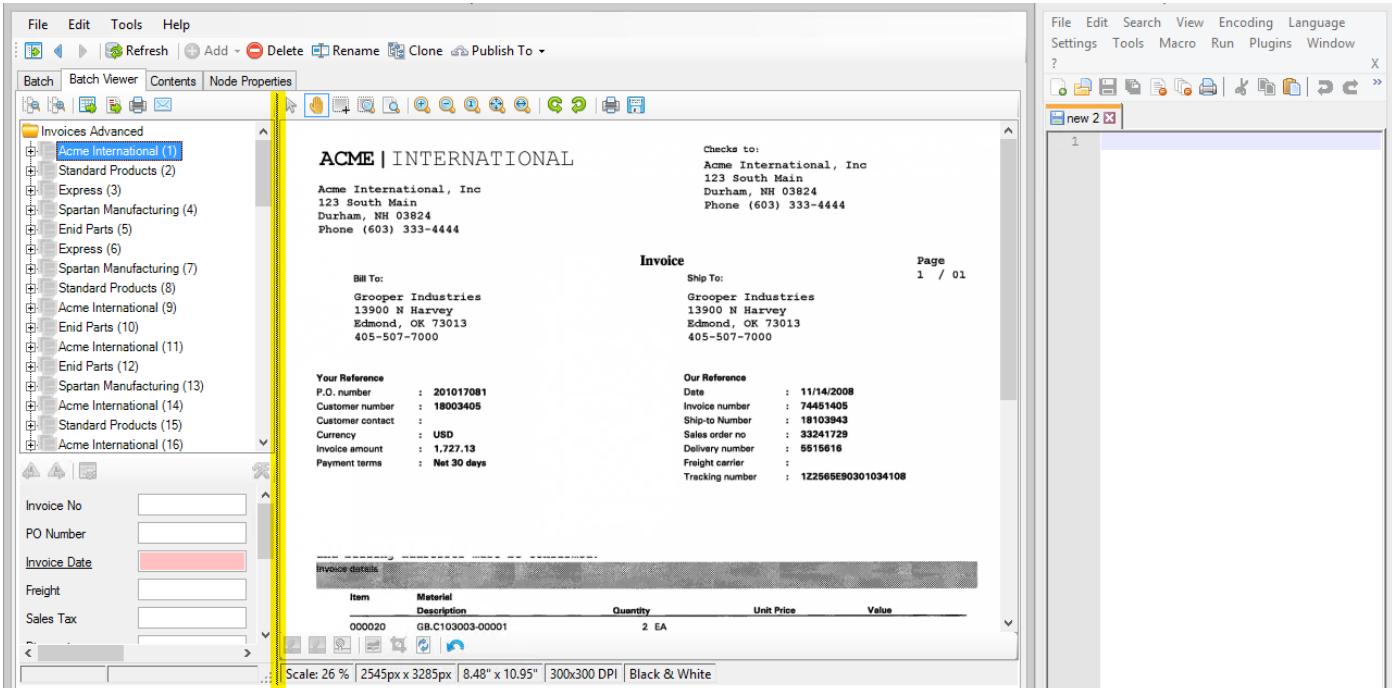
STEP 2 – OPEN A NOTEPAD APPLICATION AND HIDE TREE PANEL

In Windows, (1) open a **Notepad** program and anchor it to the side of the screen. You'll be writing multiple lines of short strings of information, so feel free to make it narrow, as you'll be sharing screen real estate with **Grooper**. In **Grooper**, (2) click the **Hide Tree Panel** button. Clicking the same button will bring it back as well.



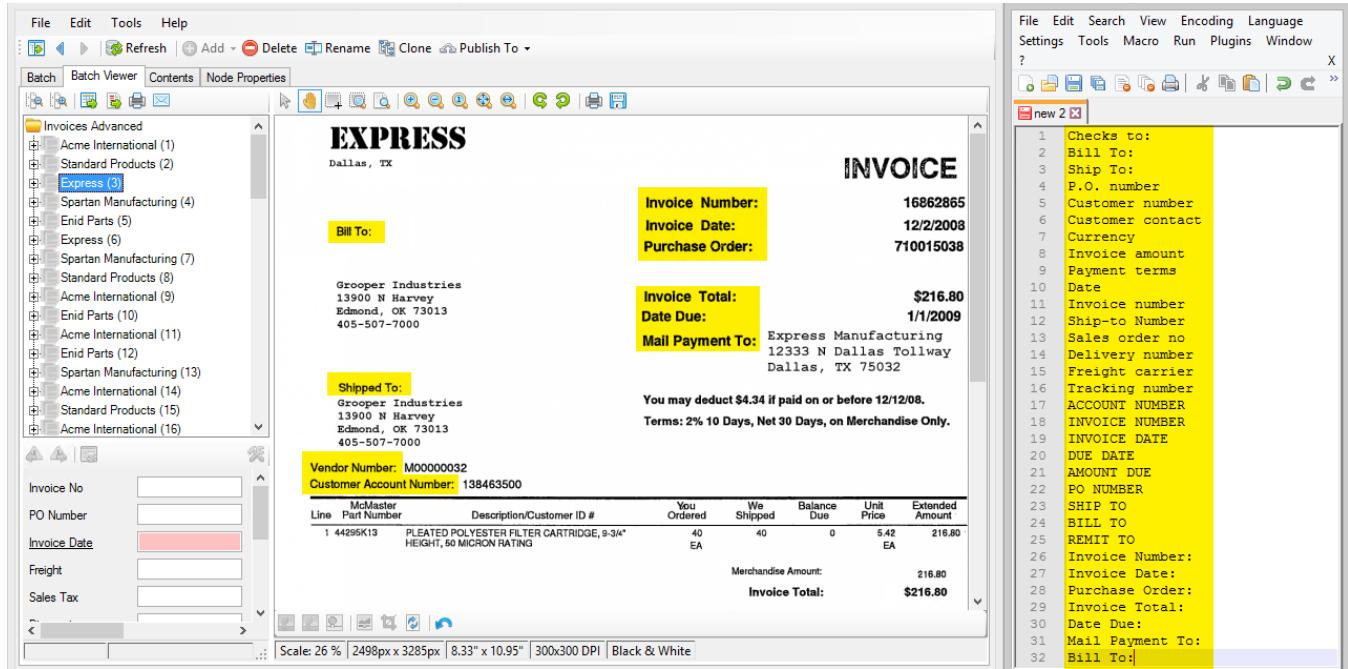
STEP 3 – ADJUSTING THE UI

Anchor **Grooper** to the other side of the screen, then move the panels around within the **Grooper** UI so that the **Page Viewer** allows easy reading of the documents.



STEP 4 – CREATING A LIST OF LABELS

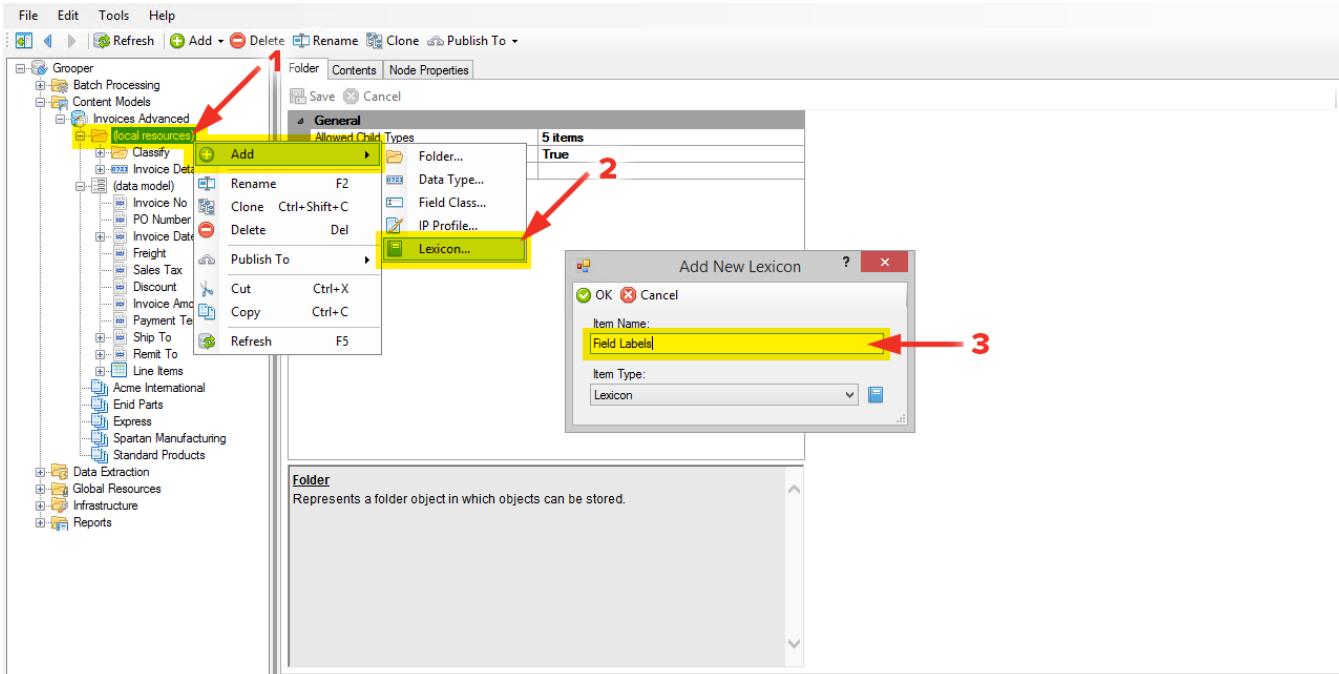
Look over the documents and begin to type into the **Notepad** application all the labels that define the fields of information within the documents. Don't worry about duplicate entries or what order they're in. Be sure to spell exactly as they appear on the documents, including spacing and special characters. Considering that **Grooper** can be configured for case sensitivity, it's best practice to also type the correct casing.



There are only 5 main **DocTypes** to go through, and when you get to the **Enid Parts (5)**, don't bother listing the labels that are stacked on top of one another like **Order Number** and **Purchase Order Number** and so forth. While you read them as a single unit, they're on separate lines and aren't returned together. This will be clear later when we capture them with arrays.

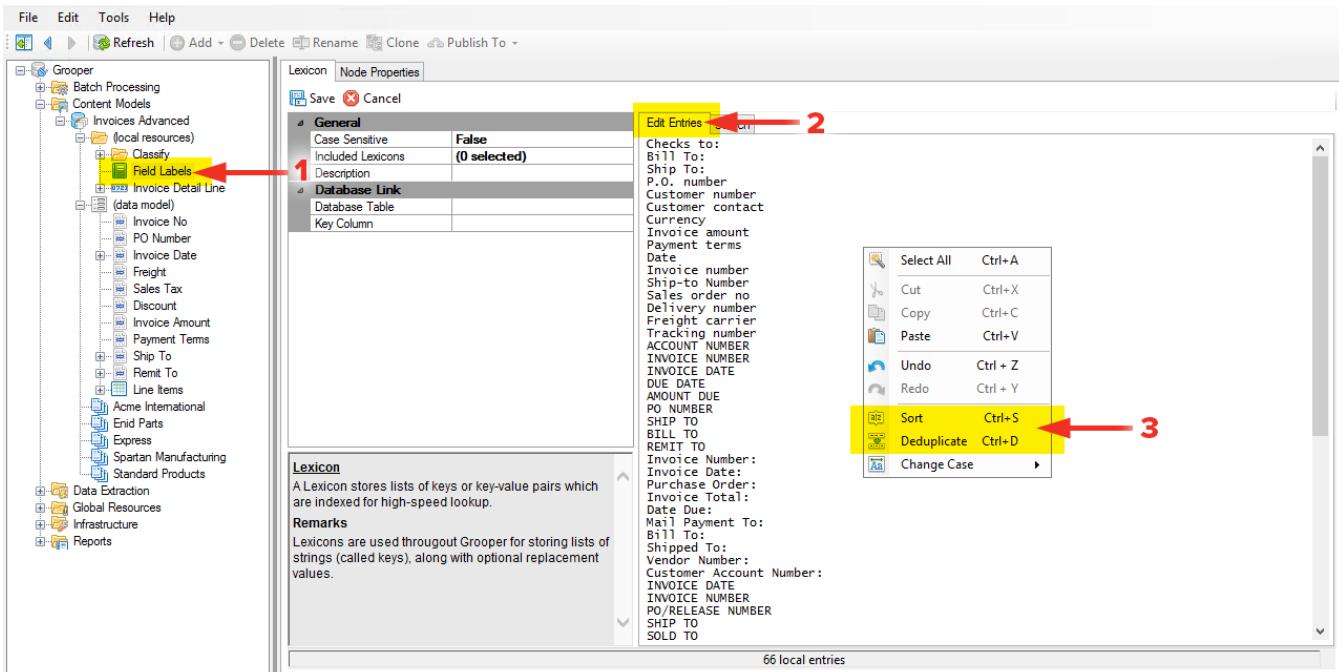
STEP 5 – CREATING A GROOPER LEXICON

When you've gotten through all 5 DocTypes, copy the list to the clipboard. Back in **Grooper**, (1) navigate to **Grooper > Content Models > Invoices Advanced** and (2) right-click on the **(local resources)** folder node and Add > Lexicon... (3) Name it **Field Labels**.



STEP 6 – CREATE LEXICON ENTRIES AND SORT

(1) With the **Field Labels** Lexicon selected, (2) paste the contents of the clipboard into the **Edit Entries** area. This should be all the things you typed in the **Notepad** program. With entries in the list, (3) right-click and use the object commands for **Deduplicate** as well as **Sort**. This will eliminate duplicate entries and alphabetize the list.



STEP 7 – ADDING TRANSLATIONS

Notice that there are entries for labels from the different doc types that are called different things. For example **invoice number** is listed as **Invoice number**, **INVOICE NUMBER**, **Invoice Number**, and **INVOICE #**. Lexicons can allow multiple entries to equate to the same output, similar in a fashion to using an **Output Format** in a **Data Type** or **Data Format RegEx** pattern. For the multiple entries of **invoice number**, put an equals sign after them and type how you'd like those values translated. For example:

INVOICE #=Invoice Number

The screenshot shows the Grooper interface with the 'Lexicon' node selected in the left navigation pane. The main window displays the 'Node Properties' dialog for the 'Lexicon' node. In the 'General' tab, the 'Case Sensitive' field is set to 'False'. The 'Included Lexicons' section shows '(0 selected)'. The 'Database Link' section shows 'Database Table' and 'Key Column' both set to empty. The 'Edit Entries' tab is selected, showing a list of local entries. One entry, 'INVOICE #=Invoice Number', is highlighted with a yellow background and has a red arrow pointing to it from the bottom right.

Entry
INVOICE #=Invoice Number
ACCOUNT NUMBER
ACCT NUMBER
AMOUNT DUE
BILL TO
BILL TO:
CASH DISCOUNT
Checks to:
COMMENTS:
Currency
Customer Account Number:
Customer contact
Customer number
Date
Date Due:
Delivery number
DU DATE
ENT BY:
F.O.B.
Fax:
FAX:
Freight
Freight carrier
INVOICE DATE
INVOICE DATE:
Invoice amount
INVOICE DATE
Invoice Date:
Invoice number=Invoice Number
INVOICE NUMBER=Invoice Number
Invoice Number=Invoice Number
Invoice Total:
Mail Payment To:
MDSE. TOTAL
OCN
ORDER DATE
ORDER DUE DATE
OTHER CHARGES
P.O. number
PAGE
Payment terms

61 local entries

STEP 8 – GIVING YOU THE LIST

While it was important to understand the process of typing out the labels and so forth, the list has also been provided below. This list includes all translations that will be leveraged by the [Data Model](#) moving forward. Providing the list will also make sure what you have is in sync with what is expected from the remainder of this document. To save space on this document, I've put the list into a few columns of a table. Just copy the contents from each column and paste them in succession in your [Lexicon](#). Overwrite what you already have in the [Lexicon](#) as well.

General	
Case Sensitive	False
Included Lexicons	(0 selected)
Description	
Database Link	
Database Table	
Key Column	

Lexicon
A Lexicon stores lists of keys or key-value pairs which are indexed for high-speed lookup.
Remarks
Lexicons are used throughout Grooper for storing lists of strings (called keys), along with optional replacement values.

Edit Entries **Search**

- Account Number
- ACCT Number:=Account Number
- Amount Due:=Invoice Amount
- Balance Due:=Invoice Amount
- Bill To
- Bill To:=Bill To
- Cash Discount
- Checks to:=Remit To
- Comments:
- Currency
- Customer Account Number:=Account Number
- Customer Contact
- Customer Number:=Account Number
- Date
- Date Due:=Due Date
- Delivery Number
- Due Date
- Ent By:
- F.O.B.
- Fax:
- Freight
- Freight Carrier
- Invoice #:=Invoice Number
- Invoice Amount
- Invoice Date
- Invoice Date:=Invoice Date
- Invoice Number
- Invoice Number:=Invoice Number
- Invoice Total:=Invoice Amount
- Mail Payment To:=Remit To
- MDSE, Total=Merchandise Total
- Merchandise Amount:=Merchandise Total
- OCN
- Order Date
- Order Due Date
- Other Charges
- P.O. Number=PO Number
- Page
- Payment Terms
- PH:=Phone
- Phone:=Phone
- PO Number
- PO NUMBER:=PO Number
- PO/Release Number=PO Number

65 local entries

Account Number
 ACCT Number:=Account Number
 Amount Due:=Invoice Amount
 Balance Due:=Invoice Amount
 Bill To
 Bill To:=Bill To
 Cash Discount
 Checks to:=Remit To
 Comments:
 Currency
 Customer Account Number:=Account Number
 Customer Contact
 Customer Number:=Account Number
 Date
 Date Due:=Due Date
 Delivery Number
 Due Date
 Ent By:
 F.O.B.
 Fax:
 Freight
 Freight Carrier

Invoice #:=Invoice Number
 Invoice Amount
 Invoice Date
 Invoice Date:=Invoice Date
 Invoice Number
 Invoice Number:=Invoice Number
 Invoice Total:=Invoice Amount
 Mail Payment To:=Remit To
 MDSE, Total=Merchandise Total
 Merchandise Amount:=Merchandise Total
 Total
 OCN
 Order Date
 Order Due Date
 Other Charges
 P.O. Number=PO Number
 Page
 Payment Terms
 PH:=Phone
 Phone:=Phone
 PO Number
 PO NUMBER:=PO Number
 PO/Release Number=PO Number

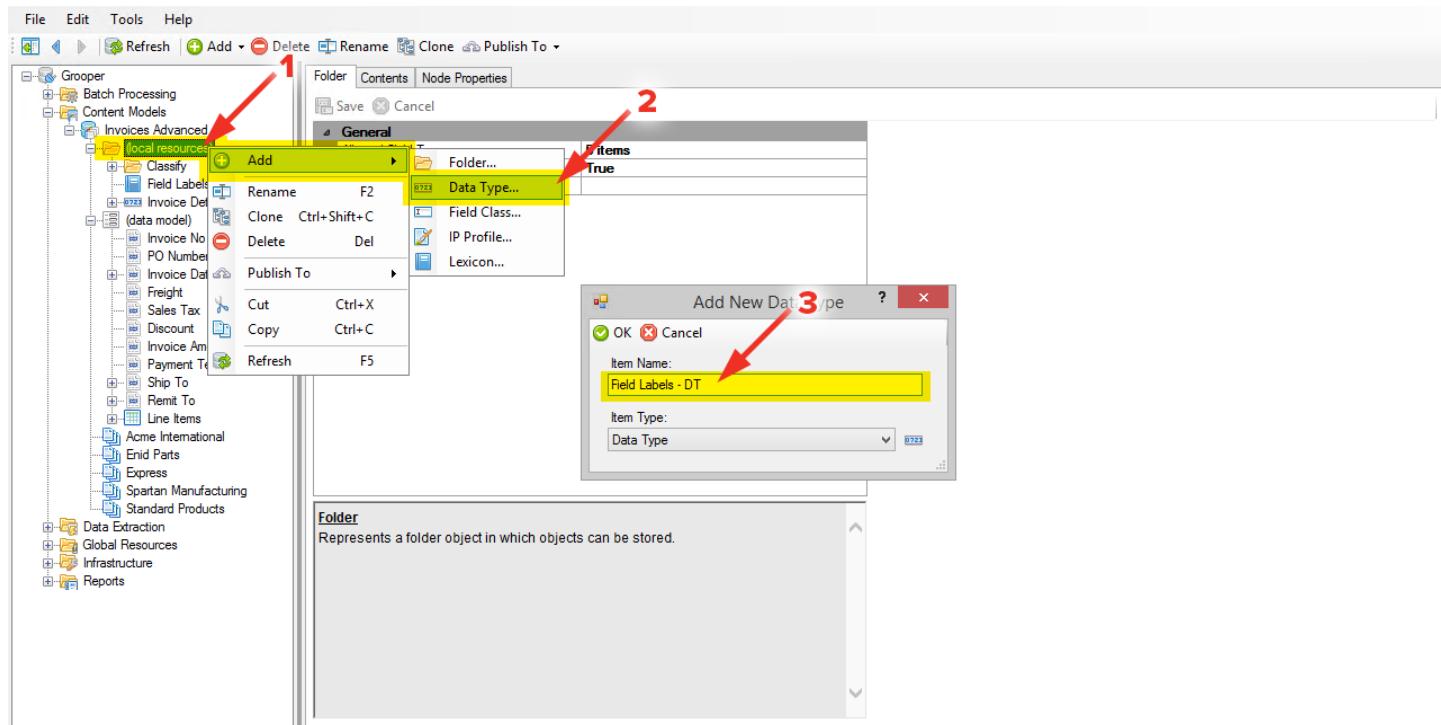
PO / Release Number=PO Number
 Purchase Order:=PO Number
 Remit To:=Remit To
 Sales Order No
 Sales Tax
 Ship=Ship To
 Ship Date
 Ship To
 Ship To:=Ship To
 Ship Via
 Shipped To:=Ship To
 Shipped Via:
 Ship-To Number
 Sold To
 Taken By:
 Terms
 Total Due:=Invoice Amount
 Total Packages:
 Total Weight:
 Tracking Number
 Vendor Number:

CREATE THE DATA TYPE THAT WILL LEVERAGE THE FIELD LABELS LEXICON

The [Field Labels Lexicon](#) by itself doesn't accomplish anything. It needs to be leveraged by an extractor for its purpose to become apparent.

STEP 1 – CREATE NEW DATA TYPE IN LOCAL RESOURCES

- (1) Right-click the (local resources) folder node and (2) Add > Data Type... (3) Name it [Field Labels – DT](#).

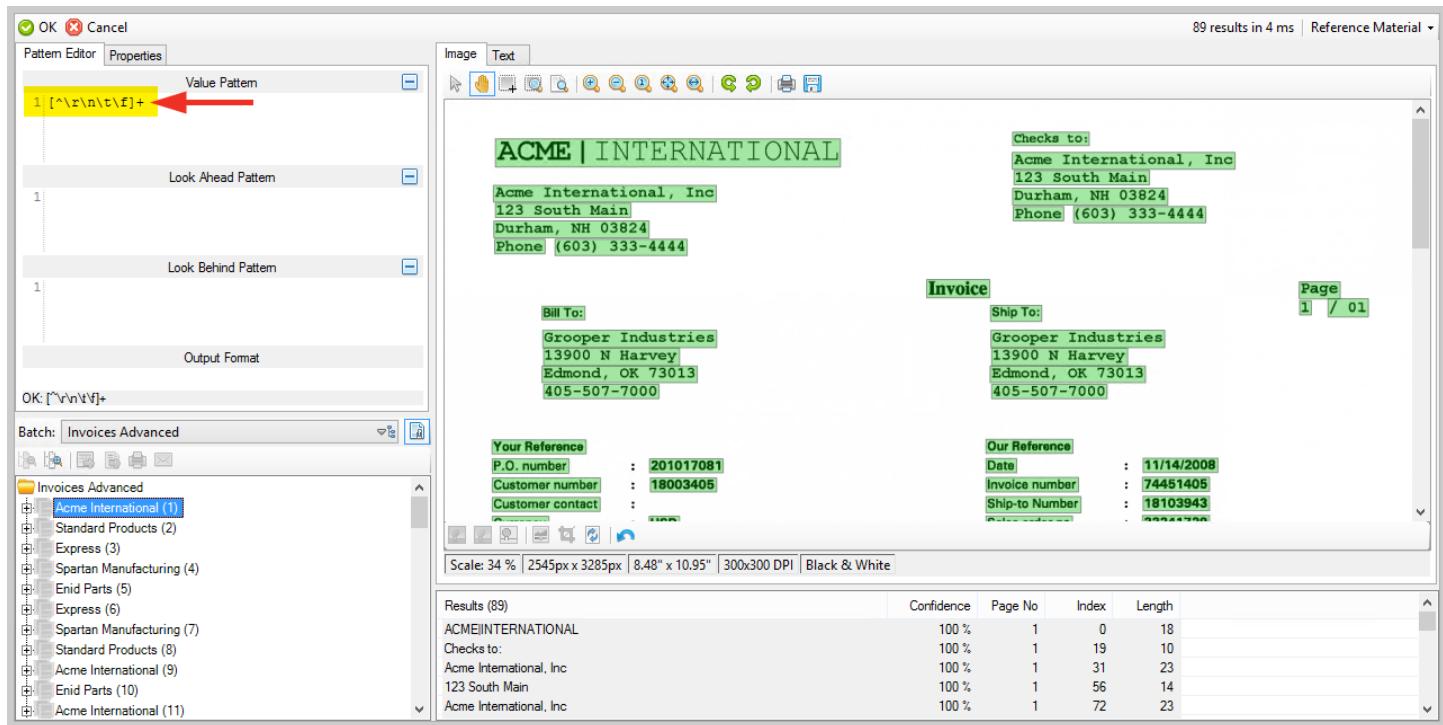


STEP 2 – CHEAT CODE REGEX

With the **Data Type** made, click on the ellipsis button for the **Pattern** property to pull up a **Pattern Editor**. In the **Properties** tab, in the **General** section, within **Text Preprocessing**, be sure **Tab Marking** is set to **Enabled**. Use the following pattern:

[^\r\n\t\f]+

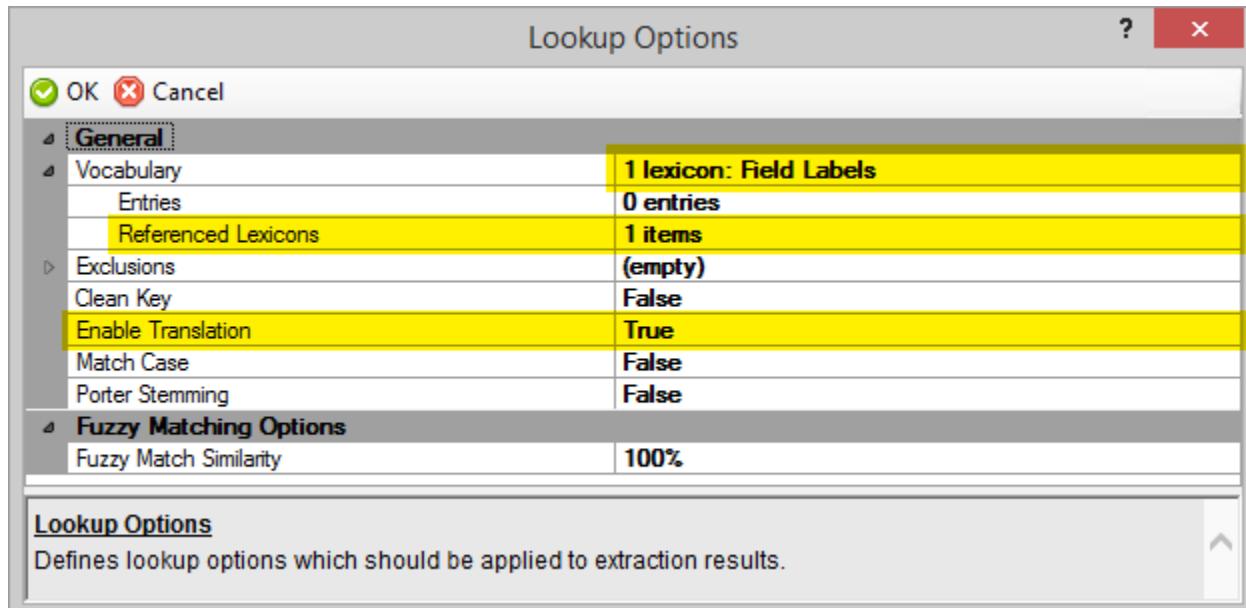
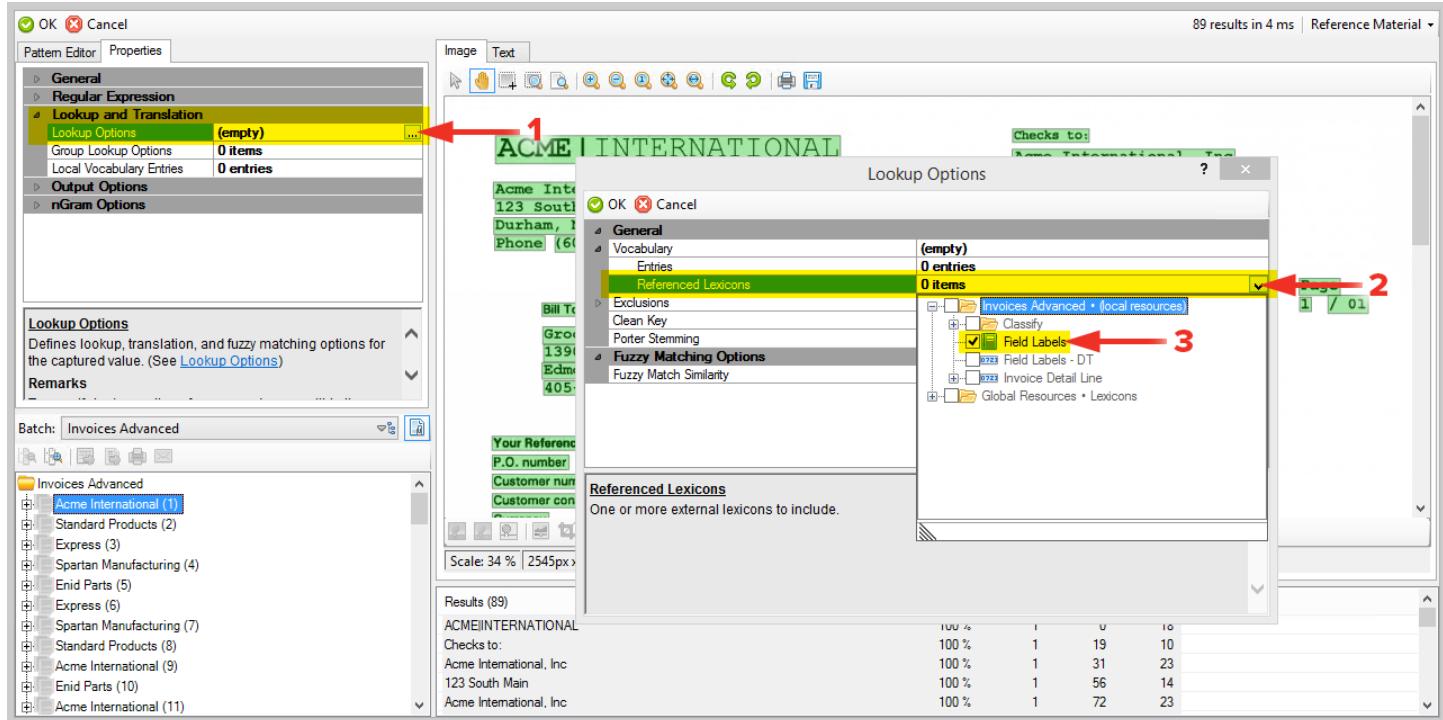
This very simple pattern is lovingly referred to as the **Cheat Code**, because it very effectively grabs every chunk of logical data on a document. The character set **[]** starting with the caret **^** determines not, then following it is carriage returns **\r**, new line feeds **\n**, tabs **\t**, and form feeds **\f** (these are the invisible characters at the end of documents that tell a document to continue to the next page.) So NOT any of those things, and the one or more quantifier **+** says as many of those in a row as you can get until you get to one of the things listed to not get.



