Merging common values from separate fields

You have fields in your data that contain some commonalities. For example:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team A</td>
<td>12_357_p</td>
</tr>
<tr>
<td>Team B</td>
<td>12367_yzx</td>
</tr>
</tbody>
</table>

You want to create a third field that combines the common values in the existing fields.

You can accomplish this using a number of multivalue evaluation functions. The following search returns the following value: 1237.

```
| makeresults
| eval sourcefield="Team B: 12367_yzx Team A: 12_357_p"
| rex field=sourcefield "Team B: (?<f1>[^\s]+)\sTeam A: (?<f2>.*"
| eval f1split=split(f1, "")
| eval f2split=split(f2, "")
| eval f1split=mvfilter(f1split!="_")
| eval f2split=mvfilter(f2split!="_")
| eval f1matchingf2=mvmap(f1split, if(in(f2split, f1split), f1split, null()))
| eval flattened=mvjoin(f1matchingf2, "")
| fields _time, flattened
```

**Splunk Search**

| makeresults
| eval sourcefield="Team B: 12367_yzx Team A: 12_357_p"
| rex field=sourcefield "Team B: (?<f1>[^\s]+)\sTeam A: (?<f2>.*"
| eval f1split=split(f1, "")
| eval f2split=split(f2, "")

**Explanation**

Create a temporary, dummy event.

Create a field that contains the data you want to merge.

Extract team data into distinct fields (called \texttt{f1} and \texttt{f2}).

This \texttt{rex} command creates 2 fields from 1. If you have 2 fields already in the data, omit this command.

Make multi-value fields (called \texttt{f1split} and \texttt{f2split}) for each target field.

The split function uses some delimiter, such as commas.

---

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.
### Splunk Search

| eval f1split=mvfilter(f1split!=""), f2split=mvfilter(f2split!=""), flattened=mvjoin(f1matchingf2, "") | Filter out the underscores. For each f1split field value, check to see if it's in f2split. If so, return the f1split value. Otherwise, do nothing. Flatten the multi-value field into a single string. The mvmap command takes a multi value field, runs eval functions against each item in the array, and returns the result of each function into the field on the left side of the equals, in this case f1matchingf2. Return only the time stamp and flattened field. |
| eval flattened=mvjoin(f1matchingf2, "") | Flatten the multi-value field into a single string. The mvjoin command is often is used with commas or hyphens or other padding, but in this example we want only a string of characters. |
| fields _time, flattened | |

### Explanation

- or dashes, to split a string into multiple values. If you pass in a blank string, as in this example, the function will return each character of the string individually. So if your string is 15 characters long, you will get a resulting field that has 15 individual elements.
- Filter out the underscores.
- For each f1split field value, check to see if it's in f2split. If so, return the f1split value. Otherwise, do nothing.
- The mvmap command takes a multi value field, runs eval functions against each item in the array, and returns the result of each function into the field on the left side of the equals, in this case f1matchingf2.
- Flatten the multi-value field into a single string.
- The mvjoin command is often is used with commas or hyphens or other padding, but in this example we want only a string of characters.

### Next steps

These additional Splunk resources might help you understand and implement these recommendations:

- Docs: [makeresults command](#)
- Docs: [Multivalue eval functions](#)