

Wrapped Data Extraction

Earlier, it was difficult to extract index field values that spanned across multiple rows; for example, addresses.

If a Value Pattern regex does not find a complete match, a value pattern is not found and no Value is returned from the iteration.

To overcome this challenge, **Wrapped Data Extraction** has been introduced in Ephesoft v4.1.1.0.

You can now use **Wrapped Data Extraction** to search for regex matches in index fields with values spanning multiple rows.

When you enable **Wrapped Data Extraction**, the Value Pattern is searched in two regions:

- Directly below the partial regex match found (in the overlay)
- In the next row (extreme left of the document) below the partial regex match found at the end of the line

Overview

After the Key Pattern is identified for a document, the zonal LineDataCarriers (rows of lines) are fetched according to the overlays drawn. The system then looks for partial matches of the Value Pattern. If a partial match is found, the algorithm continues looking in the next row. Else, the value is considered as not found.

As in example, consider the image below:

```
abc def ghi jkl mno p qrs tuv wxyz ABC DEF GHI JKL MNO PQRS TUV
WXYZ !"§ $%& /() =?* '<> #|; ^~ @`' ©«» ×¼× {} abc def ghi
jkl mno p qrs tuv wxyz ABC DEF GHI JKL MNO PQRS TUV WXYZ !"§
$%& /() =?* '<> #|; ^~ @`' ©«» ×¼× {} abc def ghi jkl mno
p qrs tuv wxyz ABC DEF GHI JKL MNO PQRS TUV WXYZ !"§ $%& /()
=?* '<> #|; ^~ @`' ©«» ×¼× {} abc def ghi jkl mno p qrs tuv
wxyz ABC DEF GHI JKL MNO PQRS TUV WXYZ !"§ $%& /() =?* '<>
#|; ^~ @`' ©«» ×¼× {} abc def ghi jkl mno p qrs tuv wxyz ABC
DEF GHI JKL MNO PQRS TUV WXYZ !"§ $%& /() =?* '<> #|; ^~
@`' ©«» ×¼× {} abc def ghi jkl mno p qrs tuv wxyz ABC DEF GHI
JKL MNO PQRS TUV WXYZ !"§ $%& /() =?* '<> #|; ^~ @`' ©«»
×¼× {} abc def ghi jkl mno p qrs tuv wxyz ABC DEF GHI JKL MNO
PQRS TUV WXYZ !"§ $%& /() =?* '<> #|; ^~ @`' ©«» ×¼×
{}abc def ghi jkl mno p qrs tuv wxyz ABC DEF GHI
```

We are assuming the Value Pattern to be **tuv wxyz**. While using **Wrapped Data Extraction**, the system finds the span **tuv** in the first row. Based on this value, a decision is made whether the next row needs to be searched for subsequent values.

As in this case, the next span value is **wxyz** (in row 2), the Value Pattern finds a match and continues searching in the next row.

However, for the same Value Pattern, if **Wrapped Data Extraction** is used on row 4, the last span is **mno**. The value **mno** can never match the Value Pattern regex by appending anything to it. In this case, the system discontinues the search in the next row.

Taking the example of the below sample. First, the algorithm will try to search for possible matches of the value regex pattern (if a partial match is found) directly below within the overlay drawn. In the example below, a value pattern of "Carnation Drive Brick" will be extracted as desired (irrespective of whether the overlay is drawn till the last line).

Property Information		Property Type
Address	702 Carnation Drive Brick, NJ 08724	Condominium

The extracted value will be as follows:

Key: Address

Value: Carnation Drive Brick

Fuzzy %: none

Fetch: ALL

↺

↻

🗑️

<<

<

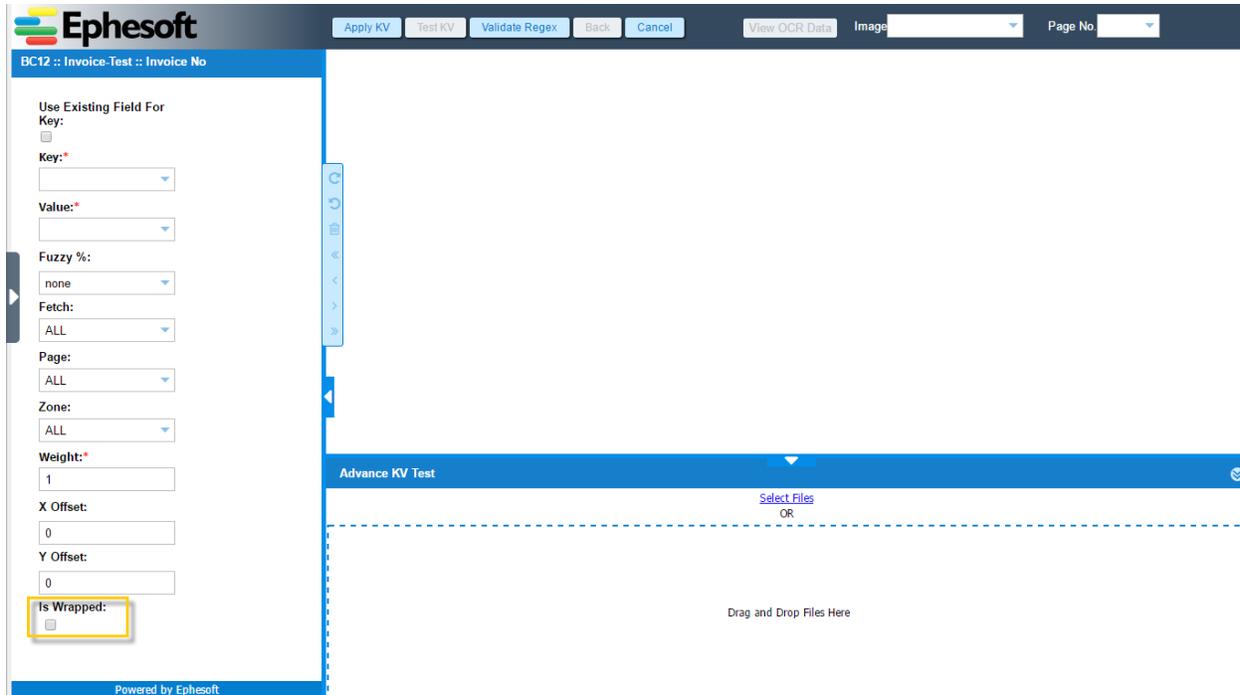
>

Total Expense Ratio	42.19%
Total Loan Amount	\$220500.0
Sales Price	\$0.0
Appraised Value	\$245000.0

Property Information	
Address	702 Carnation Drive Brick, NJ 08724

Configuration

You can configure the index fields for which **Wrapped Data Extraction** has to be used by selecting **Is Wrapped** on the **Advanced KV Extraction** screen as shown in the image below.



This is an optional feature as it might require extra computations which many use cases would not require and hence slow down the batch processing.

Configuring Wrapped Data Extraction for an Index Field

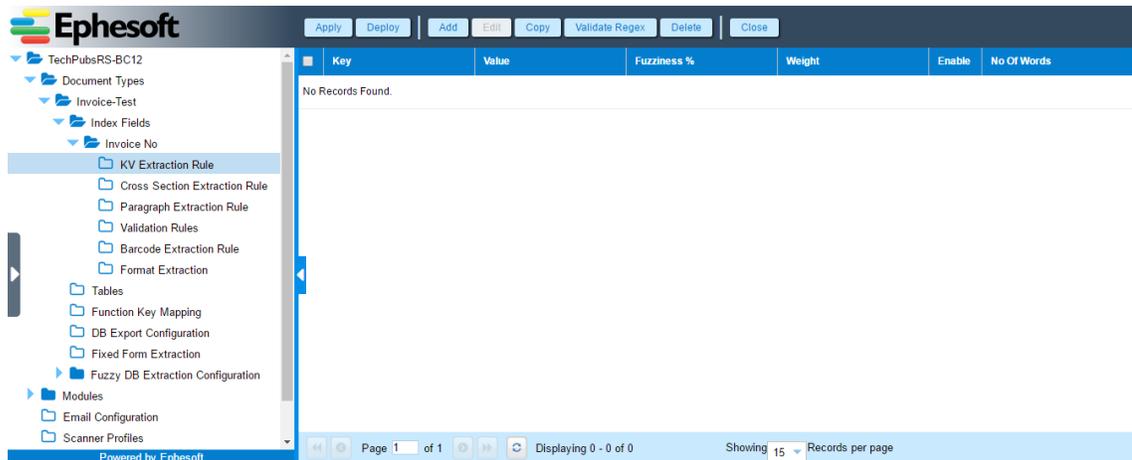
Configuration for **Wrapped Data Extraction** is similar configuring the **KV Extraction Rule**.