STEP 3 – LOOKUP AND TRANSLATION

Now that our **RegEx** pattern is grabbing every logical chunk of data, let's filter that out by comparing it against the **Field Labels Lexicon** that was just made. Click on the **Properties** tab and expand the **Lookup and Translation** section. **(1)** Select the **Lookup Options** property and click the ellipsis button to bring up the **Lookup Options** window. In this window expand the **Vocabulary** section and **(2)** select the **Referenced Lexicons** property. Click the drop-down and within that menu **(3)** put a check mark in the box for the **Field Labels Lexicon** within **Invoices Advanced • (local resources)**. Finally, set the **Enable Translation** property to **True**.

This will limit the **Cheat Code RegEx** pattern to only return results that are listed within the **Lexicon**, as well as allow the translations that were built into the **Lexicon** to function.



STEP 4 – VIEWING RESULTS

With the [Acme International \(1\)](#) document selected in the [Batch Viewer](#), notice now that the results being returned are only those listed within the [Lexicon](#). You can also see that translation is working in that **Checks to:** is being returned as **Remit To**, and **P.O. Number** is being returned as **PO Number**. Feel free to click through more documents within the [Batch Viewer](#) to see those results as well.

Pattern Editor (Properties tab selected):

- General
- Regular Expression
- Lookup and Translation
 - Lookup Options: Vocabulary(100%)
 - Group Lookup Options: 0 items
 - Local Vocabulary Entries: 0 entries
- Output Options
- nGram Options

Lookup Options: Defines lookup, translation, and fuzzy matching options for the captured value. (See [Lookup Options](#))

Remarks

Batch: Invoices Advanced

Results (18)

	Confidence	Page No	Index	Length
Remit To	100 %	1	19	10
Page	100 %	1	199	4
Bill To	100 %	1	205	8
Ship To	100 %	1	214	8
PO Number	100 %	1	393	11
Date	100 %	1	415	4
Account Number	100 %	1	432	15
Invoice Number	100 %	1	457	14

STEP 5 – PROBLEM WITH MAIL PAYMENT TO:

In browsing the other documents you'll soon come to a problem. (1) On [Express \(3\)](#), **Mail Payment To:** is a desired feature, and listed in the [Field Labels Lexicon](#), but it's not being returned. This is happening because the default settings of [Tab Marking](#) are preventing the space between the strings **Mail Payment To:** and **Express Manufacturing** from being read as a tab.

Pattern Editor (Properties tab selected):

- General
 - Extraction Mode: RegEx
 - Case Sensitive: False
- Text Preprocessing
 - Tab Marking: Enabled
 - Minimum Tab Width: 0.2
 - Character Size Ratio: 200%
 - Paragraph Marking: Disabled
 - Vertical Tab Marking: Disabled
 - Ignore Control Characters: None
- Regular Expression
- Lookup and Translation

Tab Marking: When enabled, converts space characters to tab characters based on the width of the gap occupied by the space character. Tabs can be matched using \t in regular expressions. (See [Horizontal Tab Marker](#))

Batch: Invoices Advanced

Results (9)

	Confidence	Page No	Index	Length
Invoice Number	100 %	1	28	15
Bill To	100 %	1	54	8
Invoice Date	100 %	1	63	13
PO Number	100 %	1	88	15
Invoice Amount	100 %	1	150	14
Due Date	100 %	1	191	9
Ship To	100 %	1	305	11
Merchandise Total	100 %	1	772	19

1.1 → Mail Payment To: Express Manufacturing
12333 N Dallas Tollway
Dallas, TX 75032

STEP 6 – ADJUSTING TAB MARKING SETTINGS

- (1) Click the **Text** tab and (2) take note of the string:

MailPaymentTo: Express Manufacturing<\r><\n>

So the **Cheat Code RegEx** pattern would successfully return **MailPaymentTo: Express Manufacturing** as a value, but this is not an entry in the **Field Labels Lexicon**. Because the **Cheat Code** pattern looks for NOT tabs, carriage returns, new line feeds, and form feeds, (3) we need to adjust the **Tab Marking** settings so it will insert a tab character after **MailPaymentTo:**.

Character Size Ratio is the property to adjust to allow this to happen. The tooltip for this reads:

Space characters which have a horizontal width greater than the height of the previous character times this ratio will be converted to tab characters.

Currently the size of the space after **MailPaymentTo:** is not being read as a tab with the **Character Size Ratio** at **200%**, but decreasing this to **140%** will allow the space to be a tab.

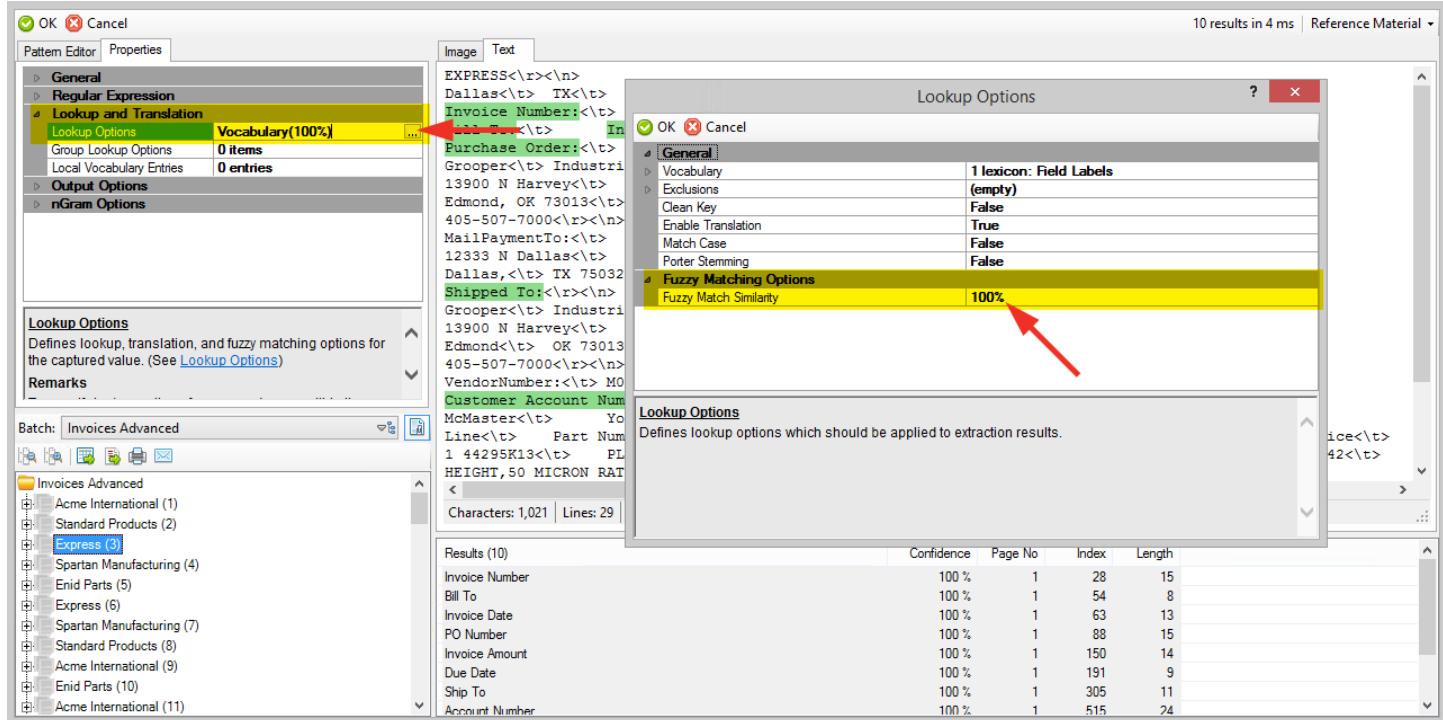
The screenshot shows the Pattern Editor interface. The **Text** tab is highlighted with a yellow box and a red arrow labeled **1**. The **Character Size Ratio** field is highlighted with a yellow box and a red arrow labeled **3**. The main pane displays a list of extracted data, with the entry **MailPaymentTo:<\t> Express Manufacturing<\r><\n>** highlighted with a yellow box and a red arrow labeled **2**. The bottom pane shows a results table with columns: Confidence, Page No, Index, and Length.

	Confidence	Page No	Index	Length
Invoice Number	100 %	1	28	15
Bill To	100 %	1	54	8
Invoice Date	100 %	1	63	13
PO Number	100 %	1	88	15
Invoice Amount	100 %	1	150	14
Due Date	100 %	1	191	9
Ship To	100 %	1	305	11
Account Number	100 %	1	515	24

STEP 7 – FUZZY MATCHING THE LIST

With the tab character successfully inserted, there's one more problem. The **Field Labels Lexicon** lists

Mail Payment To: not **MailPaymentTo:**. The lack of spaces between the words here is a result of an OCR error, and needs to be compensated for. Back in the **Lookup and Translation** section, select the **Lookup Options** property again, and click the ellipsis button to get the **Lookup Options** window again. Notice in the **Fuzzy Matching Options** section that the property **Fuzzy Match Similarity** is at **100%**, basically meaning **Fuzzy Matching** is off. This needs to be lowered to allow **OCR** errors to be translated properly.

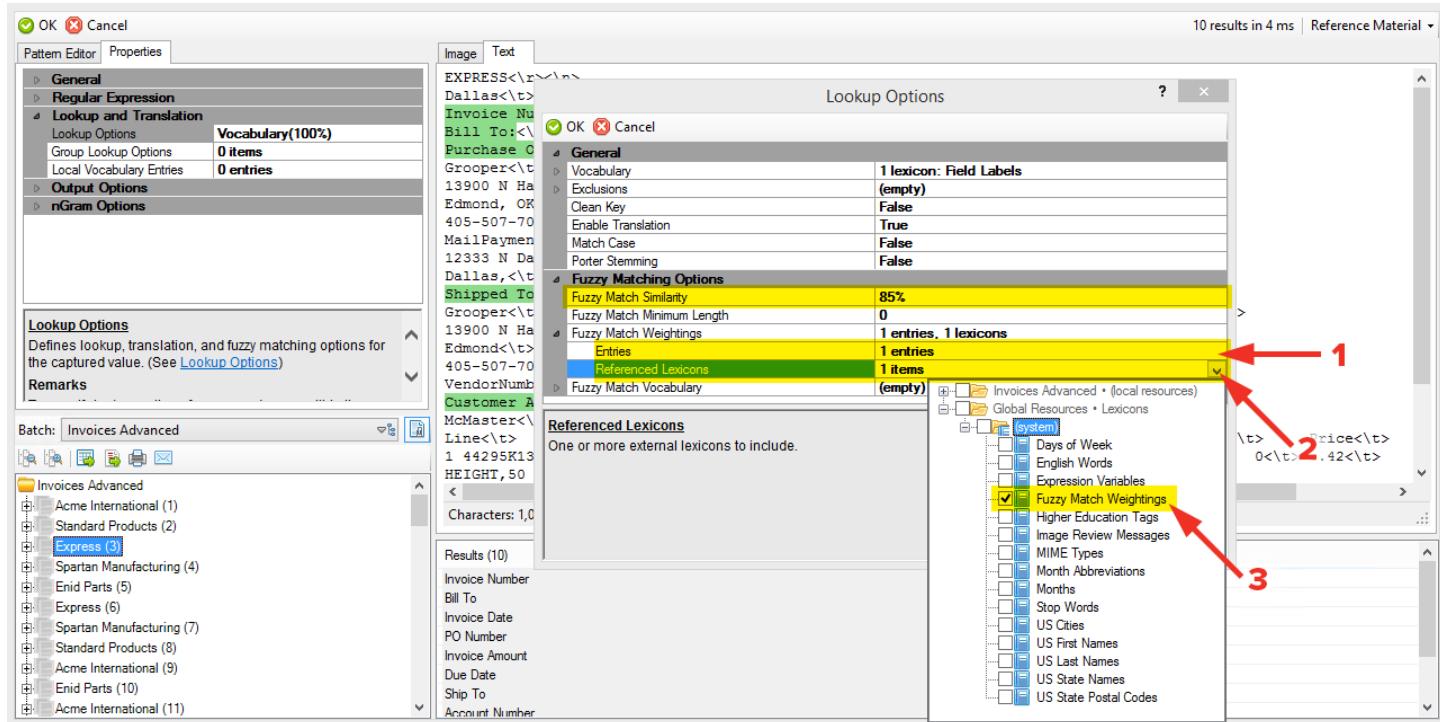


STEP 8 – LOWERING FUZZY MATCH SIMILARITY

(1) Set the **Fuzzy Match Similarity** property to **85%** and new properties will be exposed. (2) Add the following to the **Entries** list from within the **Fuzzy Match Weightings** properties area:

i/=0.25

Also, (3) click the drop-down for **Referenced Lexicons** and check the **Fuzzy Match Weightings Lexicon** in the **Global Resources • Lexicons > (system)** area. This is a system **Lexicon** with several entries for commonly missed OCR characters and reduced weightings to allow common errors to not be such a heavy penalty on **Fuzzy Matching**.



STEP 9 – VIEWING SUCCESSFUL RESULTS

(1) With these properties adjusted, **MailPaymentTo:** will now be found, (2) and because of a translation in the **Field Lables Lexicon** it will be returned as **Remit To**.

The screenshot shows the Grooper Pattern Editor interface. On the left, there's a sidebar with tabs for General, Regular Expression, Lookup and Translation, Output Options, and nGram Options. Under Lookup and Translation, 'Vocabulary(85%)' is selected. Below that is a 'Lookup Options' section with 'Group Lookup Options' and 'Local Vocabulary Entries' both set to 0 items/entries. The main pane displays a list of search results. One result, 'MailPaymentTo:<\t>', is highlighted with a yellow box and has a red arrow pointing to it from the number '1'. Another result, 'Remit To', is also highlighted with a yellow box and has a red arrow pointing to it from the number '2'. The results table at the bottom shows the following data:

	Confidence	Page No	Index	Length
PO Number	100 %	1	88	15
Invoice Amount	100 %	1	150	14
Due Date	100 %	1	191	9
Remit To	100 %	1	225	14
Ship To	100 %	1	305	11
Vendor Number:	100 %	1	490	13
Account Number	100 %	1	515	24
Merchandise Total	100 %	1	772	19

The power of, and reason for this extractor being built the way it is (specifically leveraging the **Lexicon** that utilizes translation), is the ease with which it can be updated. Say you're a company processing a lot of invoices for a plethora of different vendors, at any point, if you add a new vendor and they label something like **Invoice Number** differently than what you have in the **Lexicon** (perhaps something like **Inv.Numb.**), you can simply add an entry to the **Lexicon** and translate it to what you've established as your normalized version, and you're all set. So for this example:

Inv.Numb.=Invoice Number

This will become abundantly clear when we train a **Field Class** to leverage this, as the feature it will train against will always be the normalized translation you've established. As a result, you'll only need to train the **Field Class** once because the feature will always be returned as what you have it translated to. There may be a need to adjust **Context Zones**, but you can handle that on a case by case basis, and it'll be very infrequent.

TACKLING MULTI-LINE FIELD LABELS WITH ORDERED ARRAYS

The field labels that have been sought so far have been easy to work with, if for no other reason than that they've been on a single line. When field labels are on multiple lines we logically read them as one unit, but **OCR** synthesis doesn't understand the relationship of vertically oriented labels, so it does what it always does and puts lines of text in continuous, logical lines. As a result, writing simple **RegEx** patterns to consistently capture these labels can be a challenge. Thankfully, once again, **Grooper** has an elegant solution to this problem.

STEP 1 – OBSERVING THE MULTILINE PROBLEM

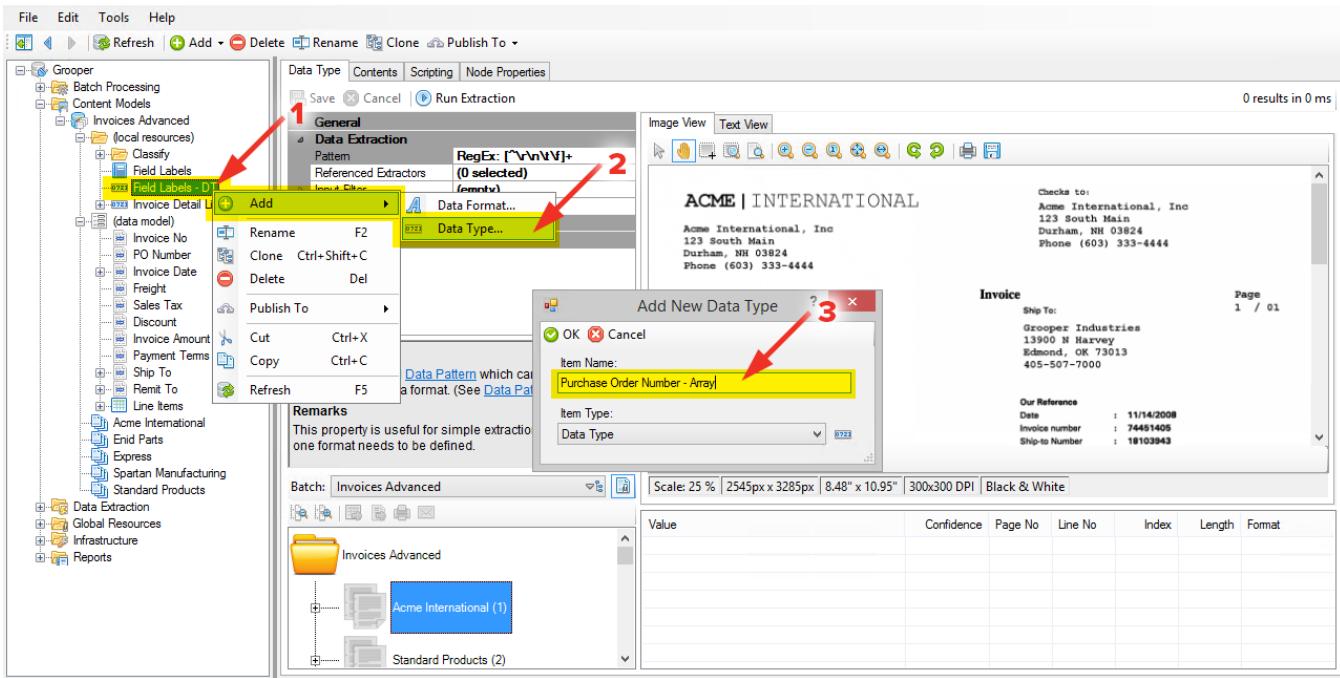
With the **Field Labels – DT Data Type** still selected, (1) select **Enid Parts (5)** from the **Batch Viewer**. There are several field labels here, but the problem is they are on different lines. While this makes visual sense, and you can read it fine, (2) if you look at the **Text** tab you'll notice (3) there's quite a lot between **PURCHASE** and **ORDER NUMBER** (we'll focus on this feature as it is a field in our **Data Model** we wish to populate.) This document is structured enough to allow a somewhat complex **RegEx** pattern to work, but there's a simpler solution to return **PURCHASE** and **ORDER NUMBER** as a single result.

The screenshot shows the Grooper Pattern Editor interface. The top navigation bar includes 'OK', 'Cancel', 'Properties', and tabs for 'Image' and 'Text'. A red arrow labeled '2' points to the 'Text' tab. The main workspace displays a 'Value Pattern' section with the regular expression `1 [^\r\n\t\f]+`. Below it are sections for 'Look Ahead Pattern' and 'Look Behind Pattern', both containing the value '1'. The 'Output Format' section shows the resulting text. A red arrow labeled '3' points to the text between 'PURCHASE' and 'ORDER NUMBER'. The bottom left pane shows a tree view of 'Invoices Advanced' with 'Enid Parts (5)' selected, indicated by a red arrow labeled '1'. The bottom right pane shows a table titled 'Results (15)' with columns: Confidence, Page No, Index, and Length. The table lists various invoice-related fields and their confidence levels.

	Confidence	Page No	Index	Length
Invoice Date	100 %	1	20	12
Invoice Number	100 %	1	33	9
Page	100 %	1	43	4
Phone	100 %	1	74	3
Fax:	100 %	1	94	4
Remit To	100 %	1	114	9
Ship To	100 %	1	238	4
Shin Via	100 %	1	338	8

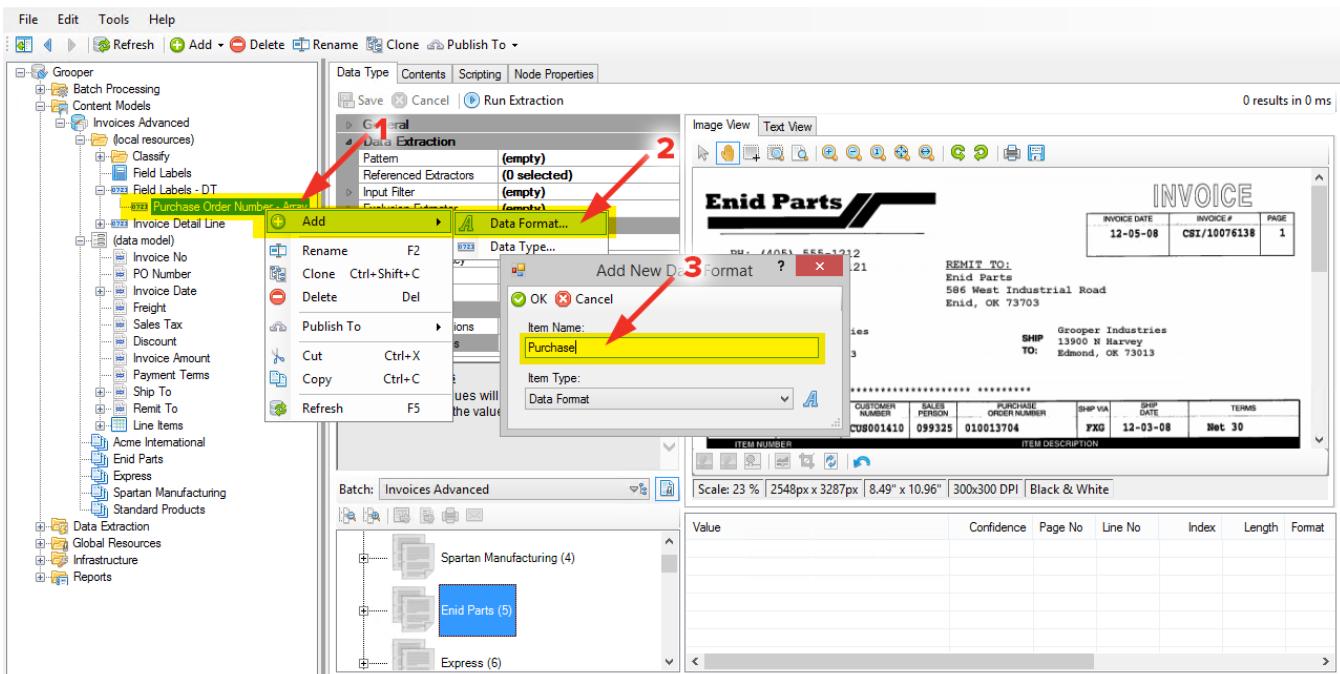
STEP 2 – ADDING CHILD DATA TYPE TO FIELD LABELS – DT

Up to this point we've seen child [Data Formats](#) to [Data Types](#), and understand that they return their results to their parent [Data Type](#). However, A [Data Type](#) has yet to be added as a child to another [Data Type](#), but it will work the same way by returning its result to its parent object. **(1)** Right-click on the [Field Labels – DT Data Type](#) and **(2)** Add > [Data Type...](#) and **(3)** name it [Purchase Order Number – Array](#).



STEP 3 – ADDING CHILD DATA FORMATS

(1) Right-click on the [Purchase Order Number – Array Data Type](#) **(2)** and [Add > Data Format...](#) **(3)** Name it [Purchase](#). Add another [Data Format](#) and name it [Order Number](#).



STEP 4 – SIMPLE REGEX PATTERN – PURCHASE TO PO

- (1) With the **Purchase Data Format** selected, (2) and the **Enid Parts (5)** document in the Batch Viewer, (3) type the following in the **Value Pattern**:

PURCHASE

- (4) In the **Output Format** type the following:

po

- (5) This will find **PURCHASE** in the **Enid Parts (5)** document, but return it as **po**.

The screenshot shows the Grooper ACE software interface. On the left, the navigation tree includes categories like Grooper, Batch Processing, Content Models, and various invoices and parts. The 'Batch Processing' section is expanded, showing 'Purchase Order Number - Array' and 'Purchase'. A red arrow labeled '1' points to the 'Purchase' node.

In the center, the 'Pattern Editor' window is open. It has tabs for 'Data Format' and 'Node Properties'. Under 'Data Format', the 'Value Pattern' field contains 'PURCHASE' (highlighted with a yellow box and a red arrow labeled '3'). Below it is the 'Look Ahead Pattern' field. Under 'Output Format', the value 'po' is typed (highlighted with a yellow box and a red arrow labeled '4').

To the right, the 'Batch Viewer' displays the 'Enid Parts (5)' document. A red arrow labeled '2' points to the document in the list. The document preview shows purchase details like PH: (405) 555-1212, FAX: (405) 444-2121, and a table of items shipped.

At the bottom, the 'Results' table shows a single row with the value 'po' (highlighted with a yellow box and a red arrow labeled '5'). The table columns include Confidence, Page No, Index, and Length.

STEP 5 – SIMPLE REGEX PATTERN – ORDER NUMBER TO NUMBER

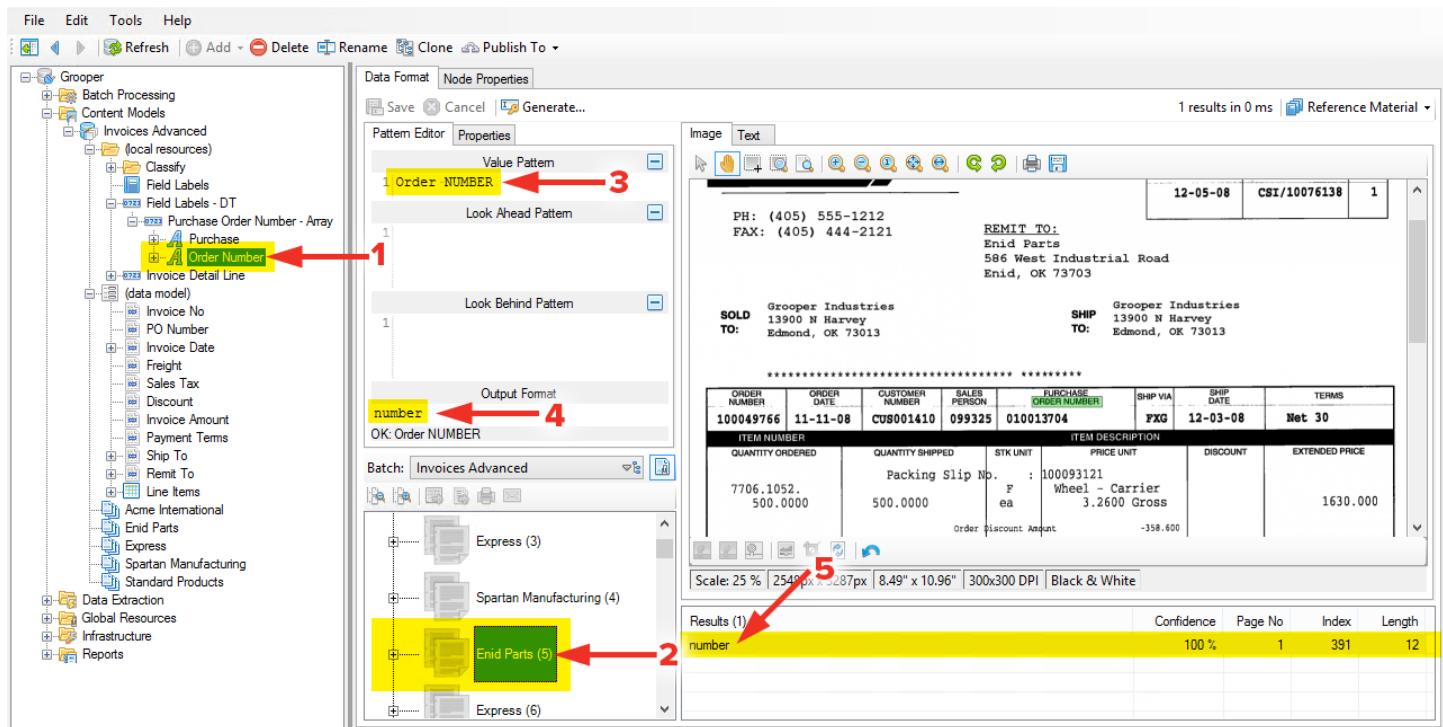
- (1) Select the Order Number Data Format, (2) and the Enid Parts (5) document in the Batch Viewer,
- (3) type the following in the Value Pattern:

ORDER NUMBER

- (4) In the output format type the following:

number

- (5) This will find **ORDER NUMBER** in the **Enid Parts (5)** document, but return it as **number**.



