

The background features a repeating pattern of blue squares and circles. The squares are arranged in a grid, and the circles are placed at the intersections and within the squares, creating a complex geometric design. The colors range from light blue to dark blue.

The logo consists of a stylized 'O' with an orange dot inside, followed by 'DC' in blue, and '2026' in orange. The '20' and '26' are stacked vertically.

Omnis Developer Conference

# Text gezielt hervorheben

Maurin Wannags,  
Omnis Software  
Germany GmbH



# Inhalt

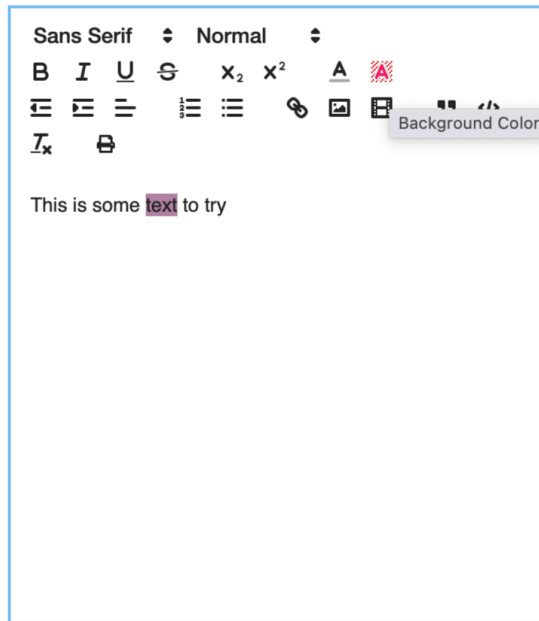
## **Text gezielt hervorheben**

- Text markieren in Omnis Studio
- Text dynamisch markieren
- Ansatz: JSON String Modulation
- Demonstration

# Text markieren in Omnis Studio

# Text gezielt hervorheben

## Text markieren in Omnis Studio

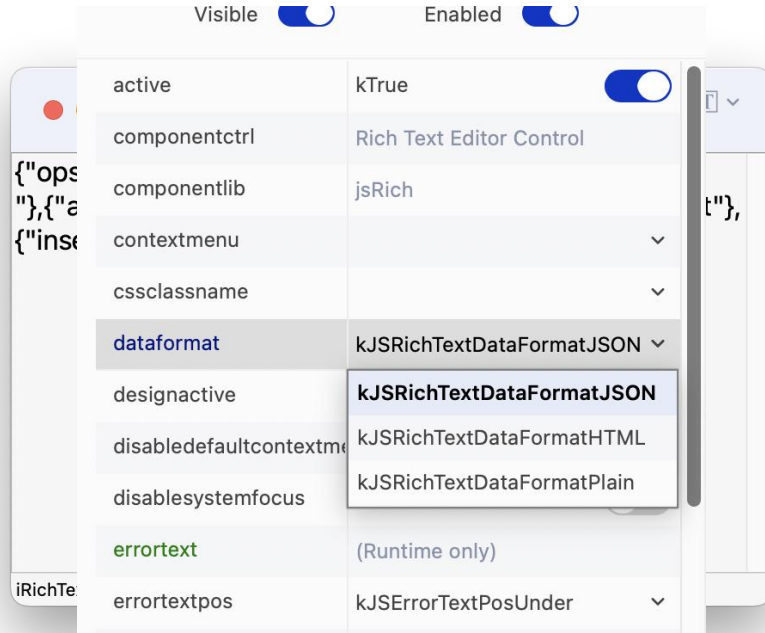


- Nicht direkt in normalen Feldern/Komponenten möglich
- Ausgenommen: der Rich Text Editor
  - *Hintergrundfarbe*

# Text dynamisch markieren

# Text gezielt hervorheben

## Text dynamisch markieren



- Um Formatierungen in der verknüpften Variable oder Row zu sehen, sollte das \$dateformat des Rich Text Editors auf [...] gesetzt sein:
  - *kJSRichTextDataFormatJSON*
  - *kJSRichTextDataFormatHTML*
- In diesem Fall (JSON) wird die Markierung mit einem "background attribute" realisiert.

# Text gezielt hervorheben

## Text dynamisch markieren

```
1  {
2  |
3  |   "ops": [
4  |     {
5  |       "insert": "This is some text to try\n"
6  |     }
7  |   ]
8  }
9
10
11
12
13
14
15
16
```

- Suchtext im JSON lokalisieren
- JSON-Object schließen
- Neues JSON-Object mit dem gefundenen Text und „background attribute“ einfügen
- JSON-Object wieder öffnen

# Text gezielt hervorheben

## Text dynamisch markieren

```
{
  "ops": [
    {
      "insert": "This is some "
    },
    {
      "attributes": {"background": "#e60000"},
      "insert": "text"
    },
    {
      "insert": " to try\n"
    }
  ]
}
```

- Entweder direkt die verknüpfte Variable bearbeiten („Field Reference“)
- Oder den JSON-String als Ergebnis in die verknüpfte Variable laden

# Ansatz: JSON String Modulation

# Text gezielt hervorheben

## Ansatz: JSON String Modulation

```
# here the magic begins ;-)
Begin text block
If pCaseSensitive
  Text:[left(pText,pos(pWord,pText)-1)]
Else
  Text:[left(pText,pos(low(pWord),low(pText))-1)]
End If
Text:},{ "attributes":{"background":
Text:[pColor]
Text:},"insert":
Text:[iDummy]
Text:},{ "insert":
If pCaseSensitive
  Text:[mid(pText,pos(pWord,pText)+len(pWord))]
Else
  Text:[mid(pText,pos(low(pWord),low(pText))+len(pWord))]
End If
End text block
Get text block pText
# note that the calling method needs to replace the dummy string back to the word
```

- Groß- und Kleinschreibung berücksichtigen (pCaseSensitive)
- Damit es nicht zu einer Endlosschleife kommt, da das Wort immer wieder gefunden werden würde, wird ein "Dummy-Wort" verwendet.

# Demonstration

Link zur

**Tech Note / Library:**

<https://www.omnis.net/developers/resources/technotes/tnjs0017.jsp>



**Vielen Dank**