To configure **Wrapped Data Extraction** for an index field

1. From the DCMA Home page, click **ADMINISTRATOR** and select **BATCH CLASS MANAGEMENT**.
The Ephesoft Enterprise **Login** page displays.
2. Enter valid credentials to log in.
The **Batch Class Management** screen displays.
3. Select the batch class from the list in the **Batch Class Management** screen and click **OPEN**.
The batch class opens with **Document Types** node selected by default.
4. Select the document type from the list and click **OPEN**.
The document type node expands displaying a list of index fields.

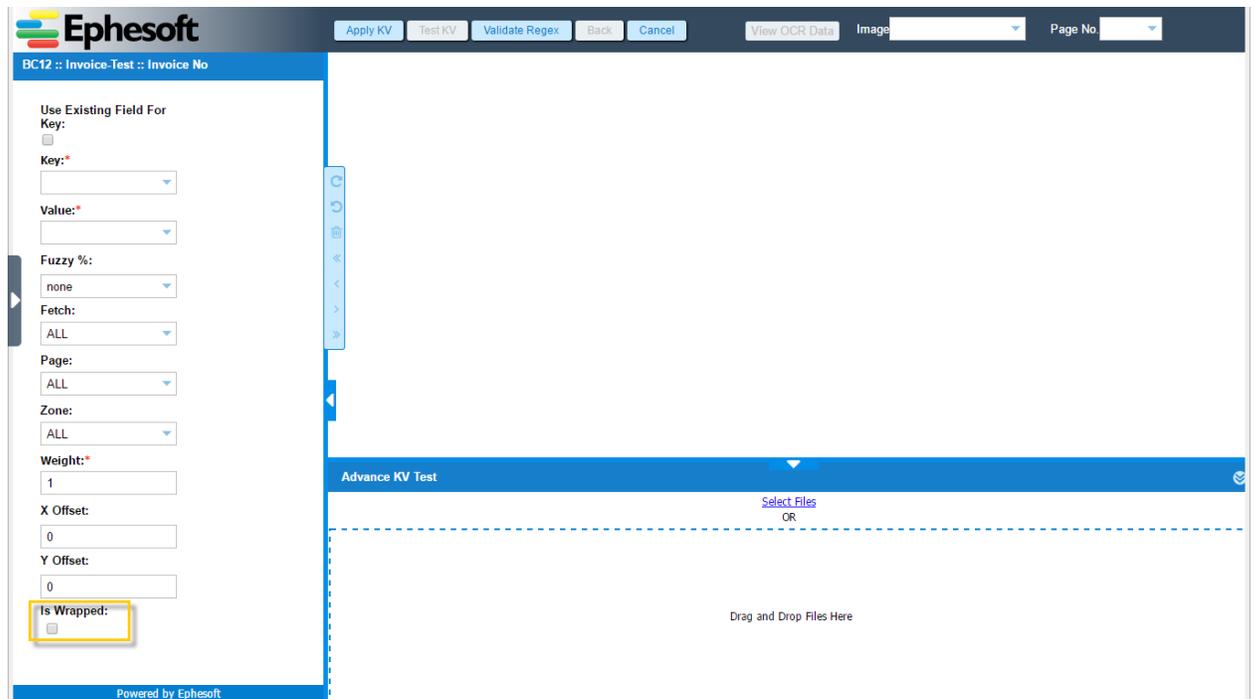
5. Select the index field from the list and click **OPEN**.

The index field node expands displaying all the available extraction rules in the left navigation pane and **KV Extraction Rule** selected by default as shown in the image below.



6. Click **ADD**.

The following screen displays.

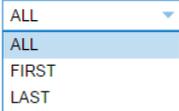
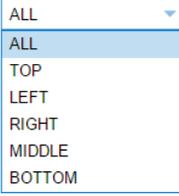


7. Click **Select Files** link from **Advance KV Test** section and upload an image file.

The uploaded image is displayed in the image view pane.