STEP 6 – SEEING RESULTS ON THE PARENT DATA TYPE

(1) Select the Purchase Order Number – Array Data Type and (2) notice the results from the child Data Formats being returned. We don't want separate results, so we'll setup this Data Type to leverage a Collation Method called an Ordered Array.

The screenshot shows the Grooper ACE interface with the following details:

- File Bar:** File, Edit, Tools, Help, Refresh, Add, Delete, Rename, Clone, Publish To.
- Left Sidebar:** Grooper, Batch Processing, Content Models, Invoices Advanced (selected), Field Labels (local resources), Field Labels - DT, Invoice Detail Line (data model), Invoice No, PO Number, Invoice Date, Freight, Sales Tax, Discount, Invoice Amount, Payment Terms, Ship To, Remit To, Line Items, Acme International, Enid Parts, Express, Spartan Manufacturing, Standard Products, Data Extraction, Global Resources, Infrastructure, Reports.
- Central Area:**
 - Data Type Tab:** General, Data Extraction, Output (selected), Deduplication.
 - Output Settings:** Collation Method: Normal, Output Extractor Key: False, Output On: Order, Result Filter: (empty).
 - Data Type Description:** A Data Type defines extraction logic for a distinct type of data, such as a field value or a table row. Each data type defines one or more extractors, along with settings which control how the extractor results are transformed into a final result set.
 - Batch Selection:** Invoices Advanced.
 - Results View:** Shows two results in 0 ms. It includes an Image View (receipt) and a Text View table.
- Bottom Area:** Results (2) table showing the extracted data.

Annotations with red numbers:

- 1.1**: Points to the "Enid Parts (5)" item in the sidebar under "Content Models".
- 2**: Points to the "po number" entry in the "Results (2)" table.
- 2.1**: Points to the "SHIP NUMBER" column header in the Text View table.

STEP 7 – SETTING UP THE ORDERED ARRAY

- (1) In the Output section set the Collation Method to OrderedArray. (2) In the Array Options section set the Array Layout to Vertical. (3) In the Vertical Array section, set the Maximum Vertical Distance to 0.25, Alignment to Center, and Alignment Tolerance to 0.1. Save and Run Extraction.

This will take the results of the child Data Formats and combine them into one result if they are found in a vertical layout, within a quarter of an inch of one another, aligned in their center and a tolerance of what is considered centered by 0.1 inches. These are very specific conditions that prevent the possibility of false results being returned by the very simple RegEx patterns that were written within the child Data Formats.

STEP 8 – VIEWING RESULTS ON THE FIELD LABELS – DT PARENT DATA TYPE

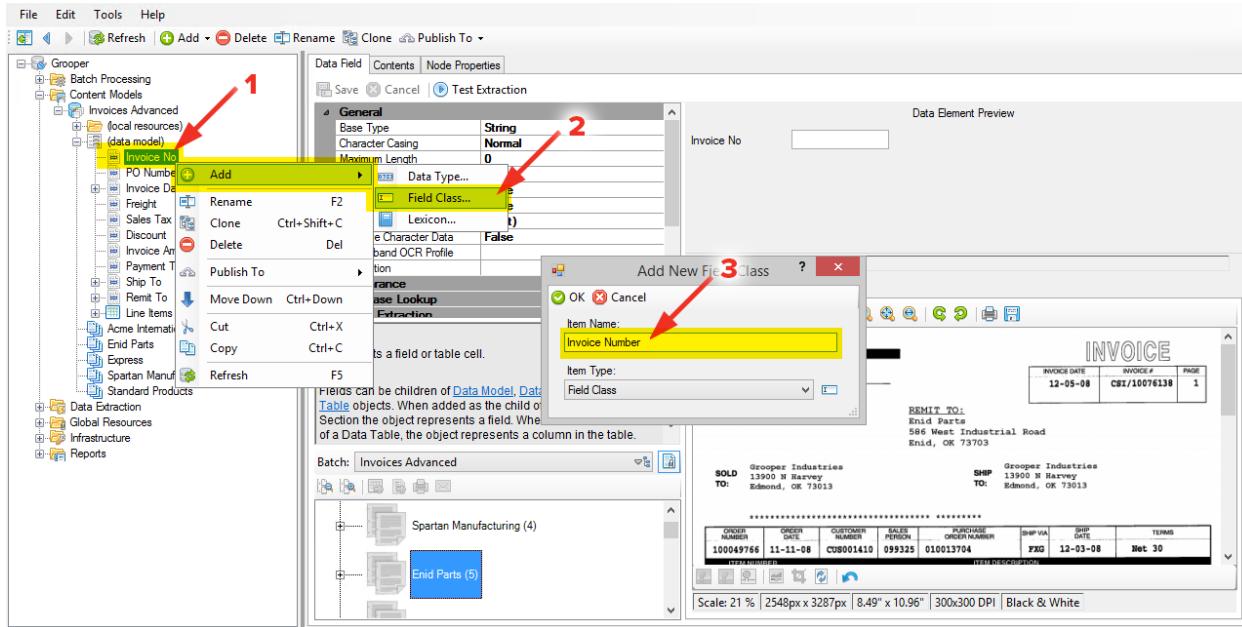
Select the Field Labels – DT Data Type and click through some documents in the Batch Viewer. Notice on the End Parts (5) document that PURCHASE ORDER NUMBER is being found, and returned as po number.

THE NEXT FIELD CLASS – USING THE NEWLY BUILT FIELD LABELS – DT DATA TYPE

All the work that just went into making the [Field Labels – DT Data Type](#) will make the creation of the last few [Field Classes](#) very straight forward. There is now one [Feature Extractor](#) to work with that will consistently return desired results from this set of documents, accounting for [OCR](#) errors as well as translating disparate values to reduce the amount of training that is required for the features within each [Field Class](#).

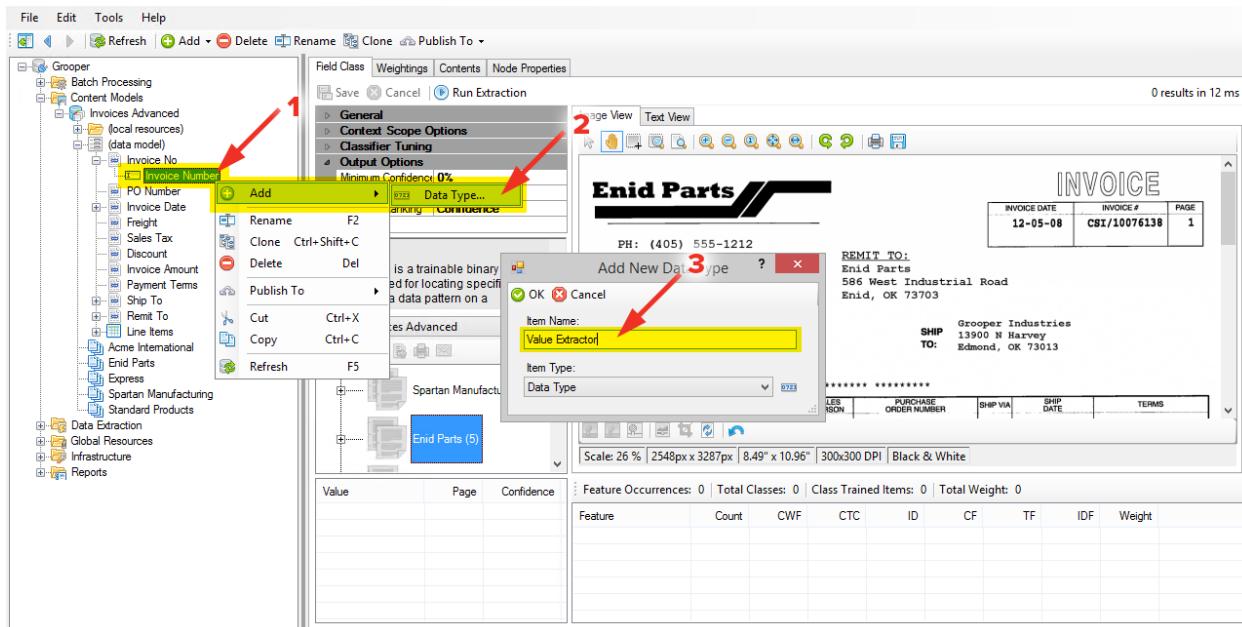
STEP 1 – ADDING A FIELD CLASS TO INVOICE NO

- (1) Select the [Invoice No Data Field](#) and (2) right-click [Add > Field Class...](#) (3) Name it [Invoice Number](#).



STEP 2 – ADDING A DATA TYPE

- (1) Select the [Invoice Number Field Class](#), (2) right-click [Add > Data Type...](#) (3) Name it [Value Extractor](#).



STEP 3 – ADDING DATA FORMATS

Add three Data Formats to the [Value Extractor Data Type](#). Name them [Acme/Standard/Express](#), [Spartan](#), and [Enid](#).

The screenshot shows the Grooper ACE software interface. On the left, the navigation tree includes 'Grooper', 'Batch Processing', 'Content Models', 'Invoices Advanced' (selected), 'PO Number', 'Invoice Date', 'Freight', 'Sales Tax', 'Discount', 'Invoice Amount', 'Payment Terms', 'Ship To', 'Remit To', 'Line Items', 'Acme International', 'Enid Parts', 'Express', 'Spartan Manufacturing', 'Standard Products', 'Data Extraction', 'Global Resources', 'Infrastructure', and 'Reports'. The 'Invoices Advanced' node has a yellow box around it, and a red arrow points from the 'Value Extractor' node under it to the 'Value Pattern' section of the central editor.

The central area contains a 'Data Format' editor with tabs for 'Node Properties' and 'Pattern Editor'. The 'Pattern Editor' tab is active, showing sections for 'Value Pattern', 'Look Ahead Pattern', 'Look Behind Pattern', and 'Output Format'. Below these is an 'Empty Expression' section. A 'Batch:' dropdown is set to 'Invoices Advanced'. To the right of the editor is a preview window showing an 'INVOICE' document for 'Enid Parts'. The document includes fields like 'INVOICE DATE', 'INVOICE #', 'PAGE', 'PH:', 'FAX:', 'REMIT TO:', 'SOLD TO:', 'SHIP TO:', and a detailed table of items. At the bottom of the preview window, it says 'Scale: 25 % | 2548px x 3287px | 8.49" x 10.96" | 300x300 DPI | Black & White'.

STEP 4 – VALUE PATTERNS AND OUTPUT FORMATS

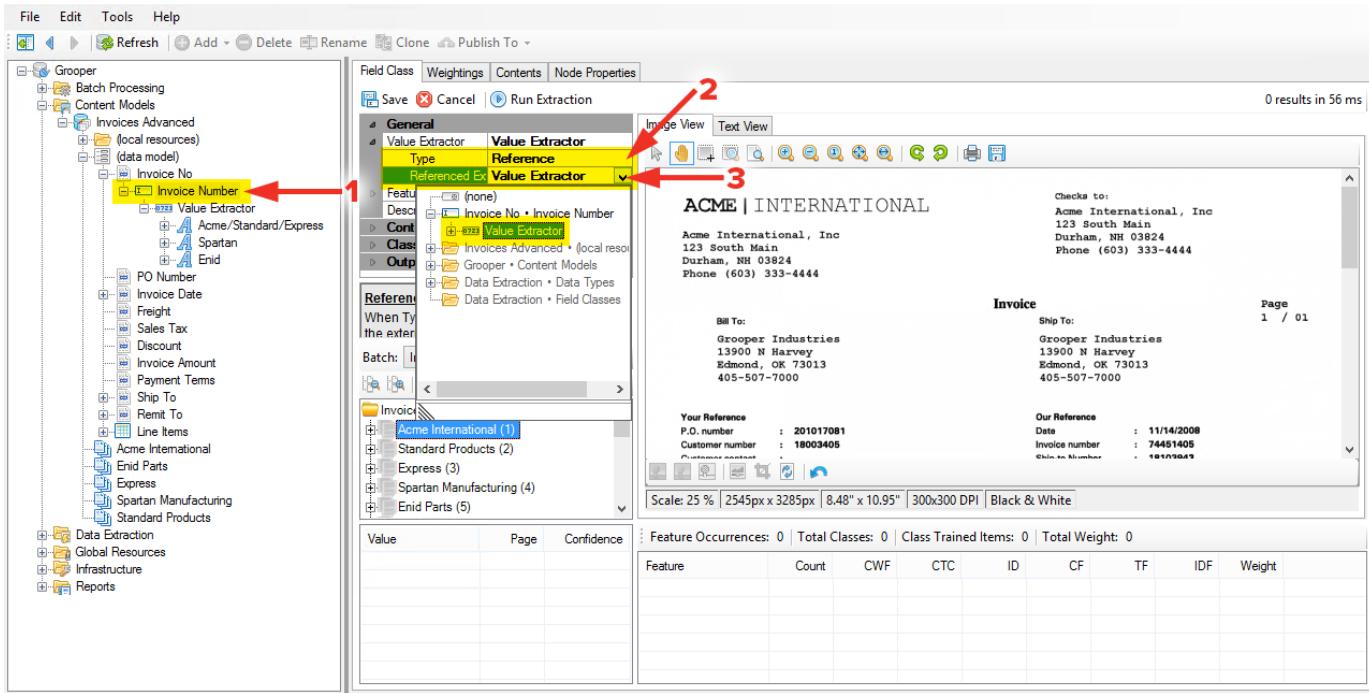
- (1) Use the Information below to populate the **Pattern Editor** of the newly created **Data Formats**.
- (2) Take note of the **Output Formats** for **Enid** and **Spartan**. Notice the syntax with a **number**, followed by a **colon**, and the string **Number**. This leverages the **@Number** expression variable and forces it to return the known number variants that its **RegEx** pattern is compensating for. Without this, the **@Number** expression variable would capture something like an **1**, but still return it as an **1**, as opposed to the digit it's commonly mis-OCR'd for, a **1.**

	Value Pattern	Output Format
Acme/Standard/Express	[0-9]{6,16}	
Spartan	([A-Z]{2})([@Number]{2})-(@Number){6})	{1}{2:Number}-{3}
Enid	([A-Z]{3})/(@Number){8})	{1}/{2:Number}

The screenshot shows the Grooper ACE software interface. On the left, the navigation tree includes sections like Batch Processing, Content Models, Invoices Advanced, and various company modules (Acme International, Enid Parts, Spartan Manufacturing). The main workspace displays the 'Data Format' editor for 'Invoices Advanced'. The 'Pattern Editor' tab is active, showing a value pattern `1: ([A-Z]{3})/(@Number){8})` highlighted with a yellow box and a red arrow labeled '1'. Below it, the 'Output Format' section shows `(1) / {2:Number}` highlighted with a yellow box and a red arrow labeled '2'. To the right, a sample invoice from 'Enid Parts' is displayed, showing fields like INVOICE DATE (12-05-08), INVOICE # (CSI/10076138), and PAGE (1). The invoice details include shipping and receiving information for Grooper Industries.

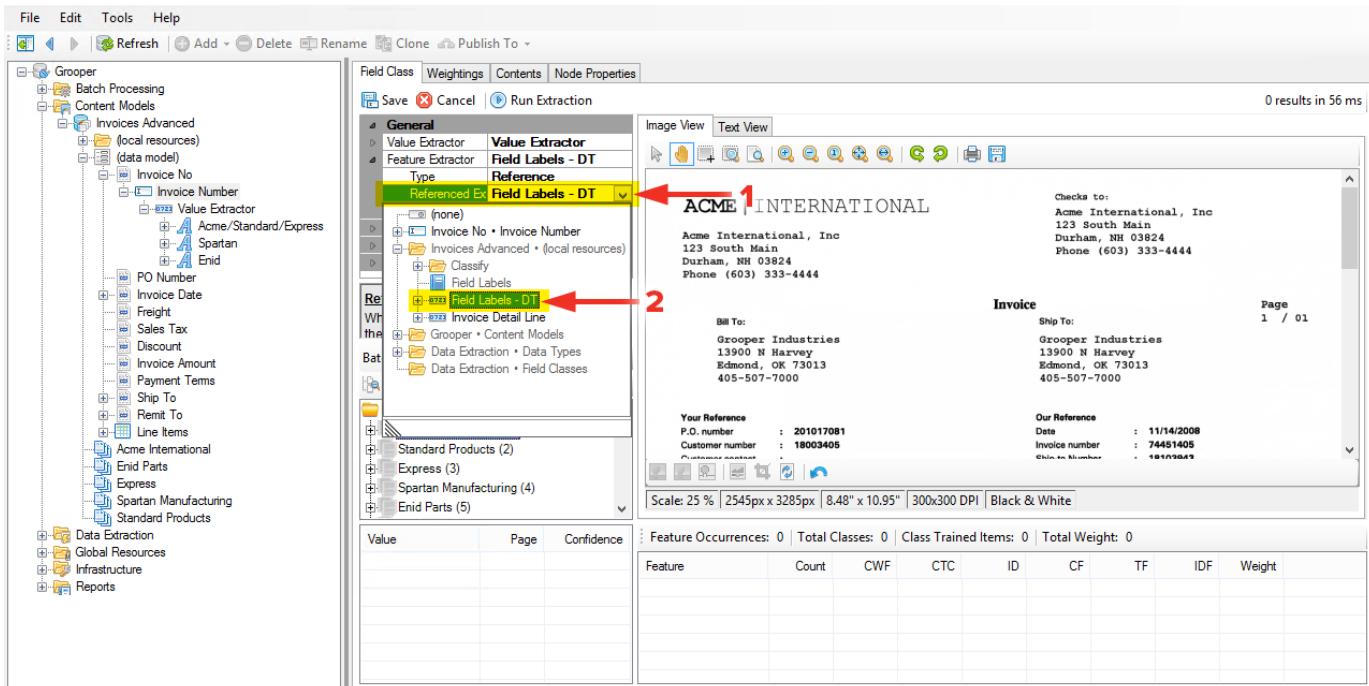
STEP 5 – SET THE VALUE EXTRACTOR

(1) Select the **Invoice Number Field Class** and set the (2) **Value Extractor Type** to **Reference** and (3) set the **Referenced Extractor** to the **Value Extractor Data Type** that is a child of the **Invoice Number Field Class**.



STEP 6 – SET THE FEATURE EXTRACTOR

(1) Expand the **Feature Extractor** properties and change the **Referenced Extractor** from the **nGrams 1-3 Data Type**, (2) to the **Field Labels – DT Data Type** within the **Invoices Advanced • (local resources)**. Save and Run Extraction with the **Acme International (1)** document selected in the Batch Viewer.



STEP 7 – TRAIN A FEATURE

(1) On the **Acme International (1)** document, (2) select the **74451405** value in the **Value List View** and right-click **Train as Positive**. (3) This will add the feature **invoice number** to the **Positive Weightings**, as well as negatively weight all other features captured within the **Context Zones** of the other values that were listed.

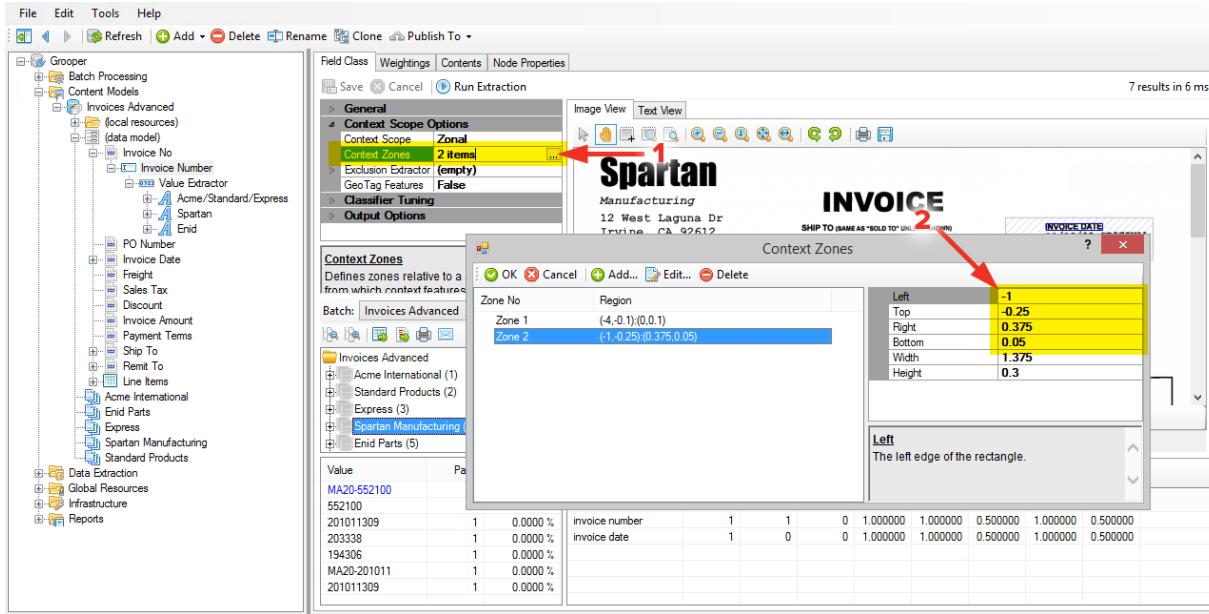
STEP 8 – FEATURE DETECTION PROBLEM – BAD CONTEXT ZONES

Due to translation in the **Lexicon** that the **Field Labels – DT Data Type** is using, all variations of **Invoice Number** will be returned the same, so, this one feature being trained should return **100%** on all **DocTypes**. However, (1) select **Spartan Manufacturing (4)** and (2) notice the correct value in the **Value List View** being returned at **~70%**. (3) Notice in the **Page Viewer** that the **Zone 2 Context Zone** is too tall, and consequently grabbing the invoice date feature.

STEP 9 – ADJUSTING THE CONTEXT ZONES

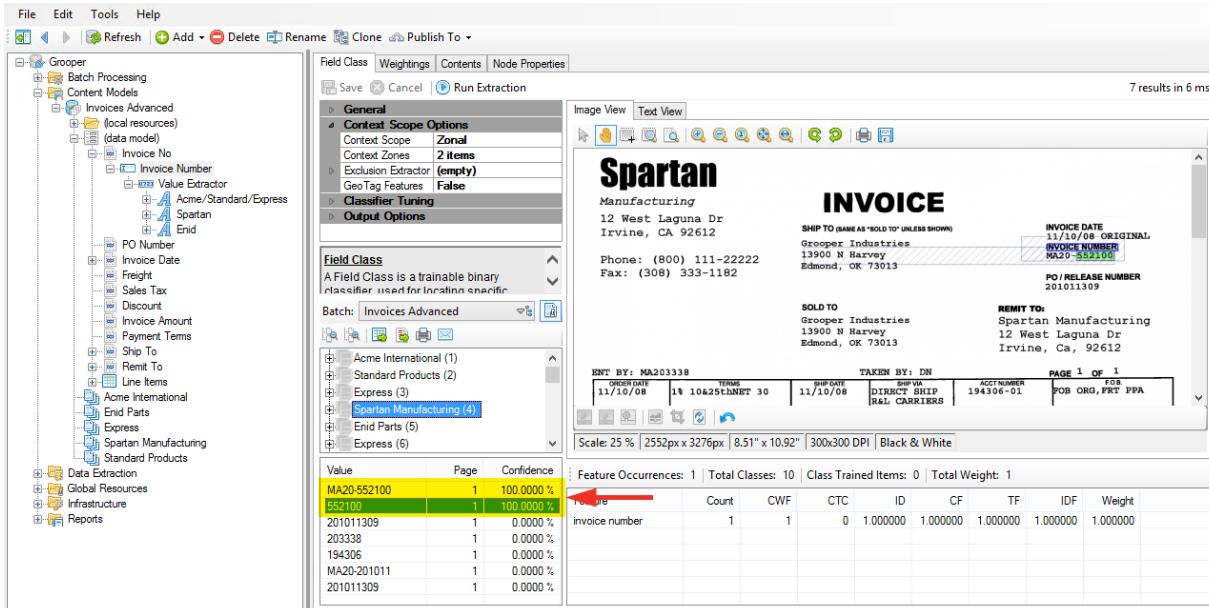
(1) Expand the **Context Scope Options**, select the **Context Zones** property, and click the ellipsis button to bring up the **Context Zones** property window. Feel free to click **Zone 2** and manually edit it, but (2) the proper values to allow this zone to function properly have been provided as well as the values for **Zone 1**. After closing the property window, **Save** and **Run Extraction**.

Zone 1 – Left: **-4** Top: **-0.02** Right: **0.35** Bottom: **0.1**
Zone 2 – Left: **-1** Top: **-0.25** Right: **0.375** Bottom: **0.05**



STEP 10 – DUPLICATE ENTRIES

Notice for **Spartan Manufacturing (4)** there are two values listed at **100%** confidence, but the values themselves are not the same. Look more closely and you'll notice that the second value is the integer portion of the first value after the hyphen. This is because there are overlapping results being returned by the child **Data Formats** of the **Value Extractor Data Type**.



STEP 11 – ANALYZING THE VALUE EXTRACTOR DATA TYPE FOR DUPLICATE VALUES

Select the **Value Extractor Data Type** and observe the results being returned in the **Results List View**. Take specific heed of **(1)** the **Format** **(2)** and **Pattern** columns. The **MA20-552100** result is being returned by the **Spartan** format, and the **552100** result is being returned by the **Acme/Standard/Express** format. The latter's format is generic enough to capture this six-digit value.

The screenshot shows the Grooper ACE interface with the following details:

- Data Type:** Invoices Advanced
- Output - Deduplication:** Duplicate Locations: False, Duplicate Values: False
- Results List View:**
 - Results (7):** MA20-552100, 552100, 201011309, 203338, 194306, MA20-201011, 201011309
 - Confidence:** 100 %
 - Page ...:** 1
 - Line No:** 213, 218, 285, 437, 572, 844, 849
 - Index:** 11, 6, 9, 6, 6, 11, 9
 - Length:** 11, 6, 9, 6, 6, 11, 9
 - Format:** Spartan, Acme/Standard/Express, Acme/Standard/Express, Acme/Standard/Express, Acme/Standard/Express, Spartan, Acme/Standard/Express
 - Pattern:** ([A-Z]{2})(@Number{2})(@Number{6}), [0-9]{6,16}, [0-9]{6,16}, [0-9]{6,16}, [0-9]{6,16}, ([A-Z]{2})(@Number{2})(@Number{6}), [0-9]{6,16}

Two red arrows point to the **Format** and **Pattern** columns in the Results List View table.

STEP 12 – DEDUPLICATE LOCATIONS

Data Types have a property to compensate for this problem called **Deduplicate Locations**. Its tooltip reads:

If True, instances with overlapping zones will be de-duplicated, with precedence given to larger data elements.

So, the zones for the **MA20-552100** and **552100** are definitely overlapping, and because the former pattern is the longer one and the desired result, setting this property to **True** is the solution to this problem.

The screenshot shows the Grooper ACE interface with the following details:

- Data Type:** Invoices Advanced
- Output - Deduplication:** Duplicate Locations: **True**, Duplicate Values: False
- Results List View:**
 - Results (5):** MA20-552100, 201011309, 203338, 194306, MA20-201011
 - Confidence:** 100 %
 - Page ...:** 1
 - Line No:** 213, 285, 437, 572, 844
 - Index:** 11, 9, 6, 6, 11
 - Length:** 11, 6, 9, 6, 11
 - Format:** Spartan, Acme/Standard/Express, Acme/Standard/Express, Acme/Standard/Express, Spartan
 - Pattern:** ([A-Z]{2})(@Number{2})(@Number{6}), [0-9]{6,16}, [0-9]{6,16}, [0-9]{6,16}, ([A-Z]{2})(@Number{2})(@Number{6})

A red arrow points to the **Duplicate Locations** dropdown menu in the Data Type configuration panel.

STEP 13 – SETTING MINIMUM CONFIDENCE

With the adjustment made to the **Value Extractor Data Type** and saved, select the **Invoice Number Field Class**. Because there is only one feature trained, and it is being returned at **100%** confidence, the **Minimum Confidence** can be set to anything above **0%** and all unwanted results will be trimmed.

The screenshot shows the Grooper ACE interface with the 'Output Options' configuration for the 'Invoice Number' field class. The 'Minimum Confidence' field is highlighted with a yellow box and a red arrow. The confidence value is set to 1%. The 'Collation Method' is set to 'Individual'. The 'Instance Ranking' is set to 'Confidence'. Below this, the 'Minimum Confidence' section is expanded, showing the minimum classification confidence value for an instance to be output. The 'Batch' dropdown is set to 'Invoices Advanced'. The 'Feature Occurrences' table shows two occurrences with 100.0000% confidence. The 'Feature' table shows 'invoice number' with a count of 1, CWF of 1, and a weight of 1.000000. To the right, the 'Image View' shows a scanned invoice document with extracted data, including account number, invoice number, invoice date, due date, and amount due.

STEP 14 – SETTING PROPERTIES FOR THE INVOICE NO DATA FIELD

Select the **Invoice No Data Field** and **(1)** set its **Required** property to **True**. **(2)** Set the **Default Extractor Type** to **Reference** and **(3)** set the **Referenced Extractor** to the **Invoice Number Field Class**.

