

# %kdident

---

## Overview

The **%kdident** macro is designed to beautifully split character variables for output in tables and listings. It intelligently breaks long text strings into multiple lines while maintaining readability and formatting, using specified maximum line lengths and custom formatting characters.

## Version Information

- Version: 1.0
- Last Updated: 21OCT2021
- Author: Atorus Research

## Dependencies

- SAS Version: SAS 9.4 V9
- No macro dependencies

## Parameters

Parameter	Required	Default	Description
invar	Yes	-	Name of the input variable to be split
maxln	Yes	-	Maximum number of characters allowed per line
outvar	No	<b>&amp;invar._new</b>	Name of the output variable
tabchar	No	<b>\$ \$ \$</b>	Tabulation characters used to pad newlines
newlchar	No	<b>\$n</b>	Newline character
debug	No	N	Flag to retain temporary variables

## Return Values/Output

- Creates a new variable (**outvar**) containing:
  - Original text split into multiple lines
  - Lines padded with specified tabulation characters
  - Newline characters at break points
- Maximum length of output variable: 10,000 characters
- When debug=Y, retains temporary variables:
  - **\_\_tmp\_[invar]l1-\_\_tmp\_[invar]l1000**: Array storing newline positions

## Processing Details

1. Parameter validation:
  - Checks for required parameters

- Verifies parameter values are not empty
- Warns if input variable length exceeds 10,000

## 2. Text processing:

- Initializes output variable with first word
- Processes remaining words sequentially
- Calculates current line length
- Determines line break positions

## 3. Line breaking logic:

- Adds new line if next word exceeds maximum length
- Maintains word boundaries
- Applies tabulation at line breaks
- Preserves original spacing within limits

## Examples

```
/* Basic usage - split AEBODSYS with 17-character limit */
%kdident(aebodsys, 17);

/* Custom output variable name and no tabulation */
%kdident(lbcomm1, 32,
        outvar=col10,
        tabchar=%str());

/* Custom newline character and debug mode */
%kdident(cmtrt, 25,
        newlchar=%str(#),
        debug=Y);
```

## Common Issues and Solutions

### 1. Long Input Text

- Warning: "input variable length is more significant than 10000"
- Solution: Consider breaking input into smaller chunks

### 2. Missing Parameters

- Error: "[parameter] is required parameter and should not be NULL"
- Solution: Provide all required parameters

### 3. Formatting Issues

- Issue: Unexpected line breaks
- Solution: Adjust maxln parameter or tabchar spacing

## Notes and Limitations

### 1. Text processing:

- Maximum input length: 10,000 characters
- Preserves word boundaries
- Maintains original spacing where possible
- Uses space as word delimiter

### 2. Output formatting:

- Line breaks occur at word boundaries
- Tabulation applied after each break
- Consistent indentation for wrapped lines
- Original text preserved within constraints

### 3. Performance considerations:

- Creates temporary array variables
- Processes text word by word
- Memory usage proportional to text length

## See Also

- `%jdtf1style`: Table styling and formatting
- `%kutitles`: Title and footnote management
- `%kplotrtf`: RTF output formatting

## Change Log

### Version 1.0 (21OCT2021)

- Initial release
- Basic text splitting functionality
- Custom formatting options
- Debug mode implementation