8. Enter the relevant configuration details as described in the table below:

Component	Description				
Use Existing Field for Key					
Key					
Value	<p>You can use this parameter to find values within the identified paragraph. This parameter is used after extraction is done.</p> <p>You can use this parameter to filter out the extraction results. All values which do not match the value pattern are discarded.</p> <p>Value:*</p> <div data-bbox="483 772 685 873"> <input type="text"/> <input type="text" value="Regex Builder"/> <input type="text" value="Regex Pool"/> </div> <p>You can enter a regular expression or use Regex Builder/Regex Pool options to enter a search pattern.</p>				
Fuzzy %	<p>You can use this parameter to do a fuzzy search while searching for Paragraphs using the Start Pattern.</p> <p>Fuzzy %:</p> <div data-bbox="483 1117 698 1285"> <input type="text" value="none"/> <input type="text" value="none"/> <input type="text" value="10%"/> <input type="text" value="20%"/> <input type="text" value="30%"/> </div>				
Fetch	<p>Depending on the value you select for this parameter, the Value tag in the batch XML is populated with ALL/FIRST/LAST Regex matches in the paragraph.</p> <p>Fetch:</p> <div data-bbox="483 1465 708 1608"> <input type="text" value="ALL"/> <input type="text" value="ALL"/> <input type="text" value="FIRST"/> <input type="text" value="LAST"/> </div> <p>This parameter has the following options:</p> <table border="1" data-bbox="472 1675 1468 1864"> <tbody> <tr> <td data-bbox="472 1675 597 1772">ALL</td> <td data-bbox="597 1675 1468 1772">All the values matching the regex are updated in batch.xml with space as a delimiter.</td> </tr> <tr> <td data-bbox="472 1772 597 1864">FIRST</td> <td data-bbox="597 1772 1468 1864">First value matching the regex going from left to right in the paragraph is updated in batch.xml.</td> </tr> </tbody> </table>	ALL	All the values matching the regex are updated in batch.xml with space as a delimiter.	FIRST	First value matching the regex going from left to right in the paragraph is updated in batch.xml.
ALL	All the values matching the regex are updated in batch.xml with space as a delimiter.				
FIRST	First value matching the regex going from left to right in the paragraph is updated in batch.xml.				

Component	Description
	<p>LAST Last value matching the regex going from left to right in the paragraph is updated in batch.xml.</p>
Page	<p>You have three options available to choose from for this parameter: ALL, FIRST, and LAST.</p> <p>Page:</p>  <p>Depending on the selected value, the extraction algorithm runs on ALL/FIRST/LAST Page of the document.</p>
Zone	<p>Every page in divided into 5 zones: TOP, MIDDLE, BOTTOM, LEFT and RIGHT along with the default option of ALL.</p> <p>Zone:</p>  <p>You can use this parameter to specify the portion of the page where the algorithm searches for start value of paragraph to extract it.</p> <p>For example, if you configure this parameter value as BOTTOM, the start pattern of the paragraph is searched only in the BOTTOM zone.</p>
Weight	<p>You can use this parameter to implement weighted confidence values.</p> <p>Weight:*</p>  <p>This is used to give bias/weight to a particular extraction rule.</p>
X Offset	
Y Offset	
Is Wrapped	<p>You can select this option to enable Wrapped Data Extraction in the index field.</p>

9. Click **Test KV** from the toolbar on top of the page.

The extraction result is highlighted on the image as an overlay and is also displayed in the **Advance KV Test** grid as shown in the image below.

The screenshot displays the Ephesoft interface for processing a mortgage document. On the left, a sidebar titled 'Use Existing Field For Key:' allows configuration of the 'Apply KV' rule. The main area shows a document with fields for borrower information, mortgage details, and loan terms. Numbered callouts (1-10) point to specific fields: 1 points to the borrower's name and address; 2 points to the security address; 3 points to the commitment date; 4 points to the loan type and amount; 5 points to the basic loan amount; 6 points to the interest rate; 7 points to the deal number; 8 points to the advance date; 9 points to the maturity date; and 10 points to the interest rate. A table at the bottom, titled 'Advance KV Test', shows the results of the rule application for the security address field.

Key	Value	Confidence%	Key Coordinates	Value Coordinates	Colorcode
SECURITY ADDRESS	1212 STREET TORONTO ONTARIO	100	(162.072)(357.885)	(163.694)(344.728)	

10. Click **Apply KV** to apply the rule to the index field.

If you click **Cancel** without saving changes, the following confirmation message displays.

The confirmation dialog box is titled 'Confirmation' and contains a question mark icon. The text inside reads: 'There may be some unsaved changes. Do you want to save the changes?'. Below the text are two buttons: 'Save' and 'Discard'.

Click **Save** to save changes or click **Discard** to discard any configuration changes and navigate to the **KV Extraction Rule** screen.