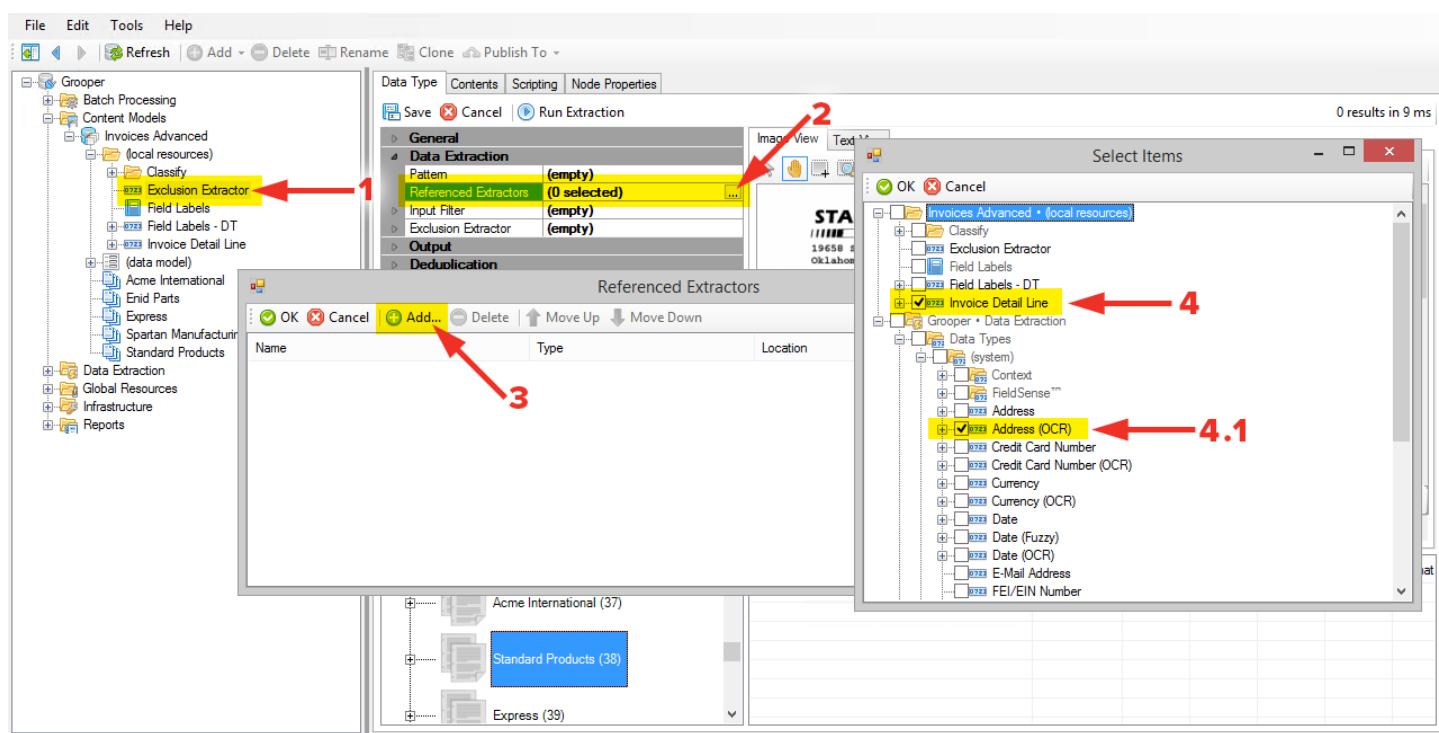
The screenshot shows the Grooper ACE interface with the properties for the 'Invoice No' data field. The 'Required' property is set to 'True' (highlighted with a yellow box and red arrow 1). The 'Default Extractor' type is set to 'Reference' (highlighted with a yellow box and red arrow 2). The 'Referenced Extractor' is set to 'Invoice Number' (highlighted with a yellow box and red arrow 3). The 'Document View' shows a scanned invoice document with extracted data, including account number, invoice number, invoice date, due date, and amount due.

THE NEXT FIELD CLASS – VALUE EXTRACTOR LEVERAGING AN EXCLUSION EXTRACTOR

A Field Class will now be made for the PO Number Data Field. The Value Extractor for this Field Class will be built locally to the PO Number, but it will use a very generic RegEx pattern that will return far more results than is necessary. While this isn't inherently a problem with a Field Class, it's a good opportunity to introduce a new property of Data Types. Data Types can reference other Data Types in multiple ways, and one of those ways is as an exclusion.

STEP 1 – BUILDING THE EXCLUSION EXTRACTOR

(1) Create a Data Type in the (local resources) folder and name it **Exclusion Extractor**. In the Data Extraction section (2) select the **Referenced Extractors** property and click the ellipsis button to bring up the **Referenced Extractors** window. (3) Click the **Add** button in the **Referenced Extractors** window to open the select Items window. (4) Put check boxes on the **Invoice Detail Line Data Type** in the **Invoices Advanced • (local resources)** area, and the **Address (OCR) Data Type** within the **Grooper • Data Extraction > Data Types > (system)** area. **Save** and **Run Extraction** and notice without adding a pattern to this **Data Type**, because it is referencing other already built **Data Types**, it will combine their results and return them.



STEP 2 – ADDING PO NUMBER AND VALUE EXTRACTOR FIELD CLASS AND DATA TYPE

(1) Add a **Field Class** named **PO Number** as a child object to the **PO Number Data Field**. (2) Also, add a **Data Type** named **Value Extractor** as a child object to the **PO Number Field Class**. (3) Add the following as the **Pattern** for this **Data Type**:

[0-9]{5,16}

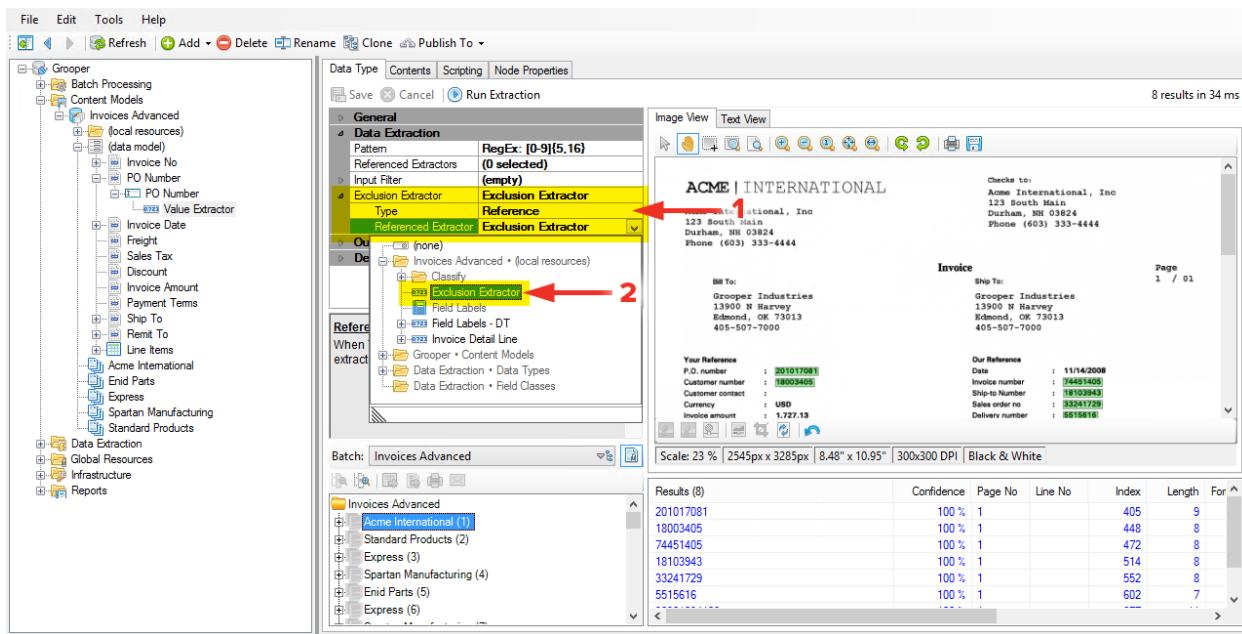
Save and **Run Extraction** and take note of the results returned.

The screenshot shows the Grooper ACE interface with the following details:

- File Bar:** File, Edit, Tools, Help.
- Sidebar:** Grooper, Batch Processing, Content Models, Invoices Advanced (local resources), (data model), Invoice No, PO Number, Value Extractor, Invoice Date, Freight, Sales Tax, Discount, Invoice Amount, Payment Terms, Ship To, Remit To, Line Items, Acme International, End Parts, Express, Spartan Manufacturing, Standard Products, Data Extraction, Global Resources, Infrastructure, Reports.
- Main Area:**
 - Data Type Tab:** General, Data Extraction, Pattern: RegEx: [0-9]{5,16}, Referenced Extractors: (0 selected), Input Filter: (empty), Exclusion Extractor: (empty), Output, Deduplication.
 - Image View:** Shows a preview of an invoice from "ACME | INTERNATIONAL".
 - Text View:** Shows the extracted data for the invoice, including Bill To and Ship To sections, and an Invoice section with details like P.O. number, Date, and Amount.
 - Results Table:** Shows the extracted results in a table format with columns: Results (16), Confidence, Page No, Line No, Index, Length, For.

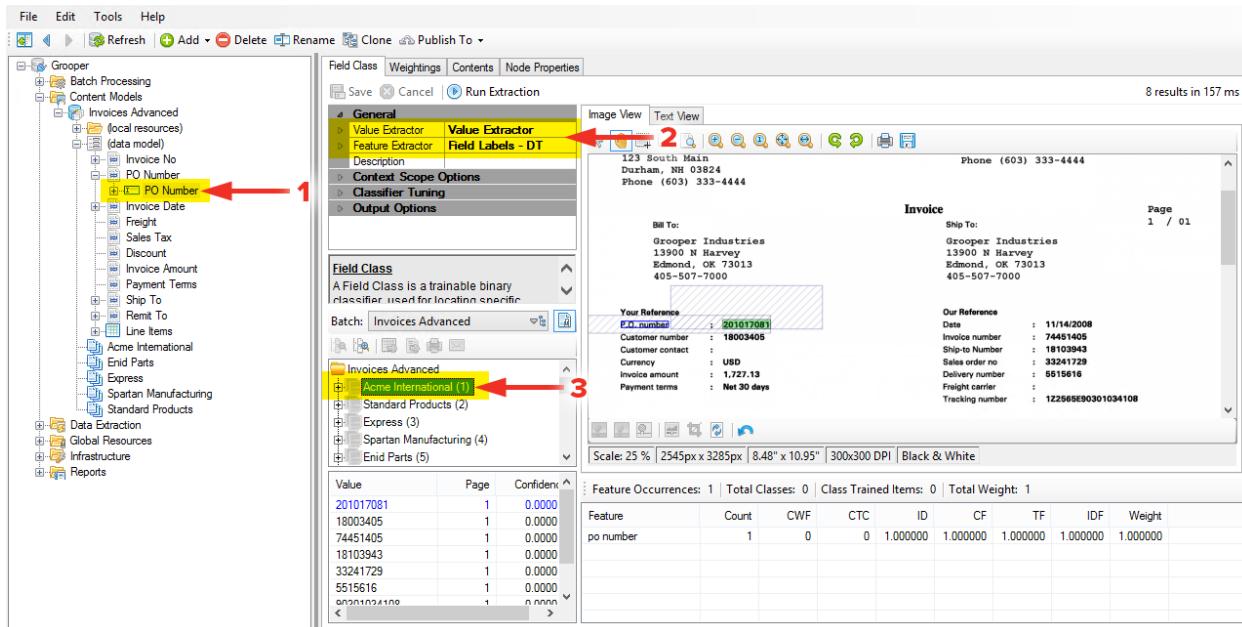
STEP 3 – SETTING THE EXCLUSION EXTRACTOR PROPERTY

This very generic pattern is getting far more results than needed. The **Exclusion Extractor Data Type** that was just built references **Data Types** of areas within this document set where **PO Numbers** will never appear: in addresses, and in line item details of the invoices. **(1)** Set the **Exclusion Extractor Type** to **Reference**, **(2)** and the **Referenced Extractor** to the **Exclusion Extractor Data Type** within the **Invoices Advanced • (local resources)** area. **Save and Run Extraction** and notice now that a bulk of unwanted results have been eliminated.



STEP 4 – SETTING THE FIELD CLASS EXTRACTORS

(1) Select the **PO Number** Field Class, **(2)** set the **Value Extractor Type** to **Reference**, and change the **Referenced Extractor** to the **Value Extractor Data Type** that is a child to the **PO Number** Field Class. Change the **Referenced Extractor** of the **Feature Extractor** section to the **Field Labels – DT** Data Type within the **Invoices Advanced • (local resources)** area. **(3)** Save and Run Extraction on the **Acme International (1)** document.



STEP 5 – TRAINING THE PO NUMBER FEATURE

(1) Select the 201017081 value in the Value List View and (2) notice again that translation is occurring for the feature from P.O. Number to po number. (3) Train this positively.

The screenshot shows the Grooper interface with the following components:

- Left Panel:** A tree view of the project structure under "Grooper".
- Center Panel:**
 - Field Class:** "Invoices Advanced" is selected.
 - Value List View:** Shows values for "PO Number" and "po number". The value "201017081" is highlighted with a yellow box and has a red arrow labeled "1" pointing to it.
 - Feature Occurrences Table:** Shows a row for "po number" with a yellow background. A red arrow labeled "2" points to the "Count" column (value 1).
 - Feature Statistics Table:** Shows a row for "po number" with a yellow background. A red arrow labeled "3" points to the "Weight" column (value 0.903090).
- Right Panel:** An "Image View" of an invoice document. A yellow box highlights the "Your Reference" section with the value "201017081". A red arrow labeled "2.1" points to this value.

STEP 6 – ADJUSTING CONTEXT ZONES

Checking extraction results against the **Standard Products (2)** document should reveal that (1) the Context Zones for this Field Class need to be adjusted due to Zone 2 picking up the unwanted features **Due Date** and **Amount Due**. (2) Use the following values for the Context Zones:

Zone 1 – Left: **-3.278** Top: **-0.054** Right: **0.312** Bottom: **0.054**

Zone 2 – Left: **-0.67** Top: **-0.37** Right: **0.5** Bottom: **0.04**

The screenshot shows the Grooper interface with the following components:

- Left Panel:** A tree view of the project structure under "Grooper".
- Center Panel:**
 - Field Class:** "Invoices Advanced" is selected.
 - Context Scope Options:** "Context Zones" is selected, highlighted with a yellow box and a red arrow labeled "1" pointing to it.
 - Context Zones Table:** Shows two zones: "Zone 1" and "Zone 2". The "Zone 1" row is highlighted with a yellow box and a red arrow labeled "2.1" pointing to its coordinates. The "Zone 2" row is also highlighted with a yellow box and a red arrow labeled "2.1" pointing to its coordinates.
 - Feature Occurrences Table:** Shows a row for "po number" with a yellow background. A red arrow labeled "2" points to the "Count" column (value 1).
- Right Panel:** An "Image View" of an invoice document titled "STANDARD ORIGINAL INVOICE". A yellow box highlights the "DUE DATE" and "AMOUNT DUE" fields. A red arrow labeled "2" points to these fields.

STEP 7 – SETTING MINIMUM CONFIDENCE

Because there is only one feature trained, and it is being returned at 100% confidence, (1) the Minimum Confidence can be set to anything above 0% (2) and all unwanted results will be trimmed.

STEP 8 – SETTING PROPERTIES FOR THE INVOICE NO DATA FIELD

(1) Select the PO Number Data Field and (2) set its Required property to True. (3) Set the Default Extractor Type to Reference and set the Referenced Extractor to the PO Number Field Class that is a child object of the PO Number Data Field.

THE LAST FIELD CLASS

Finishing off the usage of the **Field Class** for this **Data Model** will end with the **Invoice Amount**. This **Field Class** will not introduce any new concepts, but is needed to complete extraction. Sometimes it is best, however, to realize you don't have to re-invent the wheel every time and just use tools that are available and achieve quick results.

STEP 1 – ADDING THE INVOICE AMOUNT FIELD CLASS AND CHOOSE EXTRACTORS

- (1) Add a **Field Class** as a child object to the **Invoice Amount Data Field** and name it **Invoice Amount**.
- (2) Set the **Value Extractor Type** to **Reference** and the **Referenced Extractor** to the **Currency (OCR)** Data Type within the **Data Extraction • Data Types > (system)** area. Leave the **Feature Extractor Type** as **Reference** but change the **Referenced Extractor** to the **Field Labels – DT** Data Type within the **Invoices Advanced • (local resources)** area.
- (3) Save and **Run Extraction** on the **Acme International (1)** document.

The screenshot shows the Grooper ACE application interface. On the left, the navigation tree includes sections like Grooper, Batch Processing, Content Models, Invoices Advanced (with local resources), and various business units (Acme International, Enid Parts, Express, Spartan Manufacturing). The 'Invoices Advanced' section is expanded, showing fields such as Invoice No, PO Number, Invoice Date, Freight, Sales Tax, Discount, and Invoice Amount. The 'Invoice Amount' field is selected and highlighted with a yellow box and a red arrow pointing to it from the list of steps.

The main workspace shows the 'Field Class' configuration dialog for the 'Invoice Amount' field. The 'General' tab is selected, showing the 'Value Extractor' set to 'Currency (OCR)' and the 'Feature Extractor' set to 'Field Labels - DT'. A red arrow labeled '3' points to the 'Run Extraction' button at the top right of this dialog.

Below the configuration dialog, there is a preview area showing an invoice document for 'Acme International (1)'. A red arrow labeled '2' points to the 'Image View' tab. The preview shows the invoice details, including the bill to and ship to addresses, reference numbers, and payment terms. A red arrow labeled '3.1' points to the 'Acme International (1)' document itself.

At the bottom of the interface, there is a feature occurrence table:

Value	Page	Confidence
1,727.13	1	0.0000 %
984.53	1	0.0000 %
1,969.06	1	0.0000 %
12.50	1	0.0000 %
246.13	1	0.0000 %
861.47	1	0.0000 %
1,722.93	1	0.0000 %
n.nn	1	0.0000 %

STEP 2 – TRAIN THE INVOICE AMOUNT FEATURE

(1) Select the 1,727.13 value from the Values List View and (2) train its feature invoice amount.

The screenshot shows the Grooper ACE interface with the following components:

- Left Sidebar:** Shows the project structure under "Grooper" and "Content Models".
- Top Bar:** Includes File, Edit, Tools, Help, Refresh, Add, Delete, Rename, Clone, Publish To, and Run Extraction buttons.
- Field Class Tab:** Set to "Value Extractor" (Currency (OCR)).
- General Tab:** Contains sections for Value Extractor, Feature Extractor, Context Scope Options, Classifier Tuning, and Output Options.
- Image View and Text View:** Displays an invoice document with various fields like Bill To, Ship To, and Payment terms.
- Values List View:** A table showing values for "Invoice Amount" with a yellow highlight on "1,727.13". A red arrow labeled "1" points to this row.
- Feature Training Table:** Shows the feature "invoice amount" with a count of 1, CWF of 1, and confidence of 1.000000. A red arrow labeled "2.1" points to this row.
- Feature Occurrences Table:** Shows the total classes as 11, trained items as 0, and total weight as 1.04139268515822.

STEP 3 – ADJUST CONTEXT ZONES

To prevent currency values that are near the trained feature from getting any confidence, (1) the Context Zones must be adjusted. (2) Use the following values for the Context Zones:

Zone 1 – Left: **-4** Top: **-0.05** Right: **0.2** Bottom: **0.1**

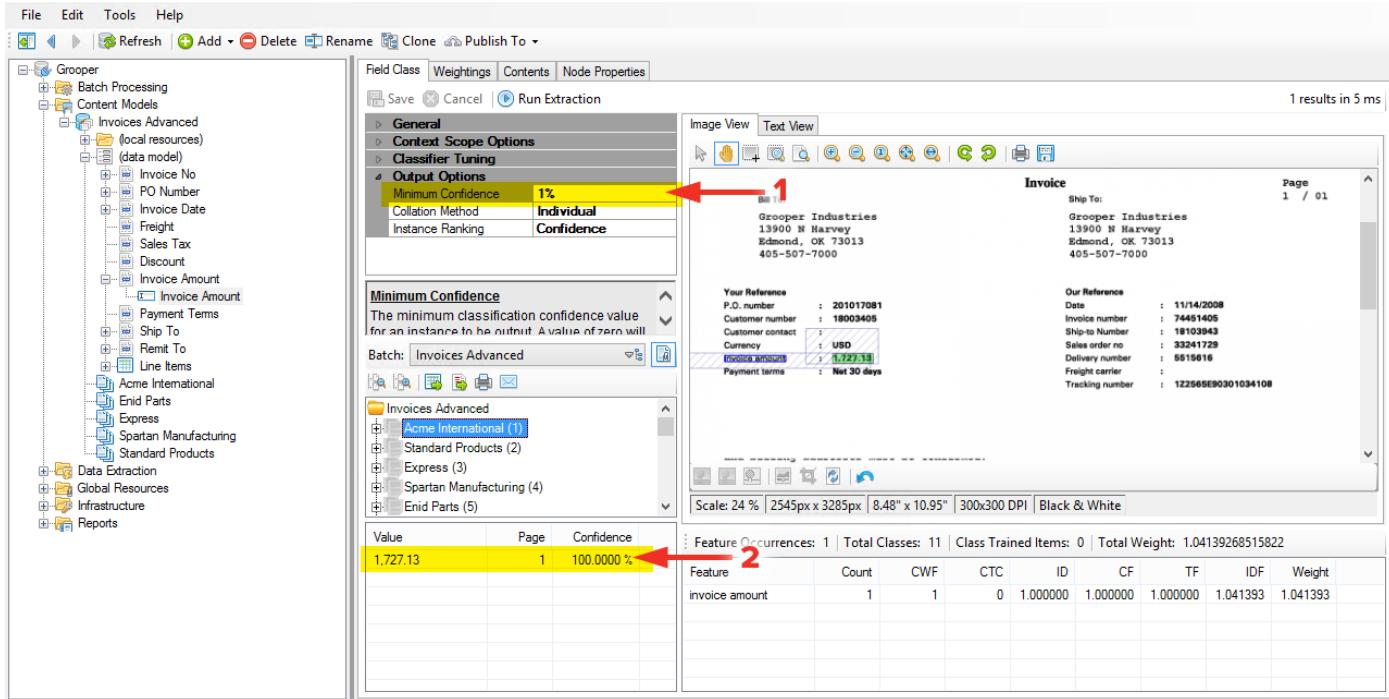
Zone 2 – Left: **-0.58** Top: **-0.36** Right: **0.3** Bottom: **0.05**

The screenshot shows the Grooper ACE interface with the following components:

- Left Sidebar:** Shows the project structure under "Grooper" and "Content Models".
- Top Bar:** Includes File, Edit, Tools, Help, Refresh, Add, Delete, Rename, Clone, Publish To, and Run Extraction buttons.
- Field Class Tab:** Set to "Value Extractor" (Currency (OCR)).
- General Tab:** Contains sections for Value Extractor, Feature Extractor, Context Scope Options, Classifier Tuning, and Output Options. A red arrow labeled "1" points to the "Context Zones" section.
- Image View and Text View:** Displays an invoice document with various fields like Sales Tax, Amount, and Total Due.
- Context Zones Dialog:** Shows two zones: Zone 1 with coordinates (-4, -0.05) to (0.2, 0.1) and Zone 2 with coordinates (-0.58, -0.36) to (0.3, 0.05). A red arrow labeled "2" points to the "Left" coordinate in Zone 1.
- Values List View:** A table showing values for "Invoice Amount" with a yellow highlight on "333.58".
- Feature Training Table:** Shows the feature "invoice amount" with a count of 1, CWF of 1, and confidence of 1.000000.
- Feature Occurrences Table:** Shows the total classes as 11, trained items as 0, and total weight as 1.04139268515822.

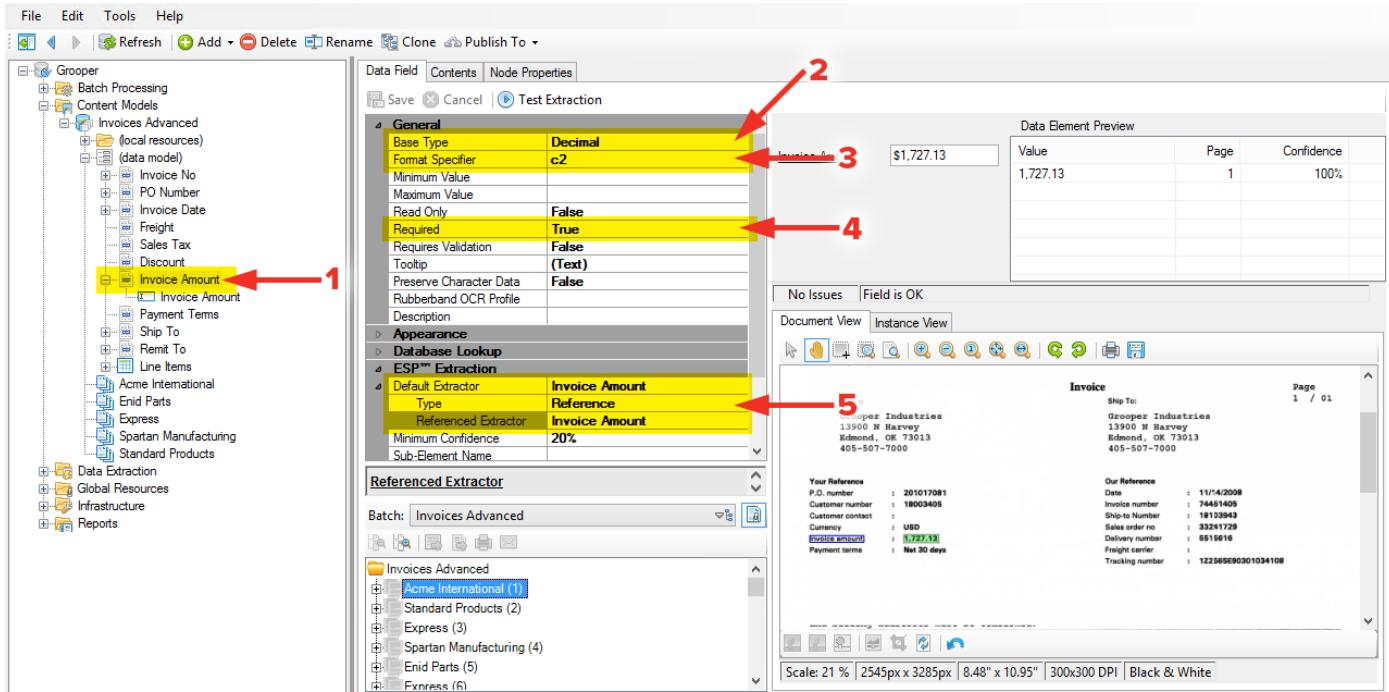
STEP 4 – SETTING MINIMUM CONFIDENCE

Because there is only one feature trained, and it is being returned at 100% confidence, (1) the Minimum Confidence can be set to anything (2) above 0% and all unwanted results will be trimmed.



STEP 5 – SETTING PROPERTIES FOR THE INVOICE NO DATA FIELD

(1) Select the **Invoice Amount Data Field**, (2) set the **Base Type** to **Decimal**, and (3) the **Format Specifier** to **c2**. (4) Set its **Required** property to **True**. (5) Finally, Set the **Default Extractor Type** to **Reference** and set the **Referenced Extractor** to the **Invoice Amount Field Class** that is a child object of the **Invoice Amount Data Field**.



THE REMAINING DATA FIELDS

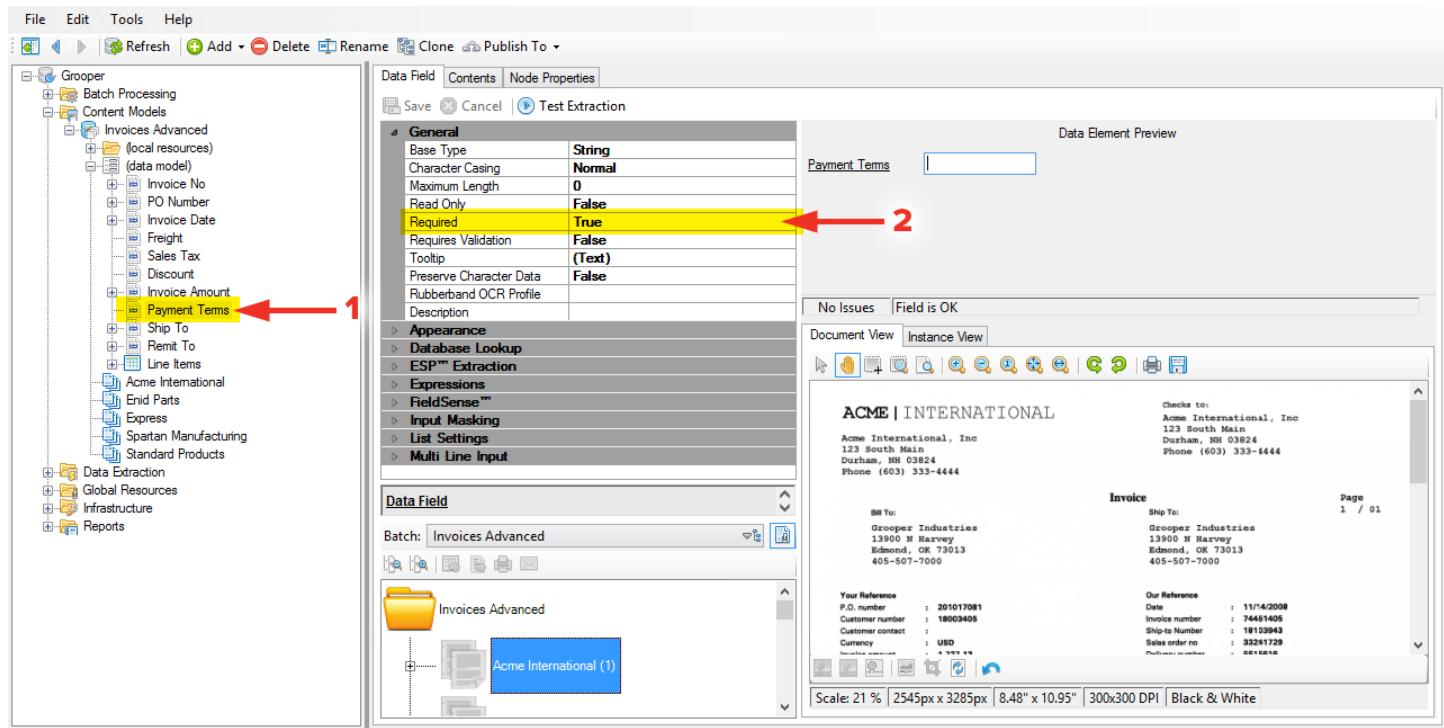
The last few **Data Fields** of this **Data Model** will not leverage a **Field Class**, but **Data Types** instead. The last **Data Field** that will be required is **Payment Terms**; the rest are not always present on the documents and don't require returned results. Because they are dollar amounts they can, however, be returned as **\$0.00**, instead of being left blank. Finally, the concept of per-**DocType** overrides called **Data Element Profiles** will be introduced.

THE FINAL REQUIRED FIELD – PAYMENT TERMS

This information is present on all documents and desired as an extracted result. As such, it will be required, but the last of its type. Its setup will be very simple, and will not even reference an extractor.

STEP 1 – SET REQUIRED PROPERTY

- (1) Select the **Payment Terms Data Field** and, as mentioned, (2) set its **Required Property** to **True**.

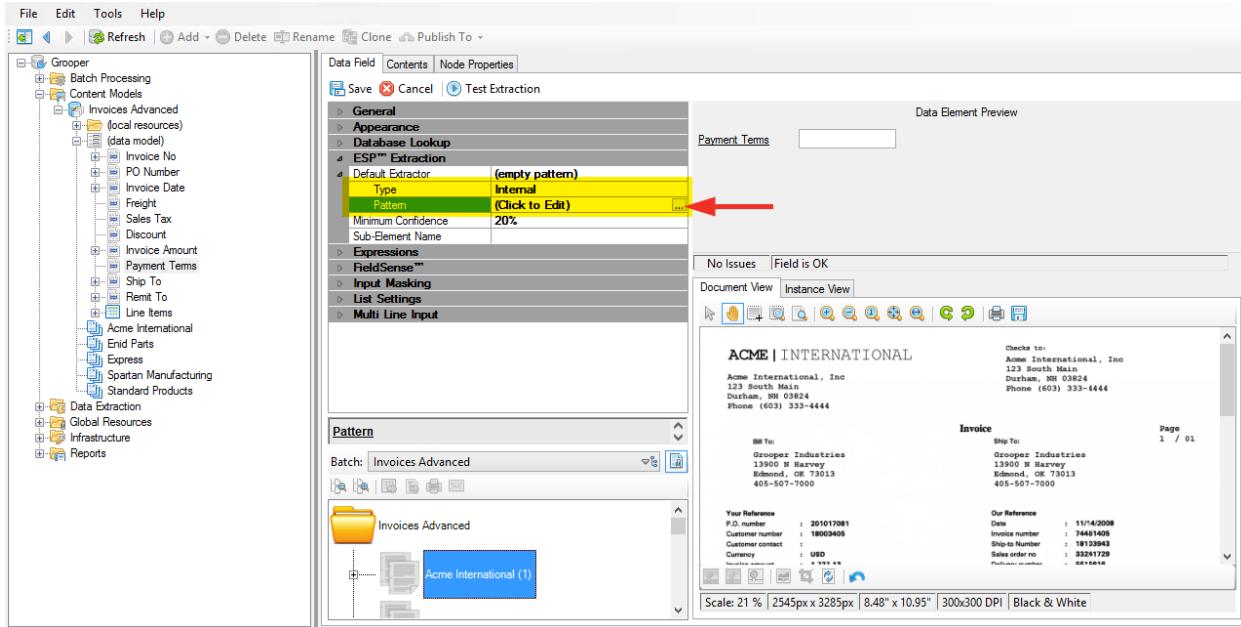


The screenshot shows the Grooper ACE interface with the following details:

- Left Panel (File Structure):** Shows the project structure under "Grooper". The "Invoices Advanced" folder is expanded, showing sub-items like "Invoice No", "PO Number", "Invoice Date", "Freight", "Sales Tax", "Discount", "Invoice Amount", and "Payment Terms". A red arrow labeled "1" points to the "Payment Terms" item.
- Middle Panel (Data Field Configuration):** A modal window titled "Data Field" is open for the "Payment Terms" field. It has tabs for "Data Field", "Contents", and "Node Properties". The "General" section is selected, showing properties like "Base Type" (String), "Character Casing" (Normal), "Maximum Length" (0), "Read Only" (False), and "Required" (True). A red arrow labeled "2" points to the "Required" field. Other sections include "Appearance", "Database Lookup", "ESP™ Extraction", "Expressions", "FieldSense™", "Input Masking", "List Settings", and "Multi Line Input".
- Right Panel (Data Element Preview):** Shows a preview of an invoice document for "ACME | INTERNATIONAL". The document includes fields for "Bill To" (Grooper Industries, Edmond, OK) and "Ship To" (Grooper Industries, Edmond, OK). Below the document, there is a table of "Your Reference" and "Our Reference" fields, along with other document metadata like "Scale: 21%" and "Black & White".

STEP 2 – INTERNAL PATTERN ON DATA FIELD

To this point all **Data Fields** have referenced an extractor, but patterns can also be written directly to **Data Fields**. In the **ESP Extraction** section, set the **Default Extractor Type** to **Internal**. Click the ellipsis button on the **Pattern** property to bring up a **Pattern Editor** window.



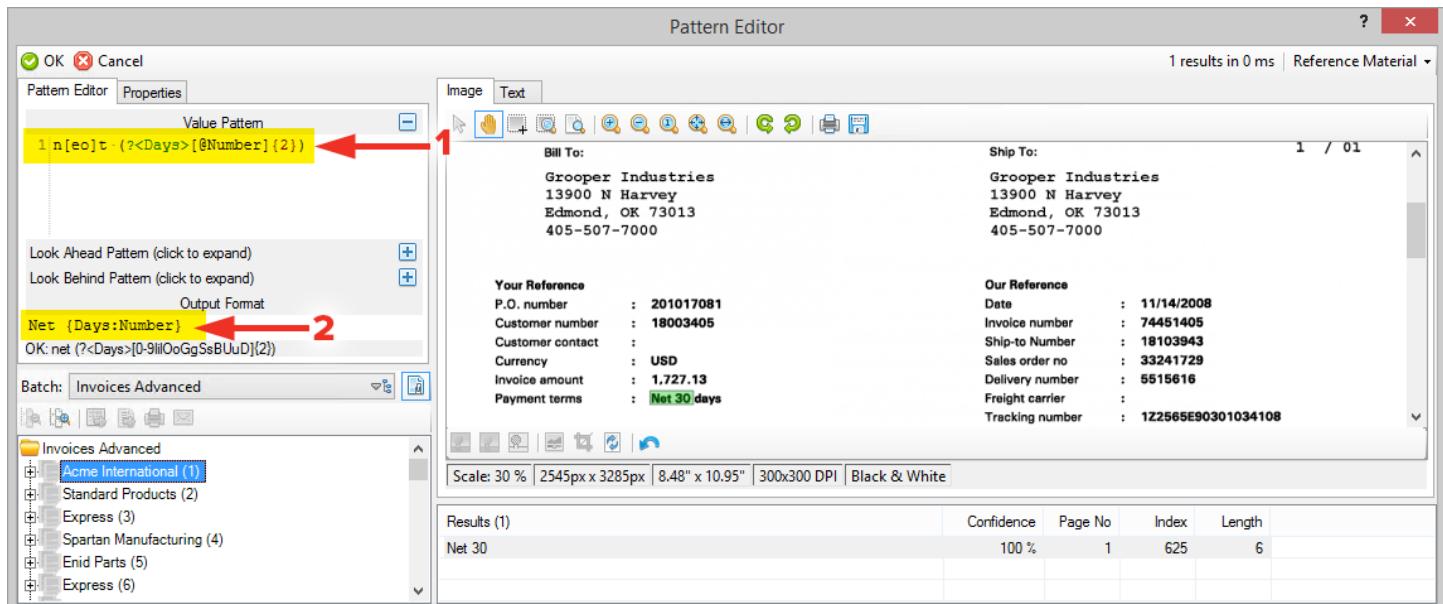
STEP 3 – VALUE PATTERN AND OUTPUT FORMAT

(1) For the **Value Pattern** use the following:

n[eo]t (?<Days>[@Number]{2})

(2) For the **Output Format** use the following:

Net {Days:Number}



STEP 4 – SAVE AND RUN EXTRACTION

Click OK to close the Pattern Editor window, then **(1) Save** and **Test Extraction** to **(2) see results**.

The screenshot shows the Grooper ACE software interface. On the left, there is a navigation tree with categories like Grooper, Batch Processing, Content Models, Invoices Advanced, and Reports. The 'Invoices Advanced' node is expanded, showing sub-items such as Invoice No, PO Number, Invoice Date, Freight, Sales Tax, Discount, Invoice Amount, Payment Terms, Ship To, Remit To, and Line Items. Under Line Items, 'Acme International' is selected.

The main workspace contains two windows:

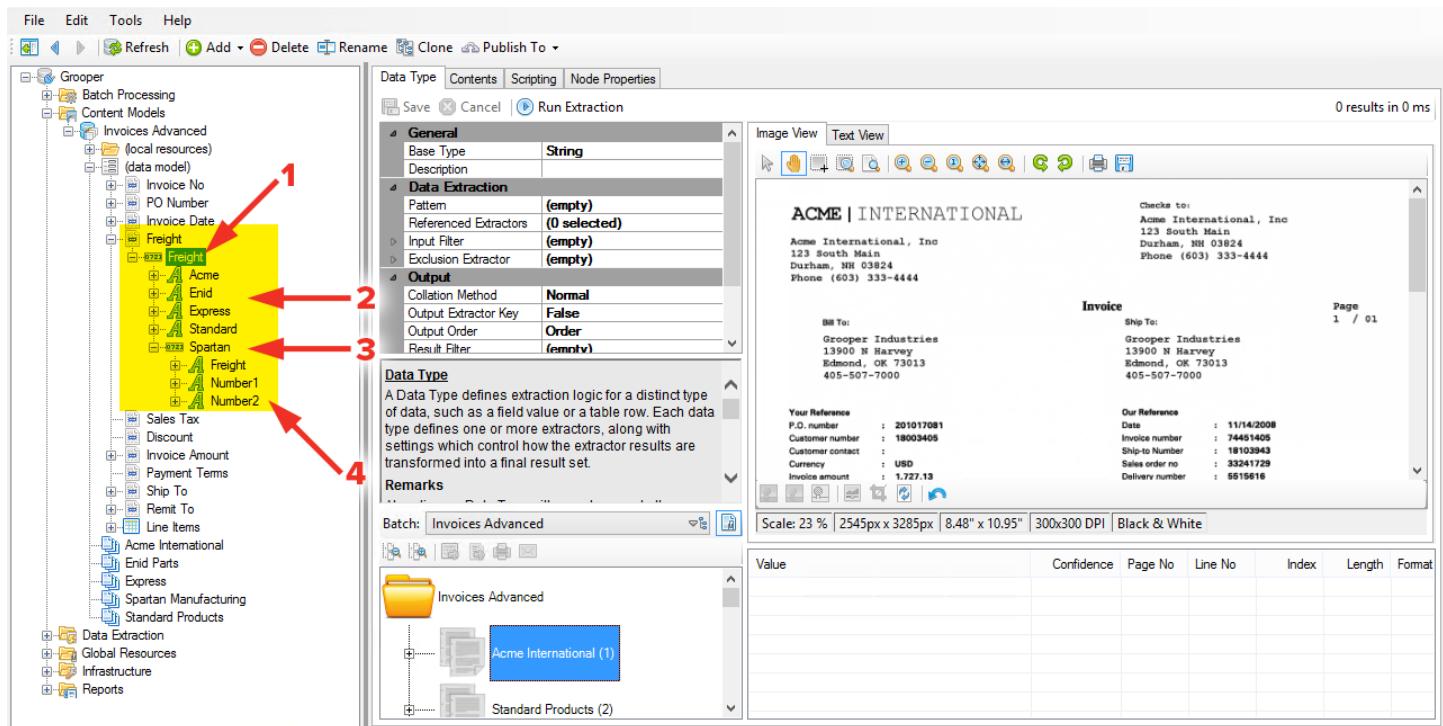
- Pattern Editor Window (Top Left):** This window has tabs for Data Field, Contents, and Node Properties. The 'Save' button is highlighted with a yellow background and a red arrow labeled '1'. Below it is a 'Test Extraction' button. The 'Default Extractor' section shows the pattern: "net (?<Days>{@Number}{2})".
- Data Element Preview Window (Top Right):** This window displays a table titled 'Data Element Preview' with one row: 'Payment Terms' and 'Net 30'. A red arrow labeled '2' points from the 'Test Extraction' button in the Pattern Editor to this preview table.
- Document View Window (Bottom Right):** This window shows a preview of an invoice document. It includes sections for 'Bill To' (Grooper Industries, 13900 N Harvey, Edmond, OK 73013, 405-507-7000), 'Your Reference' (P.O. number: 201017081, Customer number: 18003405, Customer contact: , Currency: USD, Invoice amount: 1,727.13, Payment terms: Net 30 days), and 'Invoice' (Ship To: Grooper Industries, 13900 N Harvey, Edmond, OK 73013, 405-507-7000). It also shows 'Invoice details' and other document metadata.

SETTING UP THE FREIGHT DATA FIELD

The extractor built for this field is somewhat complex, but is all based on concepts that have been seen already, except for an **Ordered Array** that returns a summed result.

STEP 1 – ADDING ALL OF THE PARTS

(1) Add a **Data Type** named **Freight** as a child object to the **Freight Data Field**. (2) Add four child **Data Formats** and name them **Acme**, **Enid**, **Express**, and **Standard** respectively. (3) Add a **Data Type** named **Spartan** as a child object to the **Freight Data Type**. (4) Add three **Data Formats** to this **Data Type** and name them **Freight**, **Number1**, and **Number2**.



STEP 2 – SETTING UP THE ACME DATA FORMAT

(1) Make sure tab marking is enabled, and for the **Value Pattern** enter the following:

```
(?<Freight>[@Number.] {3,12})
```

(2) For the **Look Ahead Pattern** enter the following:

```
WS.FREIGHT[0o]231[^n]+n
```

```
[^n]+n
```

```
[^t]+t
```

```
[^t]+t
```

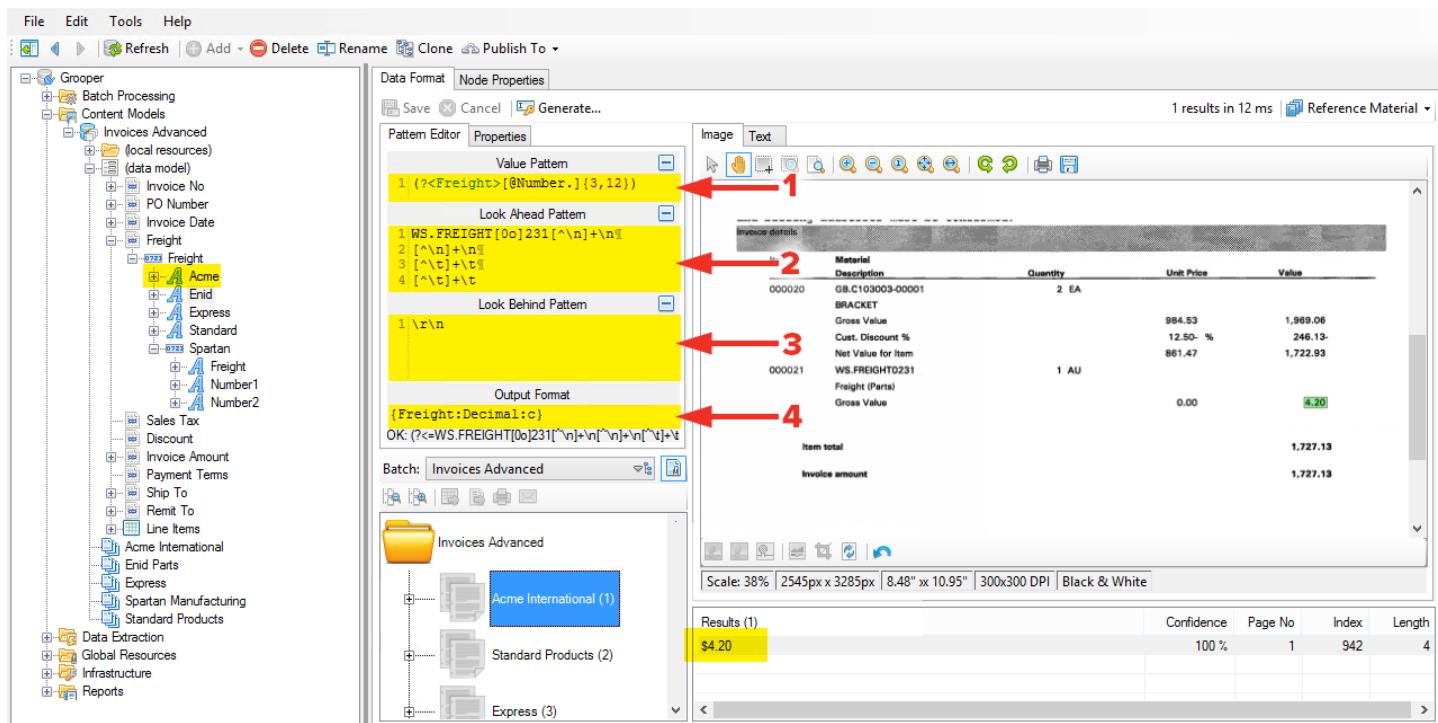
(3) For the **Look Behind Pattern** enter the following:

```
\r\n
```

(4) For the **Output Format** enter the following:

```
{Freight:Decimal:c}
```

This **Output Format** will not only return the named group **Freight**, and force the results from the **@Number Expression Variable** to return decimal results, but will return it as a currency amount due to the format specifier **c**. Finally, enable **Tab Marking** from the **Properties** tab within the **Text Preprocessing** properties.



STEP 3 – SETTING UP THE ENID DATA FORMAT

- (1) For the **Enid Data Format Value Pattern**, enter the following:
- (2) For the **Look Ahead Pattern** enter the following:
- (3) For the **Look Behind Pattern** enter the following:

Use [Enid Parts \(10\)](#) to see a result.

[@Number.] {3,12}
020-0027[^\\n]*?\\n[^\\r]*?
\\r

STEP 4 – SETTING UP THE EXPRESS DATA FORMAT

- (1) For the **Express Data Format Value Pattern**, enter:
 - (2) For the **Look Ahead Pattern** enter the following:
- Use [Express Manufacturing \(6\)](#) to get a result.

[@Number.] {3,12}
shipping charge:?:[^0-9]{0,4}

STEP 5 – SETTING UP THE STANDARD DATA FORMAT

- (1) For the Standard Data Format Value Pattern, enter the following:
- (2) For the Look Ahead Pattern enter the following:

[@Number.] {3,12}
shipping charge\s

STEP 6 – SETTING UP THE SPARTAN DATA FORMATS

- (1) For the Freight Data Format enable Tab Marking and use a tab for the Look Ahead Pattern and the Look Behind Pattern; and for the Value Pattern simply use the string: **freight**.
- (2) The Number1 and Number2 Data Formats are the same, so enable Tab Marking for them, use a tab for the Look Ahead Pattern and Look Behind Pattern, and for the Value Pattern use: **\d.+**

STEP 7 – OUTPUT OPTIONS FOR THE SPARTAN DATA TYPE

- (1) Select the Spartan Data Type and (2) change the Collation Method to OrderedArray. (3) Set the Combine Method as Sum and (4) change the Array Layout to Vertical only. (5) Finally, set a Maximum Vertical Distance of 0.25. (6) Notice the result of the two 9.10 values being added up to 18.20.

The screenshot shows the Grooper ACE interface with the following details:

- Left Panel:** Shows a tree view of the project structure under "Grooper". A red arrow labeled 1 points to the "Spartan" node under "Freight".
- Central Panel:**
 - Data Type Tab:** Shows the "Spartan" node selected. Red arrows labeled 2, 3, 4, 5, and 6 point to the following settings:
 - Collation Method: OrderedArray
 - Combine Method: Sum
 - Array Options: Vertical Array
 - Maximum Vertical Distance: 0.25
 - Result View: Image View (highlighted)
 - Results View:** Displays a table with columns: MODE, TOTAL, FREIGHT, OTHER CHARGES, RESTOCKING, PCT, SALES TAX, and AMOUNT. The FREIGHT column shows two rows: IN (9.10) and OUT (9.10). The TOTAL column shows 315.38. The AMOUNT column shows 18.20. A red arrow labeled 6.1 points to the value 18.20 in the AMOUNT column.
 - Bottom Panel:** Shows the "Results (1)" table with one row containing the value 18.20.

STEP 8 – SETTING THE PROPERTIES OF THE FREIGHT DATA FIELD

- (1) Select the Freight Data Field, (2) set its Base Type to Decimal, and (3) its Format Specifier to c2. (4) Set the Default Extractor Type to Reference and the Referenced Extractor to the Freight Data Type that is the child of the Freight Data Field. (5) Finally, in the Expressions section, set the Default Value Expression to 0. Since this value isn't required, when a result isn't returned, it will at least present \$0.00.

The screenshot shows the Grooper ACE interface with the following details:

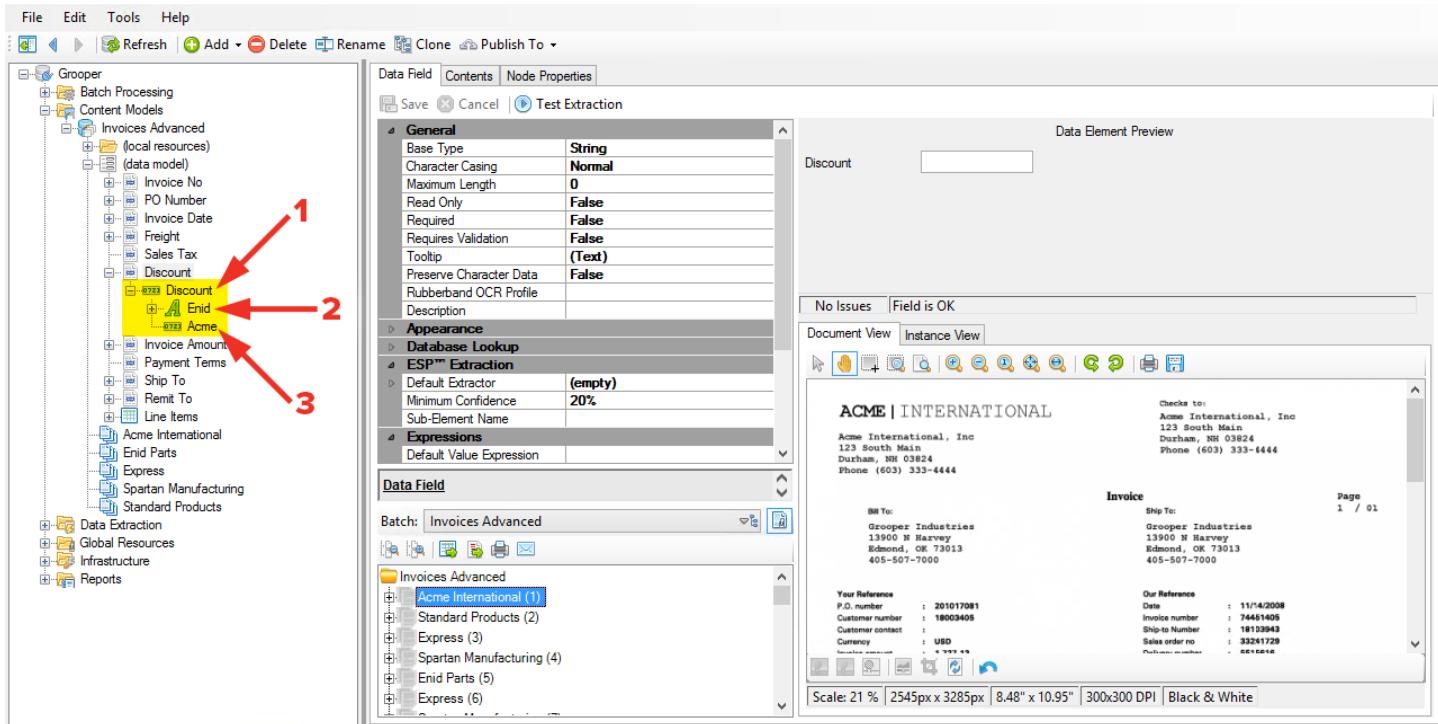
- Left Panel:** Shows a tree view of the project structure under "Grooper". A red arrow labeled 1 points to the "Freight" node under "Invoices Advanced".
- Central Panel:**
 - Data Field Tab:** Shows the "Freight" node selected. Red arrows labeled 2, 3, 4, and 5 point to the following settings:
 - General: Base Type: Decimal, Format Specifier: c2
 - ESP™ Extraction: Default Extractor: Freight, Minimum Confidence: 20%
 - Expressions: Default Value Expression: 0
 - Right Panel:** Shows a "Data Element Preview" table with one row: Value (\$4.20), Page (1), and Confidence (100%). Below it is a "Document View" pane showing a table of items with their descriptions, quantities, unit prices, and values. A red arrow labeled 5 points to the "Default Value Expression" setting in the Data Field tab.

SETTING UP THE DISCOUNT DATA FIELD

This will be another **Data Type** setup with a combination **Data Format/Data Type** sibling situation, but a bit simpler than the previous example. The **Enid Parts** and **Acme International** documents are the only ones that have discounts, so this extractor will focus on them.

STEP 1 – ADDING ALL OF THE PARTS

- (1) Add a **Data Type** as a child object to the **Discount Data Field** and name it **Discount**. (2) Add a **Data Format** to this **Data Type** and name it **Enid**. (3) Also, as a child object to the **Discount Data Type**, add another **Data Type** and name it **Acme**.



STEP 2 – SETTING UP THE ENID DATA FORMAT

- (1) Select the **Enid Data Format**, and for the Value Pattern use:

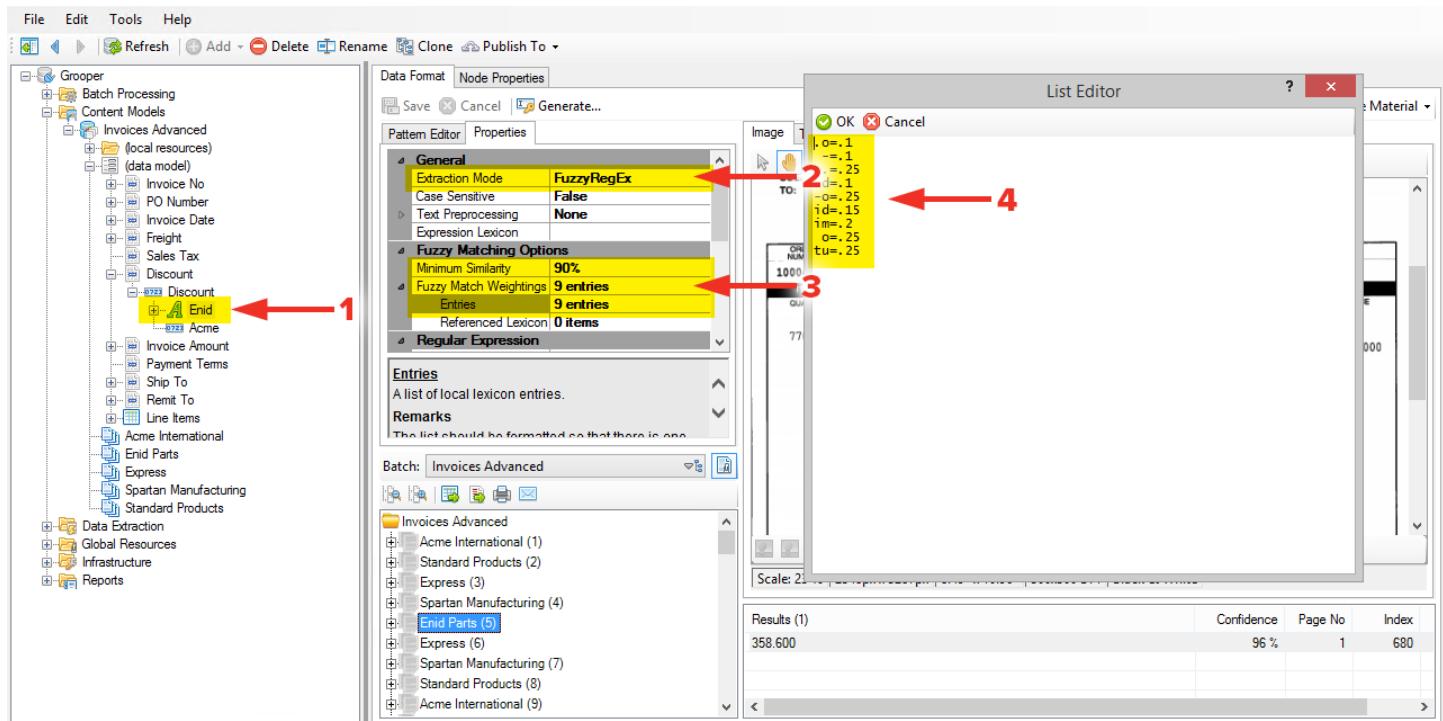
(?<Discount>\d{1,6}\.\d{2,3})

For the **Look Ahead Pattern** use:

order discount amount -

- (2) On the **Properties** tab, set the **Extraction Mode** to **FuzzyRegEx** (3) and leave the **Minimum Similarity** at **90%**. Select the **Entries** property and use the ellipsis button to bring up the **List Editor** window, and within it, (4) enter the following (the spaces at the beginning of the first three entries won't properly copy from a PDF):

```
.=.25
--.1
o=.25
.o=.1
-o=.25
1d=.1
id=.15
im=.2
tu=.25
```



The string in the **Look Ahead Pattern** falls in a bad place on the **Enid** documents due to the lines of the form. These lines throw off **OCR** pretty badly, and cause a lot of bad characters to be read. The settings used above help compensate for this issue.

STEP 3 – SETTING UP THE ACME DATA TYPE

Select the Acme Data Type and for the **Value Pattern** use the following:

[@Number.,1{3,12}]

For the **Look Ahead Pattern** use the following:

cust\\.\\\$discount\\s%(\t|\\r\\n)[^\\t]+\\t

For the **Output Format** use the following:

{0:Number}

Make sure **Tab Marking** is enabled in the **Text Preprocessing** area of the **General** section on the **Properties** tab.

(1) In the **Output** section set the **Collation Method** to **Combine** (2) and the **Combine Method** to **Sum**.

(3) Notice on **Acme International (14)** that this setup will find two discount amounts and add them together.

The screenshot shows the Grooper ACE software interface with the following details:

- File Bar:** File, Edit, Tools, Help.
- Toolbar:** Refresh, Add, Delete, Rename, Clone, Publish To.
- Left Sidebar:** Shows a tree view of projects and resources, including "Grooper", "Batch Processing", "Content Models", "Invoices Advanced", "Invoice No", "PO Number", "Invoice Date", "Freight", "Sales Tax", "Discount", "Enid", "Acme", "Invoice Amount", "Payment Terms", "Ship To", "Remit To", "Line Items", "Acme International", "Enid Parts", "Express", "Spartan Manufacturing", "Standard Products", "Data Extraction", "Global Resources", "Infrastructure", and "Reports".
- Central Panel - Data Type Properties:**
 - Data Type:** Acme
 - General Tab:**
 - Pattern:** RegEx: [@Number.,1{3,12}]
 - Referenced Extractors:** (0 selected)
 - Input Filter:** (empty)
 - Exclusion Extractor:** (empty)
 - Output Tab:**
 - Collation Method:** Combine (highlighted by red arrow 1)
 - Group By:** None
 - Enforce Page Boundaries:** False
 - Combine Method:** Sum (highlighted by red arrow 2)
 - Output Order:** Order
 - Result Filter:** (empty)
 - Deduplication:** (disabled)
 - Image View:** Shows a table of invoice items with columns: Material Description, Quantity, Unit Price, and Value. It highlights two rows with discounts:
 - Row 1: Item 000010, VALVE-AIR, 24 EA, 27.18, 652.32. A red arrow points from the "Value" column to the value 652.32.
 - Row 2: Item 000020, VALVE-AIR, 12 EA, 23.78, 281.76. A red arrow points from the "Value" column to the value 281.76.
 - Text View:** Shows the total value for each item: 3.2 (highlighted by red arrow 3.2).
 - Results (1):** Shows a table with one row containing the value 123.78. A red arrow points from the "Value" column to the value 123.78.
 - Bottom Status:** Scale: 23 % | 2539px x 3280px | 8.46" x 10.93" | 300x300 DPI | Black & White.

STEP 4 – SETTING UP THE PROPERTIES FOR THE DISCOUNT DATA FIELD

- (1) Select the Discount Data Field and set the Base Type to Decimal (2) and the Format Specifier to c2.
- (3) Set the Default Extractor Type to Reference and set the Referenced Extractor to the Discount Data Type that is a child of the Discount Data Field. (4) Set the Default Value Expression to 0.

The screenshot shows the Grooper ACE software interface for managing data fields. On the left, a tree view of content models like 'Invoices Advanced' and 'Standard Products' is visible. The central area is a configuration window for a 'Discount' data field. The 'General' tab is active, with 'Base Type' set to 'Decimal' (1) and 'Format Specifier' set to 'c2' (2). The 'ESP™ Extraction' tab is also active, showing 'Default Extractor' set to 'Discount' (3). The 'Expressions' tab is active, showing 'Default Value Expression' set to '0' (4). To the right, there's a 'Data Element Preview' window showing a single row with 'Value' \$123.78, 'Page' 1, and 'Confidence' 100%. Below it is a 'Data Element View' window showing a table of invoice details with columns for Item, Material Description, Quantity, Unit Price, and Value.

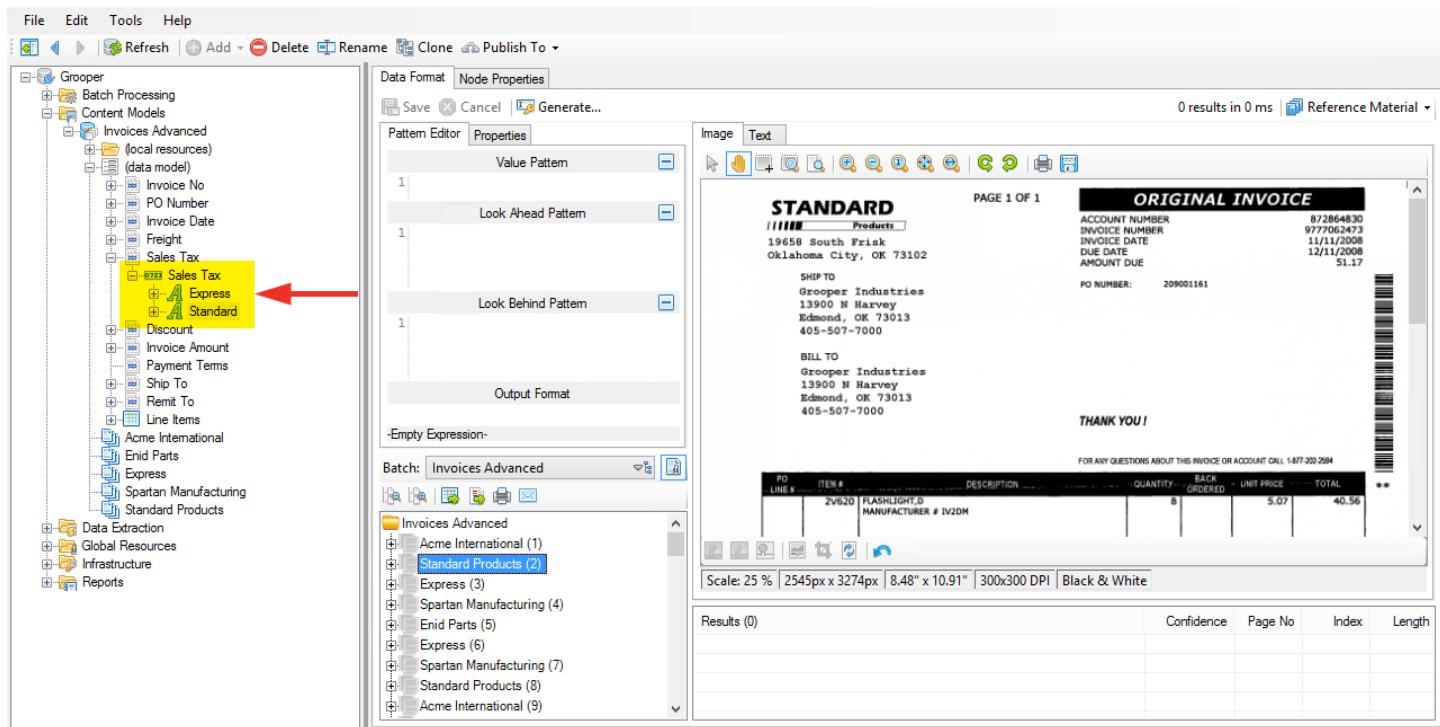
Item	Material Description	Quantity	Unit Price	Value
000010	GB.C136177-00001 VALVE-AIR	24 EA	27.18	652.32
	Gross Value		Cust. Discount %	81.54
000020	GB.C136177-00002 VALVE-AIR	12 EA	23.78	670.76
	Gross Value		Net Value for Item	337.82
000021	WS-FRIGHTO231 Freight (Parts)	1 AU	24.64	42.24
	Gross Value		Net Value for Item	295.68

SETTING UP THE SALES TAX DATA FIELD – DATA ELEMENT PROFILE OVERRIDES

The final field of information to setup within the **Data Model** is **Sales Tax**. A **Data Type** will be used to extract information for the **Express** and **Standard** documents as they contain **Sales Tax** information. **Enid Parts**, however, will be best suited to use a **Field Class** as its extractor, and because a **Data Field** can only have one **Default Extractor** it will have to be overwritten for that specific **DocType**. This will be accomplished by a profile being added to the **Enid Parts Data Element Profies**.

STEP 1 – ADDING ALL OF THE PARTS

Add a child **Data Type** to the **Sales Tax Data Field** and name it **Sales Tax**. Add two **Data Formats** as child objects to the **Sales Tax Data Type** and name them **Express** and **Standard**.

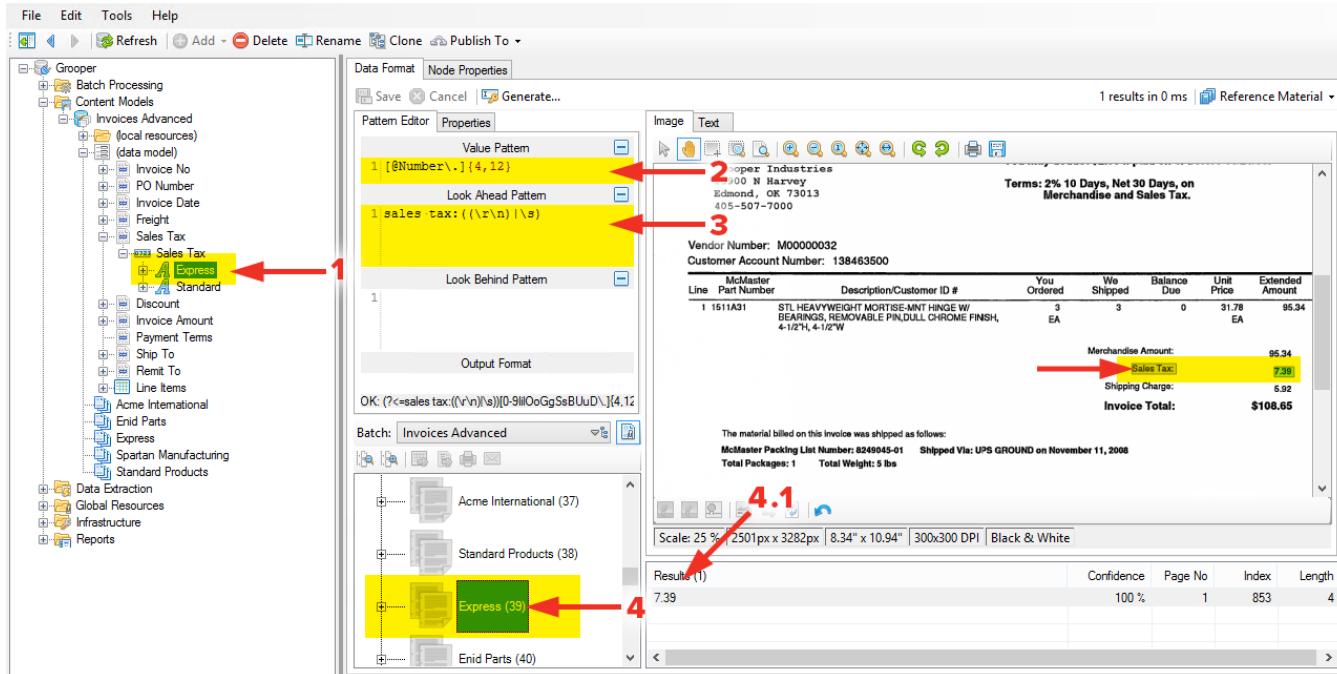


STEP 2 – SETTING UP THE EXPRESS DATA FORMAT

- (1) Select the Express Data Format and (2) for the Value Pattern use:
 (3) For the Look Ahead Pattern use the following:

`[@Number.] {4,12}`
`sales tax:((\r\n)|\s)`

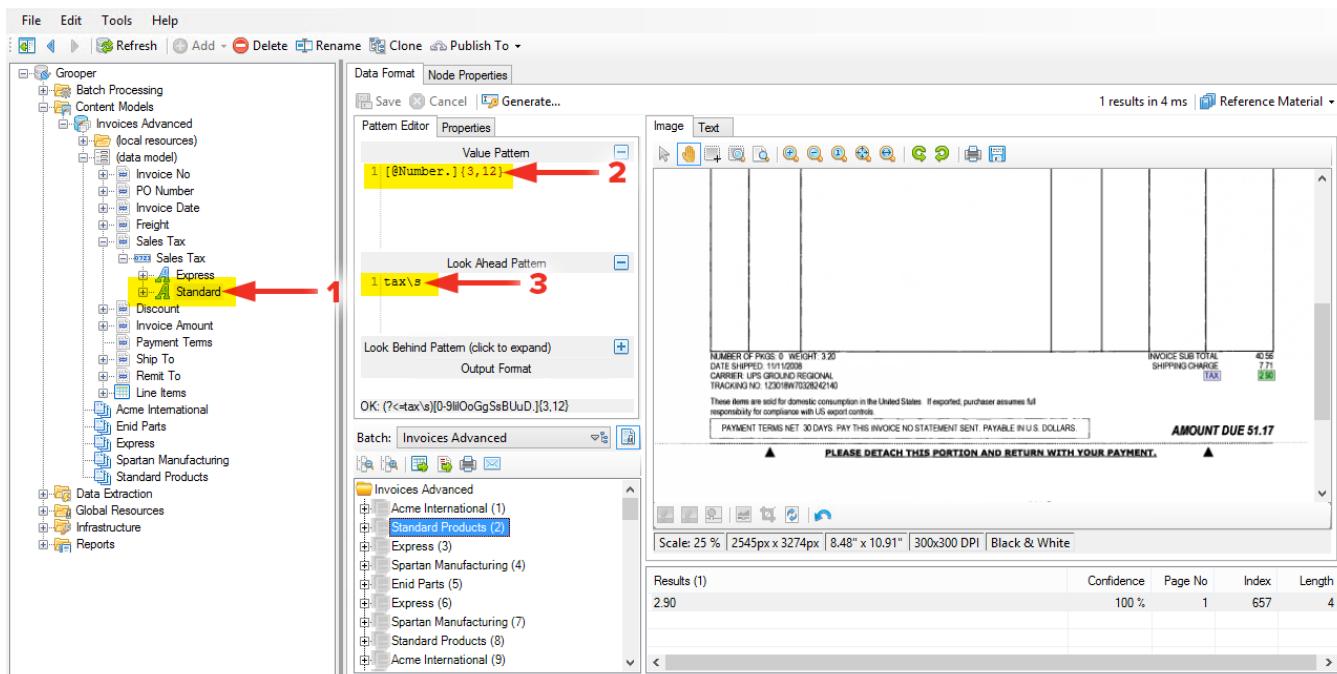
Sales Tax isn't a frequent occurrence, so don't be surprised if you don't (4) see a result until about Express (39).



STEP 3 – SETTING UP THE STANDARD DATA FORMAT

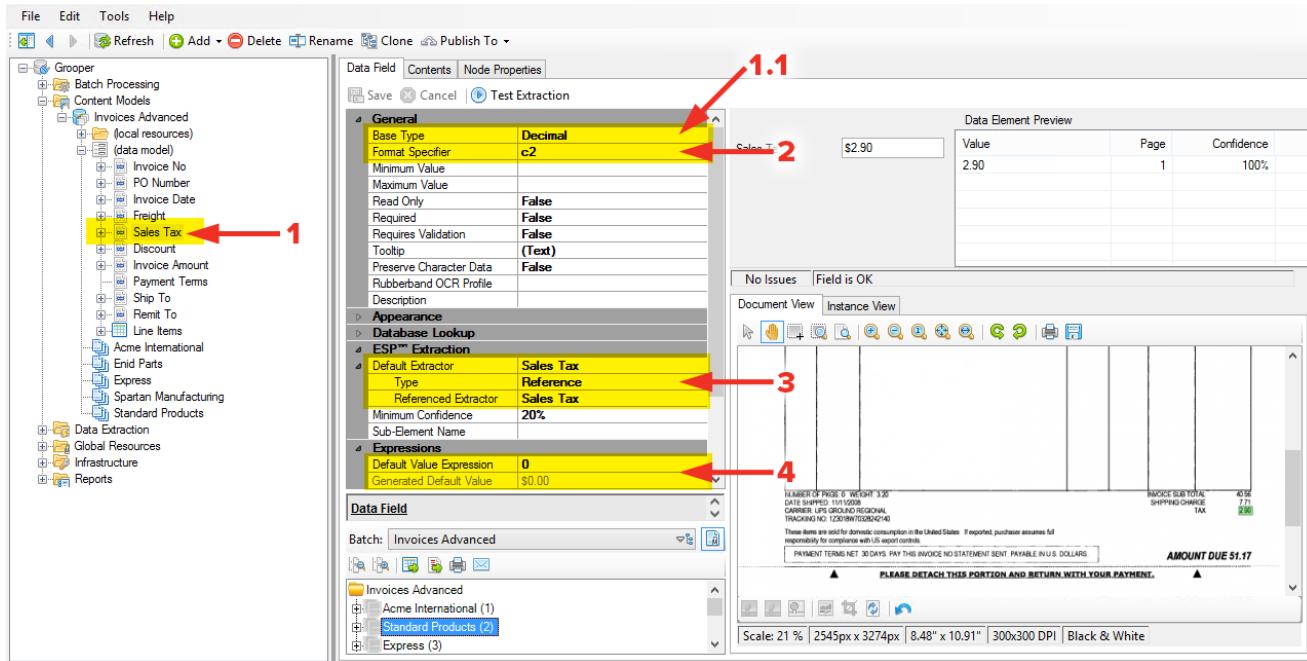
- (1) Select the Standard Data Format and (2) for the Value Pattern use:
 (3) For the Look Ahead Pattern use the following:

`[@Number.] {3,12}`
`tax\s`



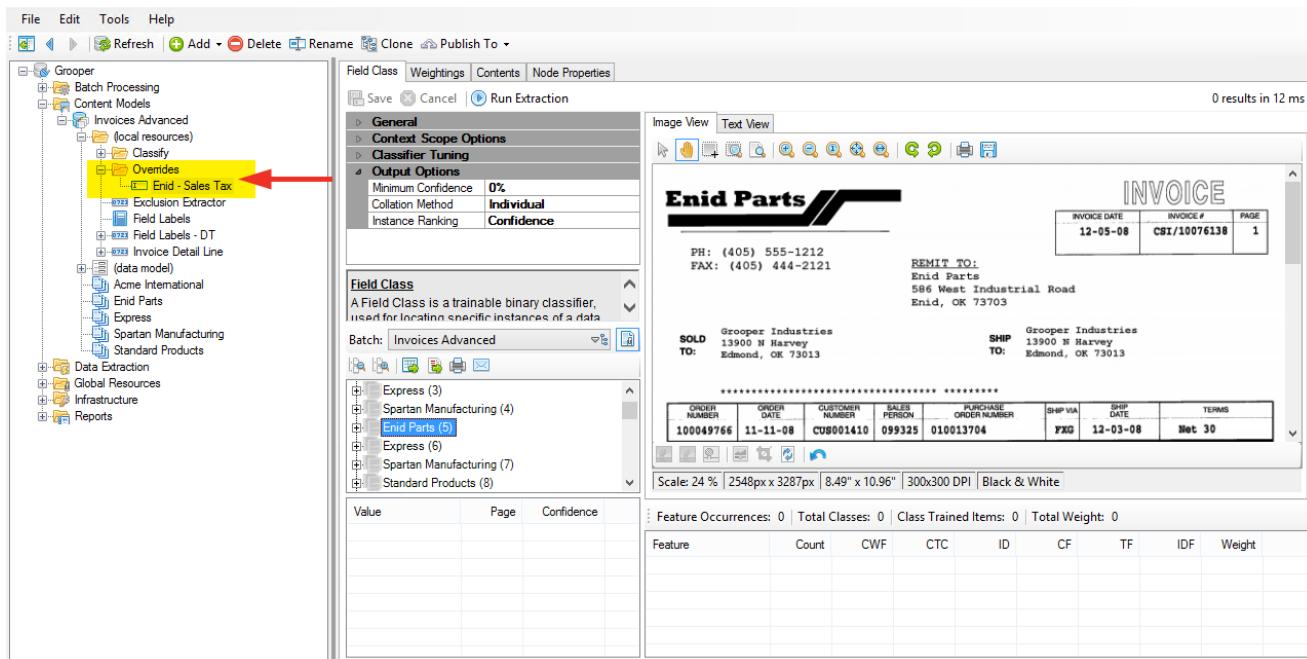
STEP 4 – SETTING UP THE PROPERTIES OF THE SALES TAX DATA FIELD

- (1) Select the Sales Tax Data Field and set the Base Type to Decimal (2) and the Format Specifier to c2.
- (3) Set the Default Extractor Type to Reference and set the Referenced Extractor to the Sales Tax Data Type that is a child of the Sales Tax Data Field. (4) Set the Default Value Expression to 0.



STEP 5 – CREATING AN ALTERNATE EXTRACTOR FOR ENID PARTS DOCTYPES

As mentioned, the **Enid Parts DocTypes** have **Sales Tax** we want to capture, but the simplest mechanism for reliably capturing it is a **Field Class**. (Okay, that's a bit of a fib, a **RegEx** pattern would be very simple too {see if you can't figure out how...}, but this is a great opportunity to introduce this concept so bear with me...) Create a folder in the **(local resources)** folder and name it **Overrides**. Then, create a **Field Class** named **Enid – Sales Tax** within it.



STEP 6 – SETTING THE EXTRACTORS FOR THE ENID – SALES TAX FIELD CLASS

With the **Enid – Sales Tax Field Class** selected, set the **Value Extractor Type** to **Internal**, then select the **Pattern** property and click the ellipsis button to bring up the **Pattern Editor** window.

(1) For the **Value Pattern** use the following:

[@Number.] {2,12}

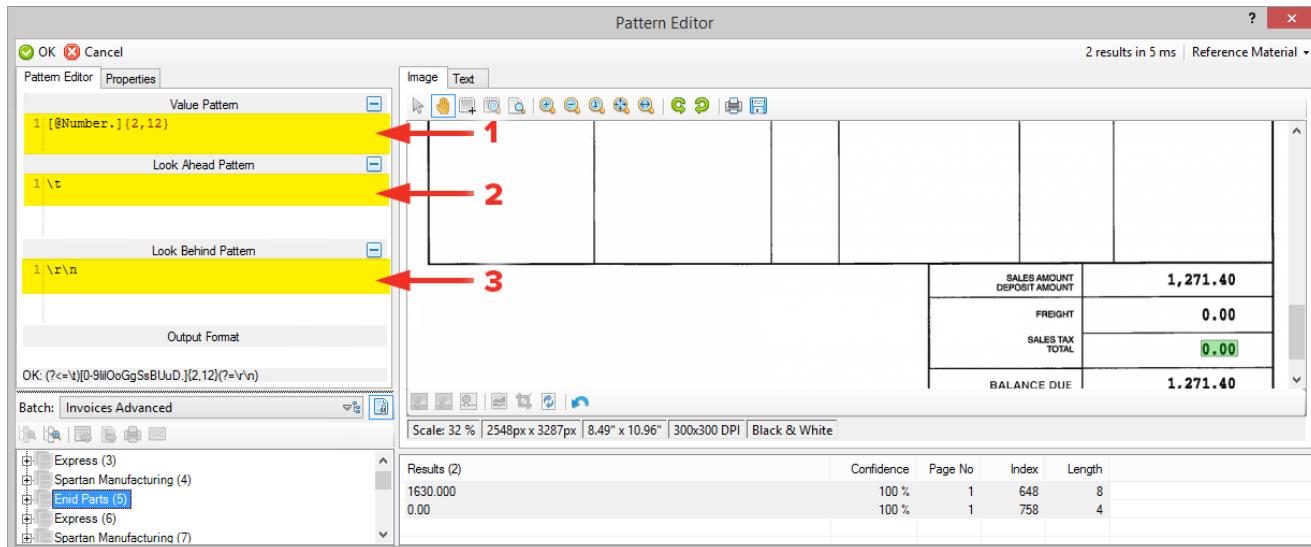
(2) For the **Look Ahead Pattern** use the following:

\t

(3) For the **Look Behind Pattern** use the following:

\r\n

Make sure **Tab Marking** is enabled in the **Text Preprocessing** area of the **General** section on the **Properties** tab. The **Feature Extractor** can be left to point to **nGrams 1-3**.



STEP 7 – EDITING THE CONTEXT ZONES

(1) Bring up the **Context Zones** options and delete Zone 2. (2) Use the following settings for Zone 1.

Zone 1 – Left: **-2.4** Top: **-0.2** Right: **0.25** Bottom: **0.1**

Save and run Extraction on the **Enid Parts (5)** document.

The screenshot shows the Grooper interface with the 'Content Models' tree on the left. A 'Field Class' dialog is open for the 'Enid Parts (5)' document. In the 'General' tab, under 'Context Scope Options', the 'Context Zones' section is highlighted. A red arrow labeled '1' points to the 'Context Zones' button. Another red arrow labeled '2.1' points to the '2 items' link. A smaller red arrow labeled '2' points to the '(-2.4, -0.2);(0.25, 0.1)' entry in the 'Zone No' column of the 'Context Zones' table. The right side of the screen shows the extracted data from the 'Enid Parts (5)' document, including a table with columns for 'SALES AMOUNT', 'DEPOSIT AMOUNT', 'FREIGHT', 'SALES TAX TOTAL', and 'BALANCE DUE'. Below the table, there's a results table showing two items with confidence levels of 100% and page numbers 1 and 648/758. At the bottom, there's a feature occurrence table.

STEP 8 – TRAINING SALES TAX AND TOTAL FEATURES

(1) On Enid Parts (5), (2) select the 0.00 value that highlights the **sales tax** and **total** features. Train it positively.

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
total	1	0	1.000000	1.000000	0.500000	1.000000	0.500000	0.500000
sales tax	1	0	1.000000	1.000000	0.500000	1.000000	0.500000	0.500000

STEP 9 – MINIMUM CONFIDENCE

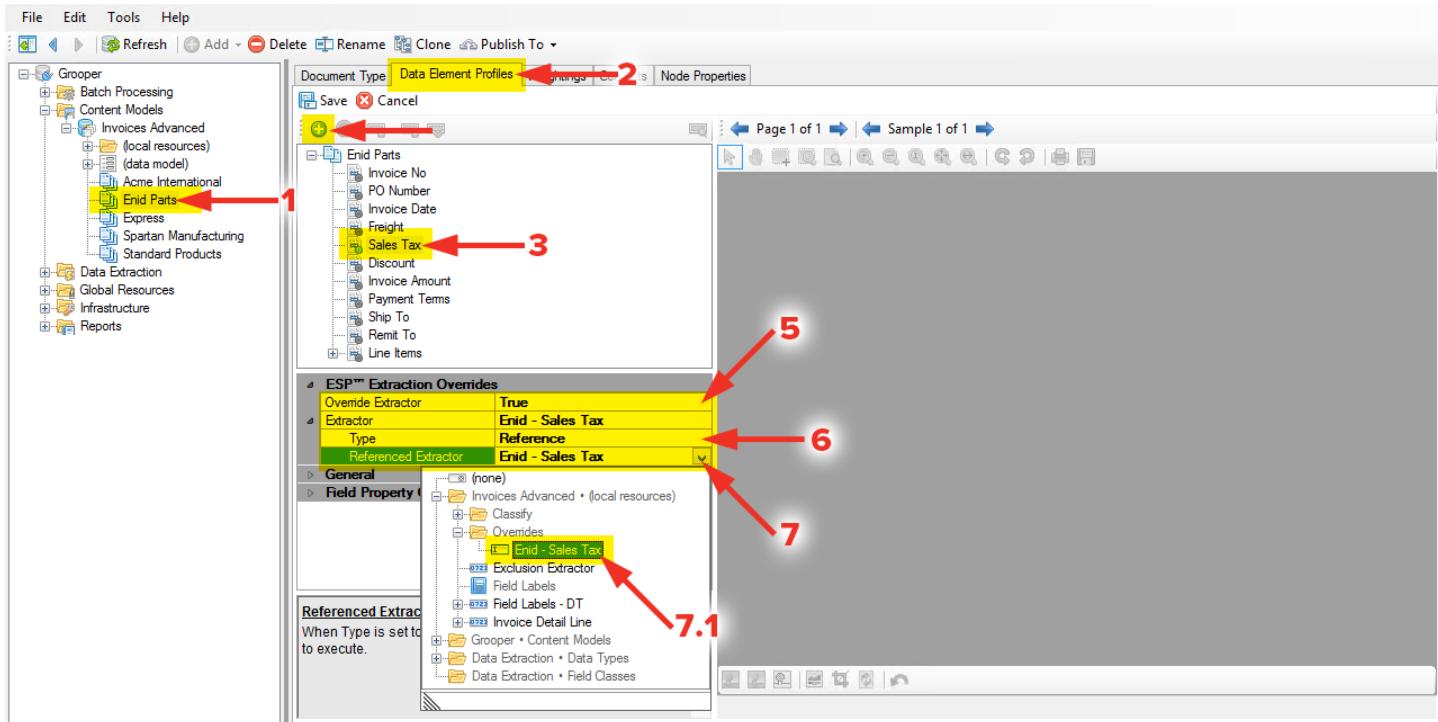
No other features need to be trained, so with a confidence of **100%**, set the **Minimum Confidence** to anything above zero to eliminate unwanted values being returned.

Feature	Count	CWF	CTC	ID	CF	TF	IDF	Weight
total	1	1	0	1.000000	1.000000	0.500000	0.301030	0.150515
sales tax	1	1	0	1.000000	1.000000	0.500000	0.301030	0.150515

STEP 10 – DATA ELEMENT PROFILES

With this new extractor made, it needs to be applied to the **Enid Parts DocType**. **(1)** Select the **Enid Parts** document and **(2)** click on the **Data Element Profiles** tab. **(3)** Select the **Sales Tax Data Field** and **(4)** click the plus button. In the **ESP Extraction Overrides** section **(5)** set **Override Extractor** to **True**. **(6)** Set the **Extractor Type** to **Reference**, and **(7)** set the **Referenced Extractor** to the **Enid – Sales Tax Field Class** within the **Invoices Advanced • (local resources) > Overrides** area. Save the changes.

When extraction is run now, this profile will override the **Default Extractor** for the **Sales Tax Data Field** and use this specified extractor instead.



FINAL DATA MODEL ADJUSTMENT AND REVIEW

There are a few minor tweaks to be made to the appearance of the **Data Model**. It is also best to look at and review, in once place, the data that is being returned by all the extractors that were just setup.

DATA MODEL AND DATA FIELD APPEARANCE SETTINGS

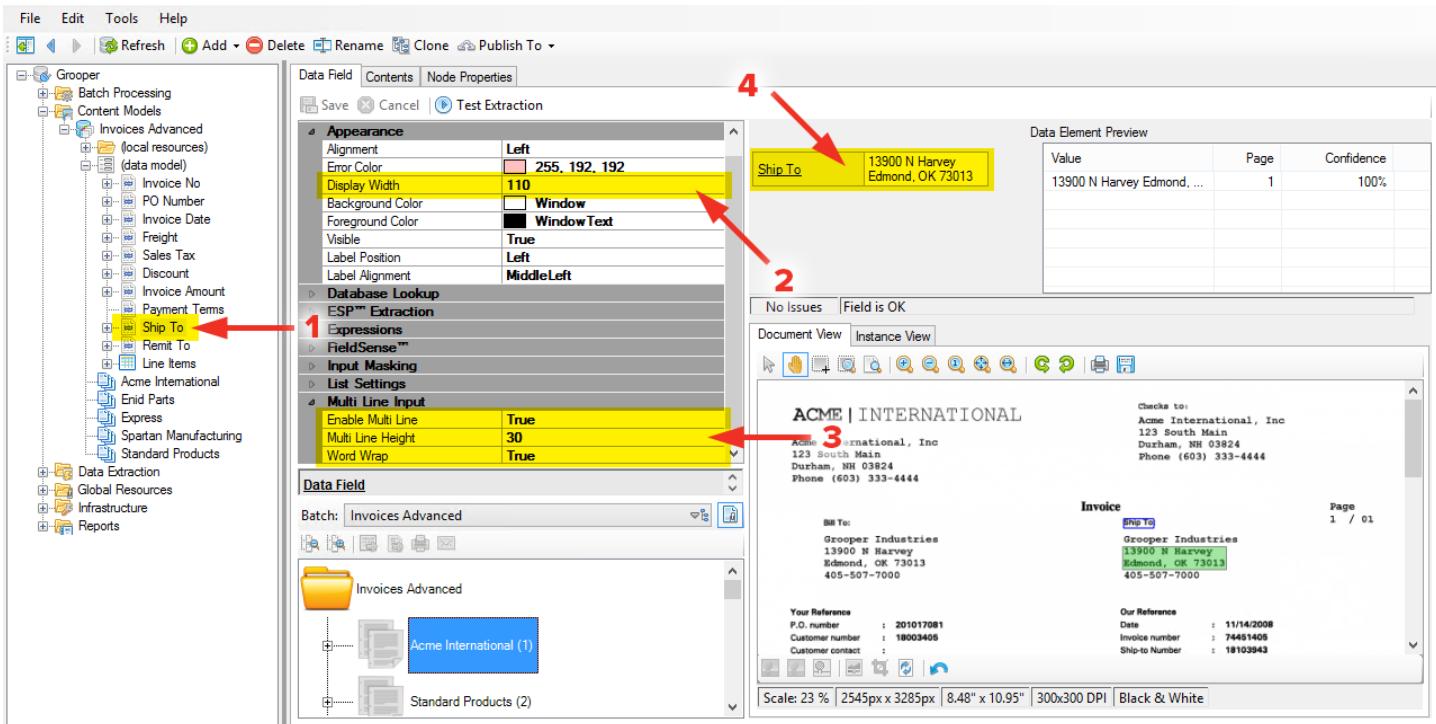
While not critical to get the data you want, appearance settings play a functional role in the attractiveness in the way that captured data is given back to you when reviewing said data. Not all the individual appearance properties need to be covered in this case, but a select few will be.

ADDRESS DISPLAY

(1) Select the **Ship To Data Field** and **Test Extraction**. Notice, in the **Data Element Preview**, that the width of the field is cutting off the way the information is being display. Also, an address isn't written on a single line.

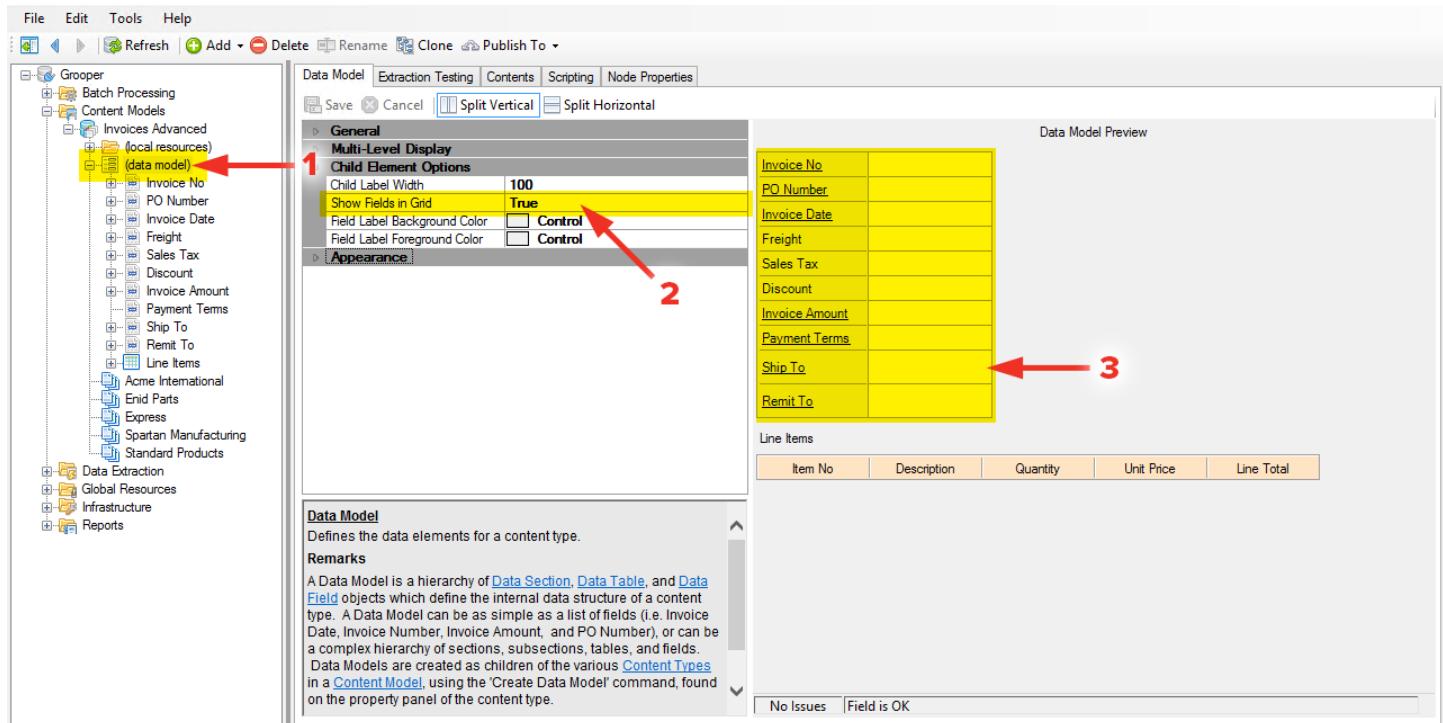
(2) In the **Appearance** section change the **Display Width** to **110**. **(3)** In the **Multi Line Input** section, change **Enable Multi Line** to **True**, **Multi Line Height** to **30**, and **Word Wrap** to **True**. **Save** and **Test Extraction** and **(4)** notice the difference.

Apply these same settings to the **Remit To** field.



DATA MODEL – SHOW FIELDS IN A GRID

- (1) Select the (data model) and in the Child Element Options section (2) set Show Fields in a Grid to True.
- (3) Notice now all the fields displayed in a table like structure. An interesting tid-bit for this feature is that the longest Display Width of a given field will determine the width for all fields when in a grid like this. Considering that, changing the Remit To Display Width to 110 just now, and using this feature, puts the Display Width for all fields to 110.

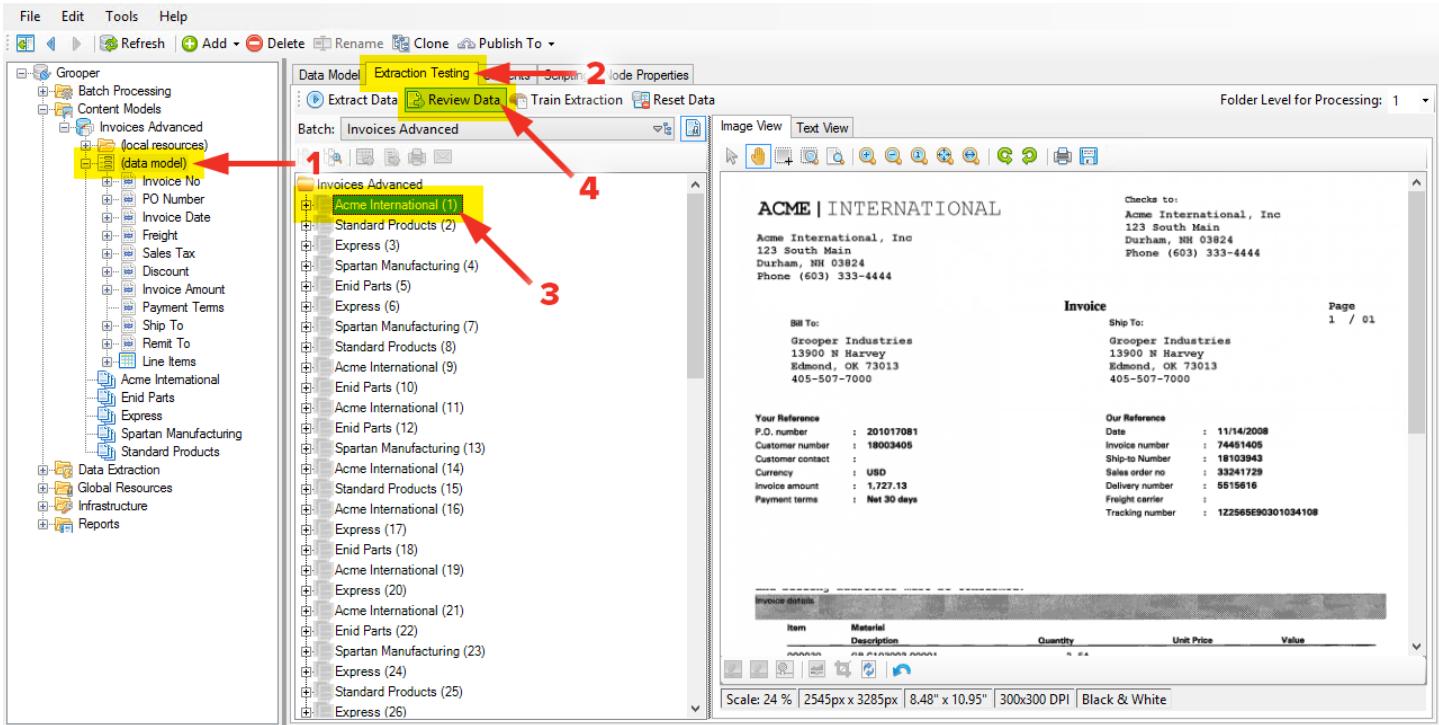


EXTRACTION TESTING

Any amount of extraction testing that has occurred as of yet has been done either while developing pattern recognition, or on a per field basis. With the **Data Model** fully built out, it is best to review the data as a whole. **Data Models** allow for this review in a module that can resemble the **Data Review Attended Module**.

REVIEW DATA

(1) Select the **(data model)** and (2) click the **Extraction Testing** tab. From here, (3) select a document in the **Batch Viewer** and (4) click the **Review Data** button.



REQUIRED FIELDS AND DEFAULT VALUES

In the [Grooper Data Extraction Test Utility](#) that appears, there are a few things to take note of. The main things to notice are that all **Data Fields** that have been set to required are empty and reddish. This is to let you know there are validation errors. This is also made apparent by the **7 Issues – Value is required.** at the bottom. There are also 3 fields that are not returning errors, and instead have default values of **\$0.00**. Given that those fields were not set to required, and that they were given values for their **Default Value Expression**, this should seem logical.

The screenshot shows the Grooper Data Extraction Test Utility interface. At the top, the menu bar includes 'Extract', 'Train', 'Auto Train', 'Document 1 of 70', 'Field', 'Edit Data Model...', 'Close', 'Extraction Results' (selected), and 'Instance Viewer'. The main area displays an 'ACME | INTERNATIONAL' invoice. On the left, a table lists various fields: Invoice No (74451405), PO Number (201017081), Invoice Date (11/14/2008), Freight (\$4.20), Sales Tax (\$0.00), Discount (\$246.13), Invoice Amount (\$1,727.13), Payment Terms (Net 30), Ship To (Grooper Industries, 13900 N Harvey, Edmond, OK 73013, 405-507-7000), and Remit To (Acme International, Inc, 123 South Main, Durham, NH 03824, Phone (603) 333-4444). Below this is a 'Line Items' table with one row: Item No (GB.C103003-0001), Description (BRACKET), Quantity (2), Unit Price (984.53), and Line Total (1,969.06). On the right, detailed sections show 'Checks to:' and 'Invoice' information. A 'Your Reference' table lists P.O. number, Customer number, Customer contact, Currency, Invoice amount, and Payment terms. An 'Our Reference' table lists Date, Invoice number, Ship-to Number, Sales order no, Delivery number, Freight carrier, and Tracking number. At the bottom, a message says '7 Issues Value is required.' and 'No Issues Field is OK'.

EXTRACT DATA

(1) Click the **Extract** button at the top left, and due to all the work that was put into the **Data Model**, the fields should be populated with information. **(2)** Feel free to click the advanced document arrow to review subsequent documents.

This screenshot shows the same interface after the 'Extract' button has been clicked. Red arrows point from the 'Extract' button (labeled 1) and the 'Field' dropdown (labeled 2) to the respective areas in the top menu bar. The 'Extraction Results' tab is still selected. The 'Line Items' table now shows two rows: the first row for 'BRACKET' and a second row for '000020 GB.C103003-0001'. The rest of the interface remains the same, displaying the ACME | INTERNATIONAL invoice details and reference tables.

PHASE 5 – DELIVER

Having built all the tools required to take our documents through the first four phases, it is now time to send the extracted information to a place it can be useful.

BUILDING A BATCH PROCESS

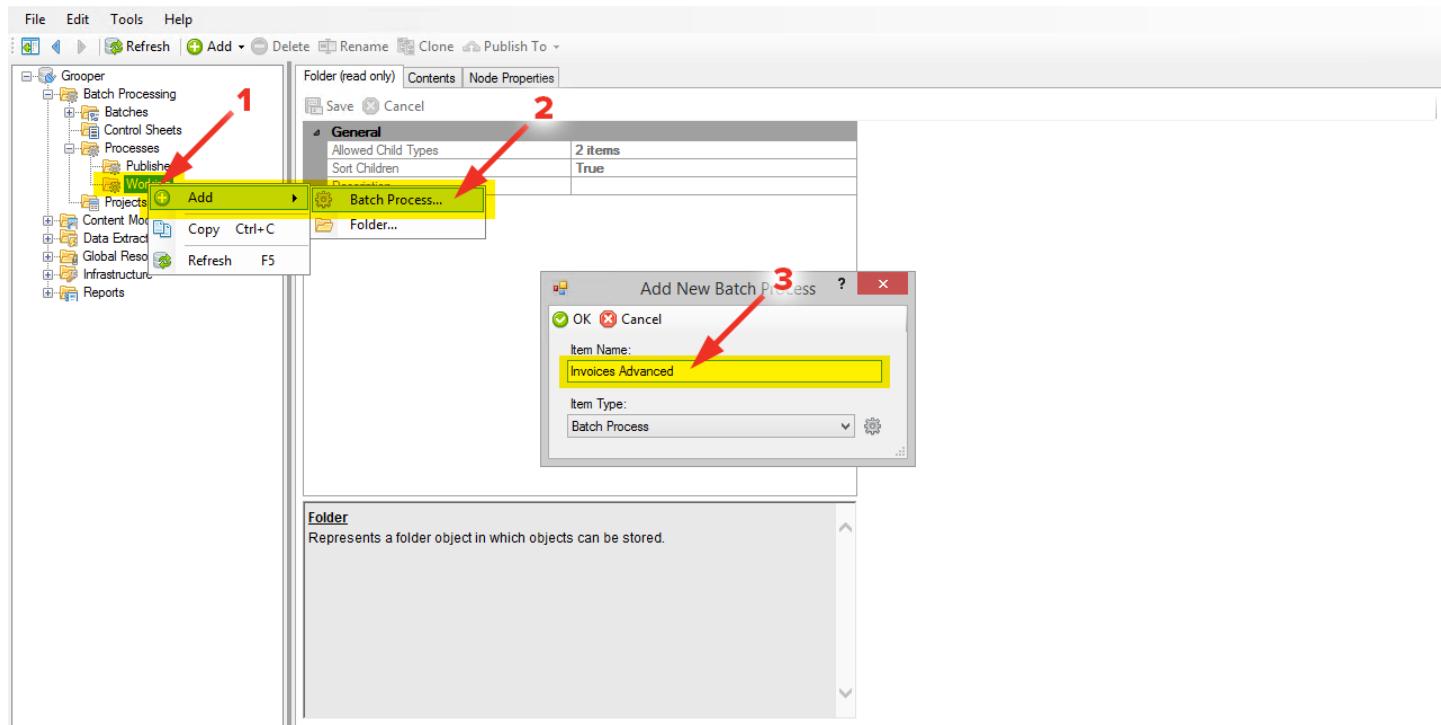
In the previous document, the [Batch Process](#) was built out as the tools were being learned about, yet in this document, a [Batch Process](#) hasn't even been mentioned. Delivery of the extracted information will make the most sense as a final step of a process.

CREATING THE STEPS OF A PROCESS

Before making the step for delivery, let's build out the logical sequence of events that will procedurally take our documents through each phase.

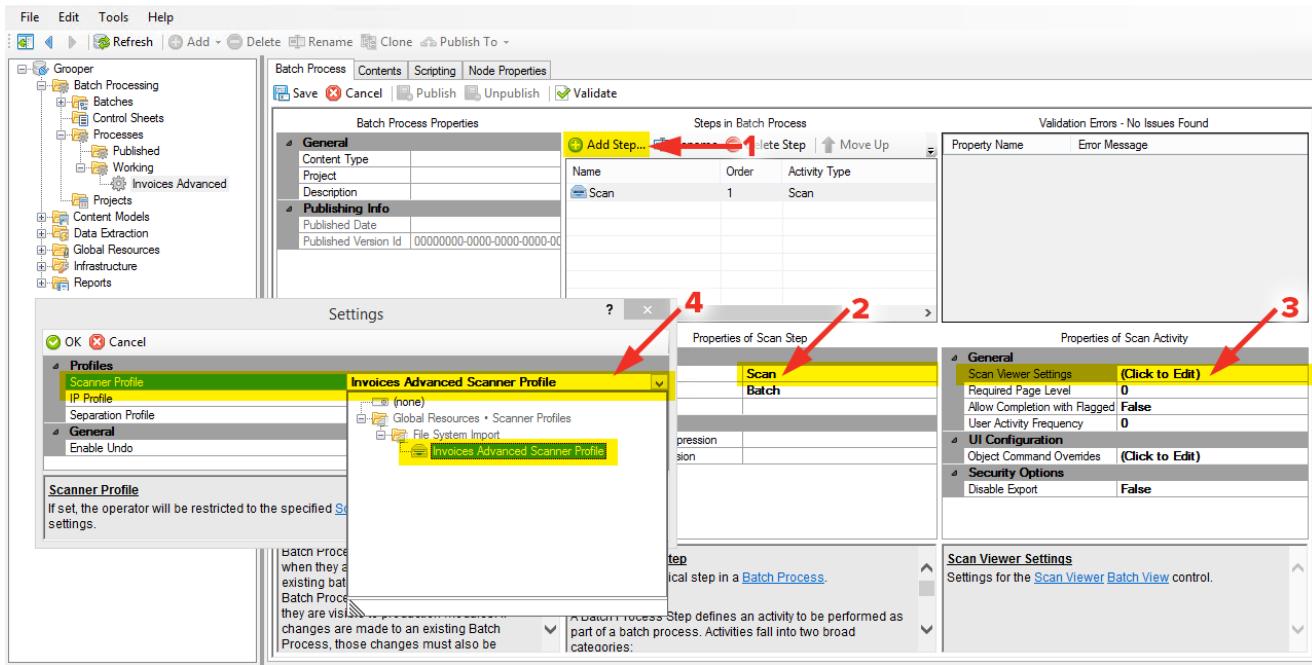
STEP 1 – CREATING A BATCH PROCESS

(1) Navigate to [Grooper > Batch Processing > Processes](#) and select the [Working](#) node. **(2)** Right click and [Add > Batch Process...](#) and **(3)** name it [Invoices Advanced](#). From here, when adding steps, think through the sequence of events about the [Phase](#) in which they occur, and it will make the most sense.



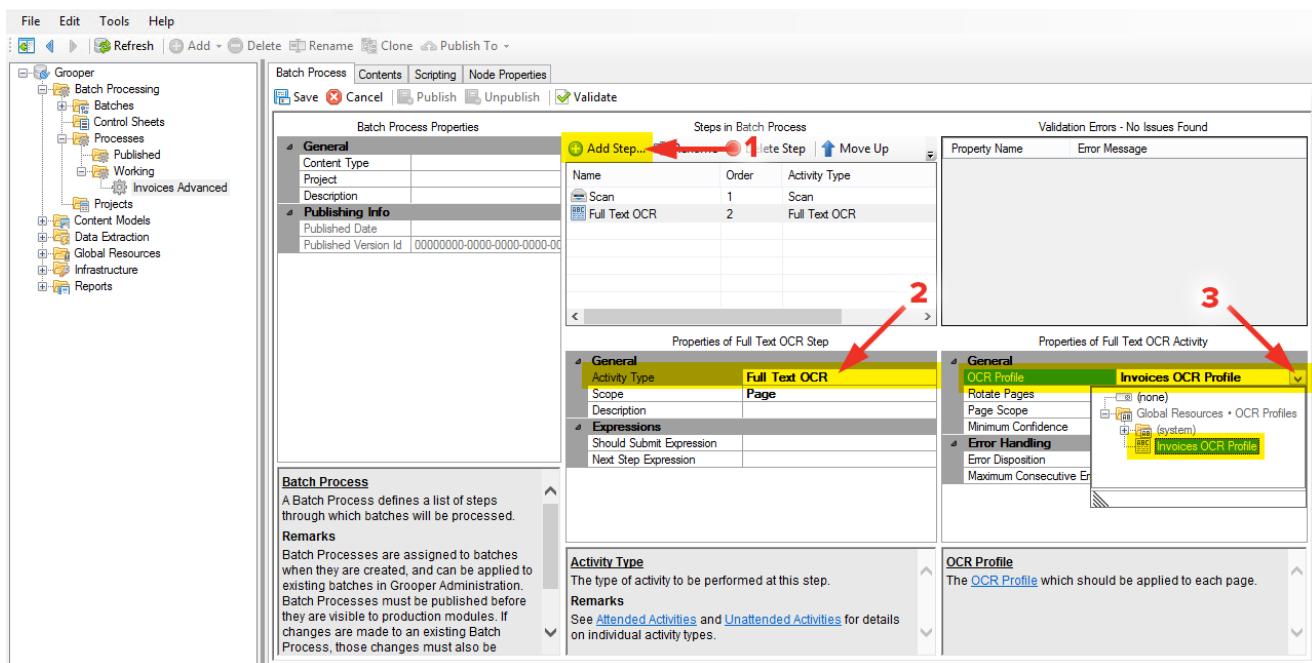
STEP 2 – PHASE 1 – ACQUIRE • SCAN

To acquire the documents for our **Batch Process** we setup a **Scanner Profile**. (1) Click **Add Step...** and (2) set the **Activity Type** to **Scan**. After this (3) select the **Scan Viewer Settings** property and click the ellipsis button. In the **Settings** window that appears (4) select the **Scanner Profile** property and set it to the **Invoices Advanced Scanner Profile** within the **Global Resources • Scanner Profiles > File System Import** area.



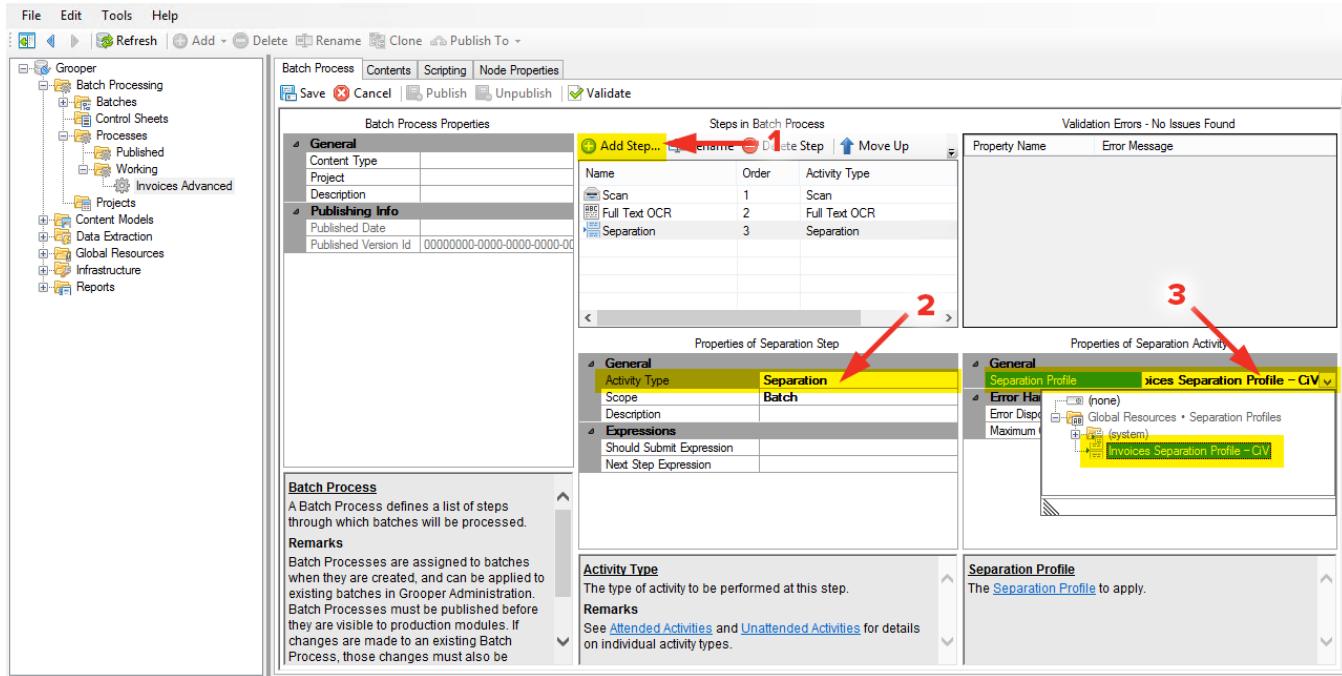
STEP 3 – PHASE 2 – CONDITION • OCR

The next step in the **Batch Process** consists of conditioning the documents. (1) Add another **Step** and (2) set its **Activity Type** to **Full Text OCR**. (3) Set the **OCR Profile** to the **Invoices OCR Profile** within the **Global Resources • OCR Profiles** area.



STEP 4 – PHASE 3 – ORGANIZE • SEPARATION

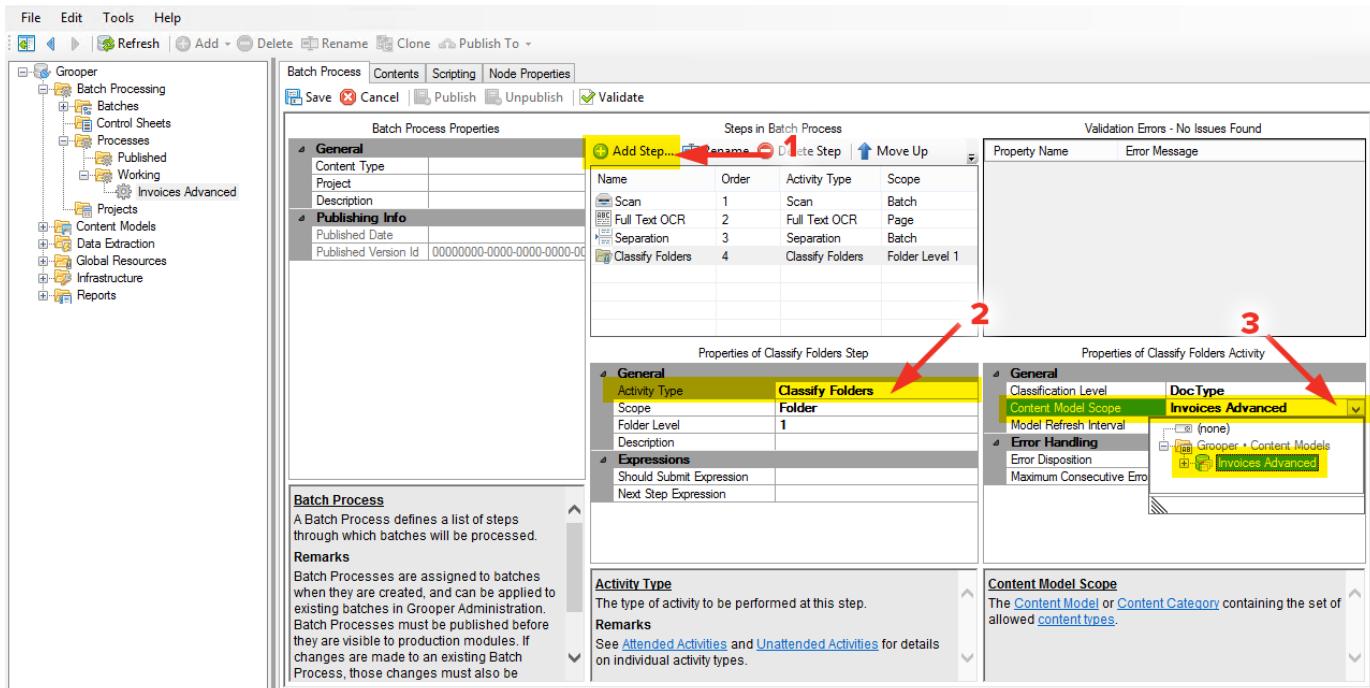
Only after conditioning the documents can they be organized. (1) Add a Step and (2) set its Activity Type to Separation. (3) Set the Separation Profile to the Invoices Separation Profile – CiV in the Global Resources • Separation Profile area.



STEP 5 – PHASE 3 – ORGANIZE • CLASSIFICATION

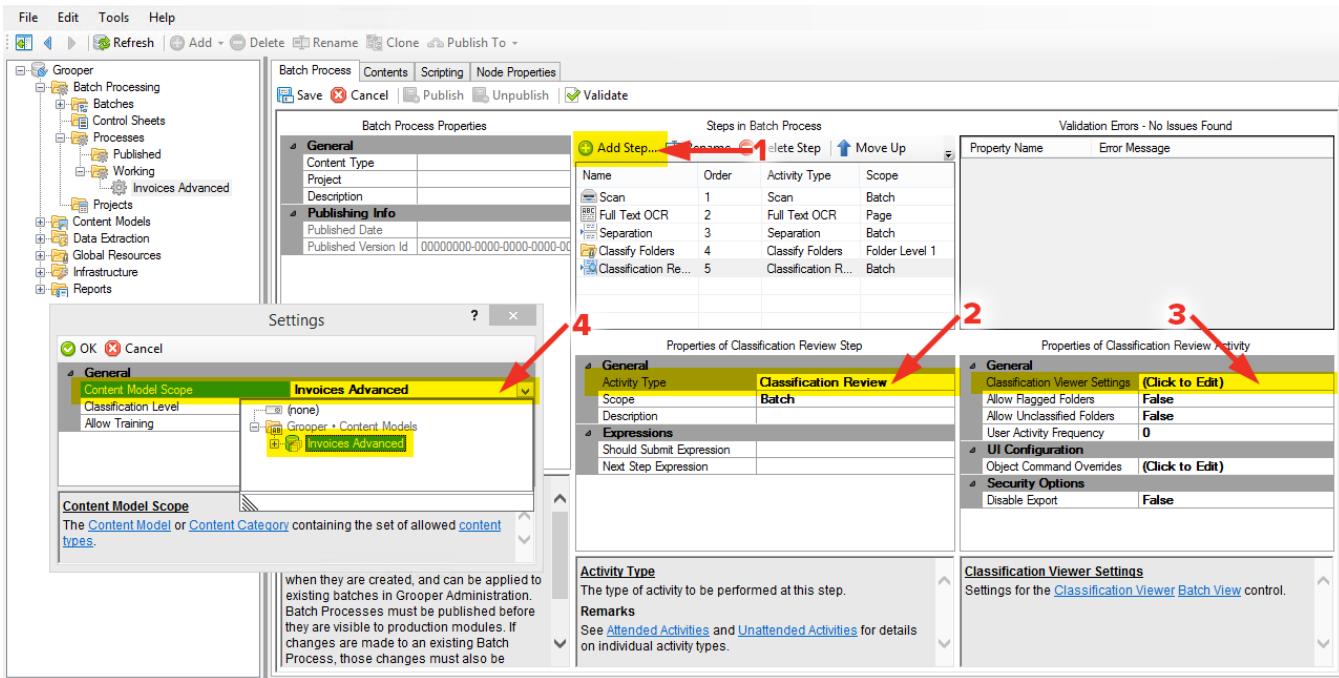
With the pages separated into logical documents, the folders that they exist within can now be classified.

(1) Add another Step, (2) set its Activity Type to Classify Folders, and (3) set the Content Model Scope to the Invoices Advanced Content Model within the Grooper • Content Models area.



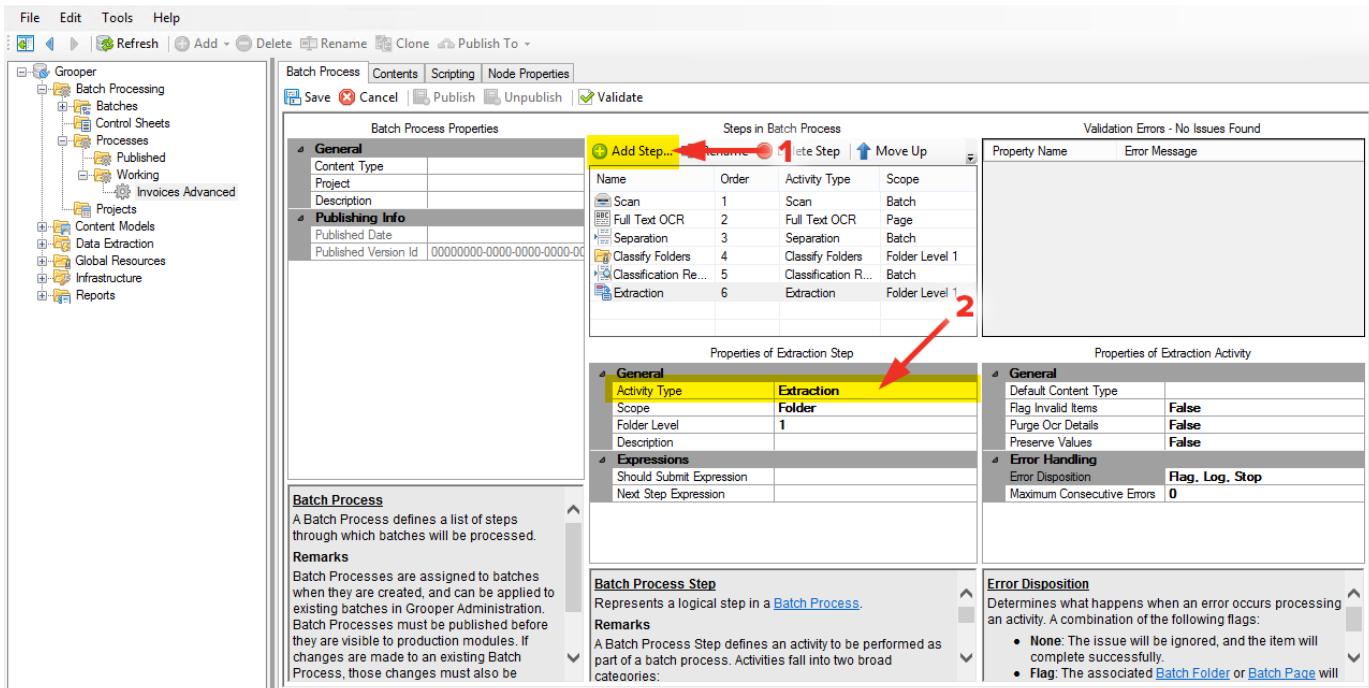
STEP 6 – PHASE 3 – ORGANIZE • CLASSIFICATION REVIEW

It is typically best practice to have a person review **Classification**. (1) Add a Step, (2) set its **Activity Type** to **Classification Review**, and (3) click the ellipsis button for the **Classification Viewer Settings** property. In the **Settings** window (4) set the **Content Model Scope** property to the **Invoices Advanced** Content Model within the **Grooper • Content Models** area.



STEP 7 – PHASE 4 – EXTRACTION

As much work went into developing the **Content Model** to perform the extraction, it almost seems anti-climactic that it comes down to one step, but I digress. As such, add another and set its **Activity Type** to **Extraction**.



STEP 8 – PHASE 4 – EXTRACTION • DATA REVIEW

Reviewing the extracted data is important as it puts human eyes on procedurally collected information to verify its accuracy. **(1)** Add a Step and **(2)** set its **Activity Type** to **Data Review**.

Name	Order	Activity Type	Scope
Scan	1	Scan	Batch
Full Text OCR	2	Full Text OCR	Page
Separation	3	Separation	Batch
Classify Folders	4	Classify Folders	Folder Level 1
Classification Re...	5	Classification R...	Batch
Extraction	6	Extraction	Folder Level 1
Data Review	7	Data Review	Batch

Properties of Data Review Step

General	Data Review
Activity Type	Data Review
Scope	Batch
Description	
Expressions	
Should Submit Expression	
Next Step Expression	

Properties of Data Review Activity

General	2 Settings (Click to Edit)
Allow Completion with Invalid	False
User Activity Frequency	0
UI Configuration	
Object Command Overrides	(Click to Edit)
Security Options	
Disable Export	False

Batch Process
A Batch Process defines a list of steps through which batches will be processed.
Remarks
Batch Processes are assigned to batches when they are created, and can be applied to existing batches in Grooper Administration. Batch Processes must be published before they are visible to production modules. If changes are made to an existing Batch Process, those changes must also be

Batch Process Step
Represents a logical step in a [Batch Process](#).
Remarks
A Batch Process Step defines an activity to be performed as part of a batch process. Activities fall into two broad categories:

Data Review
Provides a user interface for entering data or validating the results of [Extraction](#).
Remarks
The user interface for the Data Review activity includes the [Index Navigator](#) Batch View control.

STEP 9 – PHASE 5 – DELIVER • DOCUMENT EXPORT

The culmination of our **Batch Process** is the delivery of the data. **(1)** Add a Step and **(2)** set its Activity Type to **Document Export**. **(3)** Set **File System Export** as the **Export Provider** and **(4)** open the settings for the **Export Settings** property. In the **File System Export** settings window **(5)** set the **Base Export Folder** to:

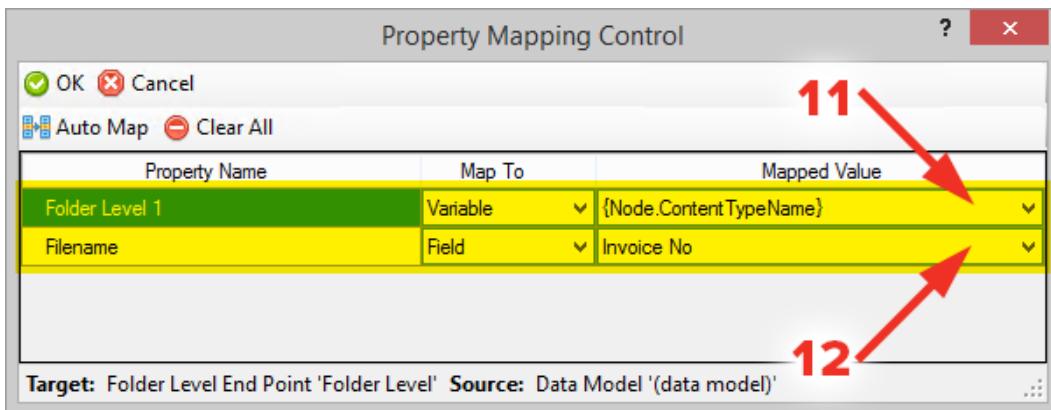
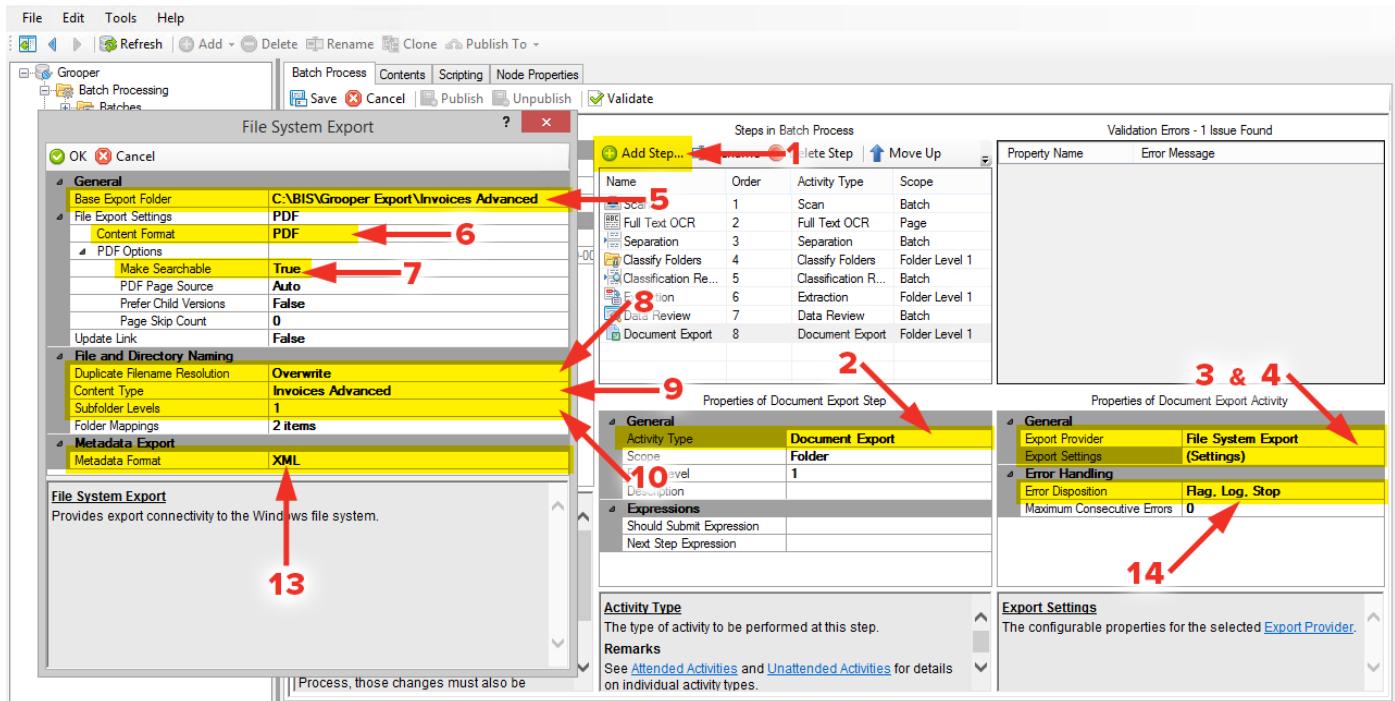
C:\BIS\Grooper Export\Invoices Advanced

Expand **File Export Settings** and **(6)** set the **Content Format** to **PDF**. Expand **PDF Options** and **(7)** set **Make Searchable** to **True**.

(8) Change **Duplicate Filename Resolution** to **Overwrite**. **(9)** Set the **Content Type** to the **Invoices Advanced Content Model**. **(10)** Set **Subfolder Levels** to **1**. Open the **Folder Mappings** settings and **(11)** set **Folder Level 1** to a Variable of **[Node.ContentTypeName]**. **(12)** Set **Filename** to a Field of **Invoice No.**.

(13) Change the **Metadata Format** to **XML**.

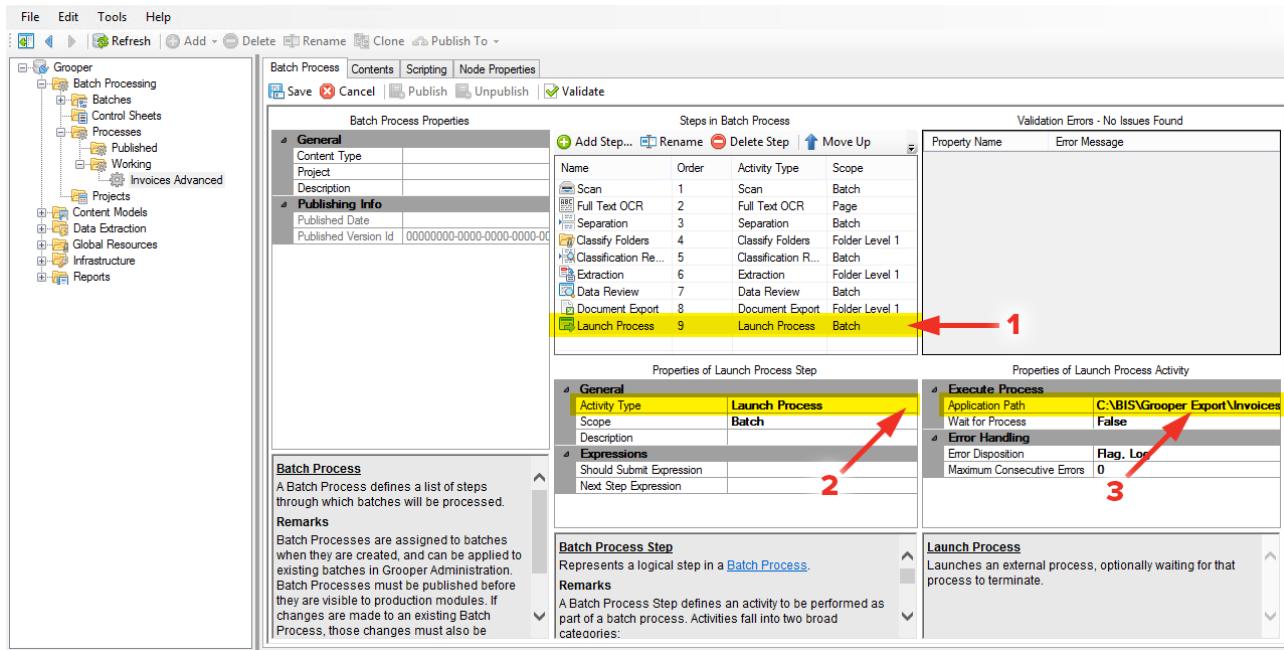
Finally, **(14)** change the **Error Disposition** to **Flag, Log, Stop**. Its worth considering this **Error Disposition** for other steps such as **Full Text OCR**, so that if an error occurs, the **Batch Process** stops completely to prompt Administrator review. It is critical, however, it be set for **Document Export**, since the batch is disposed of immediately after export.



STEP 10 – PHASE 5 – DELIVER • LAUNCH PROCESS

(1) Add a **Step** and (2) set its **Activity Type** to **Launch Process**. This is a unique activity in **Grooper** that allows external processes to be launched. In this case, we will set it to open a **Windows File Explorer** that houses the now exported documents. (3) Set the **Application Path** to:

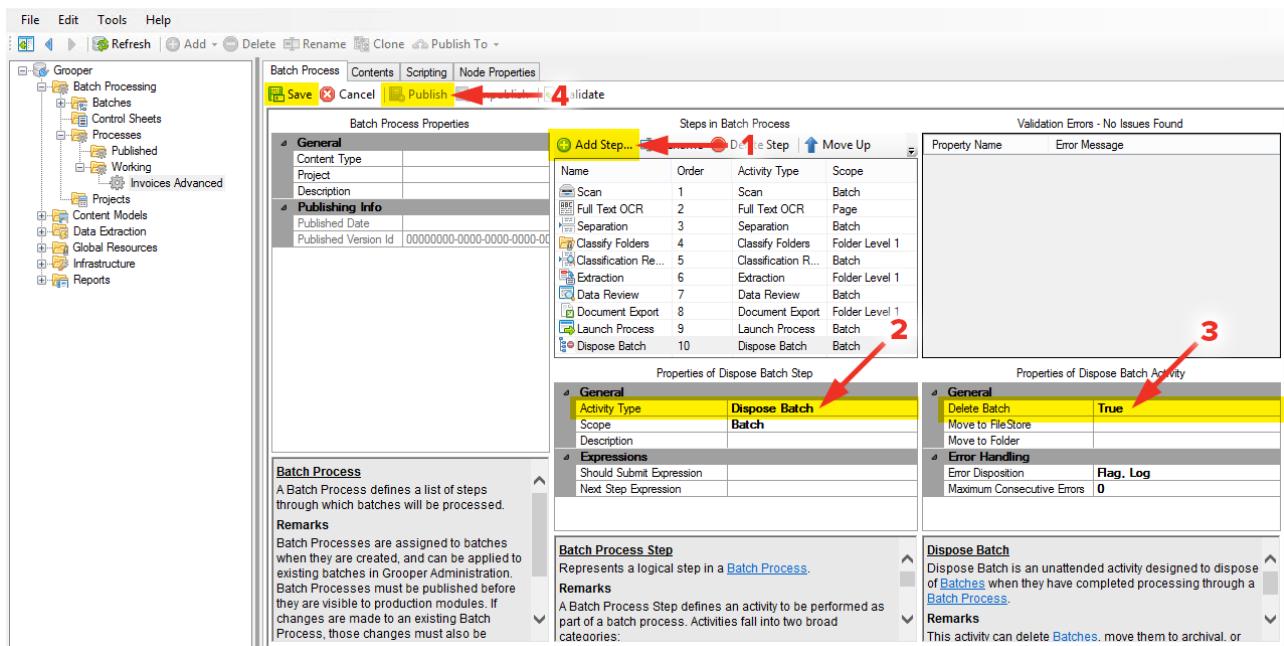
C:\BIS\Grooper Export\Invoices Advanced



STEP 11 – DISPOSE BATCH

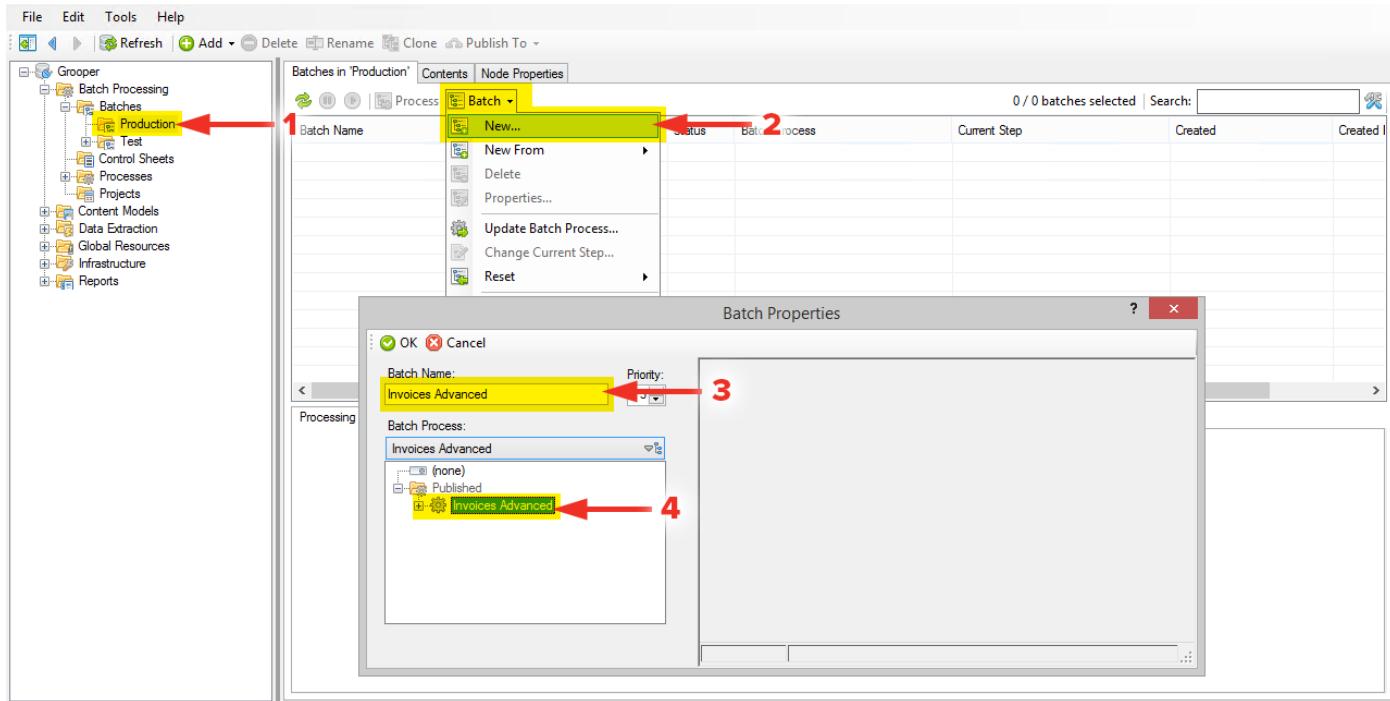
Grooper is not a destination for data, rather, a mechanism to **Extract, Transform, and Load** that data somewhere else. As such, it is prudent to dispose of batches once they are complete. (1) Add a **Step** and (2) set its **Activity Type** to **Dispose Batch**. (3) Set **Delete Batch** to **True**.

(4) Save and Publish this Batch Process.



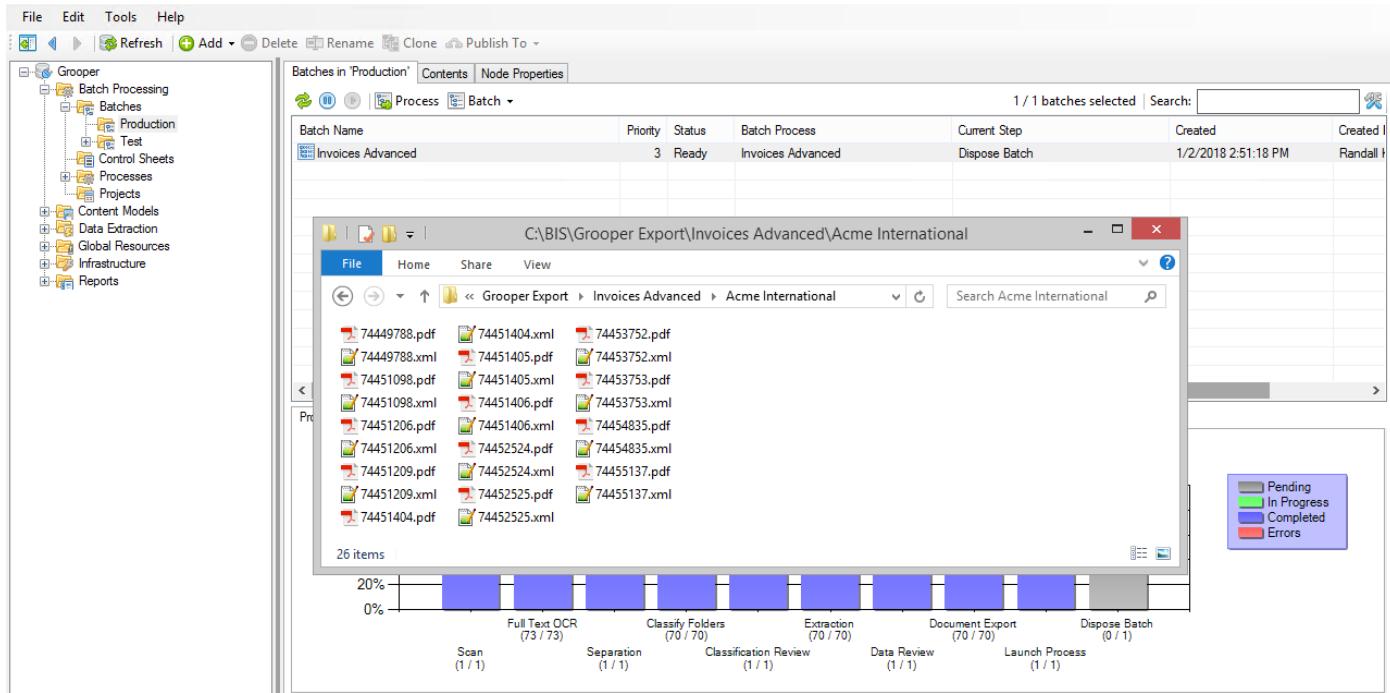
STEP 12 – LAUNCHING A PRODUCTION BATCH

(1) Navigate to **Grooper > Batches** and with either **Production** or **Test** selected, (2) click the **Batch** drop-down and create a **New...** batch. (3) Name it **Invoices Advanced**. (4) Select the Invoices Advanced Batch Process.



STEP 13 – VIEWING RESULTS

Either via **Grooper Services** you've established, or by manually processing each step, you'll get to a point where the **Launch Process** step will open a **Windows Explorer** to the export destination. Look through this export folder to view your searchable **PDFs** as well as the indexed **XML** data.



A FINAL NOTE

To begin, you can download everything that was worked on throughout the course of this document by following this link:

[Grooper A.C.E. - Architect Training Vol.2 - Advanced Invoice Processing - Final Export](#)

This zip file contains the source batch that the **Scanner Profile** refers to, as well as the complete **Batch Process**. Due to all the references it has to other profiles and the **Content Model**, the **Batch Process** object brings with it all necessary other objects to complete the process.

A lot has been covered in this volume of training. We focused on [importing objects](#), creating an [OCR Profile](#), [Change in Value Separation](#), [Classification based on positive extractors](#), [Data Tables](#), [FuzzyRegEx](#), [Field Classes](#), [Lexicons](#), [Cheat Codes](#), [Ordered Arrays](#), [Exclusion](#) and [Inclusion](#) extractors, and [Data Element Profiles](#). It's quite a lot to take in all at once, and honestly this is just the beginning.

I feel quite confidant in saying that having made it through learning a bit about applying all these tools, and the navigation in and out of **Grooper** required to get to this point, that your familiarity with **Grooper**, and ways to leverage it for your business purposes, is greatly heightened.

The next steps for you will involve diving in and applying this knowledge with **Grooper** and staying in contact with the team here at **BIS** to help you grow as a **Grooper** user.

We very much look forward to your growth within this **Grooper** family and seeing you in **Grooper A.C.E. • Consultant** training.

Finally, and most importantly, congratulations for making it through **Grooper A.C.E. • Architect** training!



Architect • Consultant • Engineer