

# **VTL specification**

# Summary

## Introduction

## User manual

## Reference manual

## String operators

## String concatenation

## EBNF

# Introduction

---

id: index-introduction  
title: Introduction  
sidebar\_label: Overview  
slug: /introduction  
custom\_edit\_url: null

---

TODO

## User manual

- - -

id: index-user-manual  
title: User manual  
sidebar\_label: Overview  
slug: /user-manual  
custom\_edit\_url: null

---

TODO

# Reference manual

- - -

id: index-reference-manual  
title: Reference manual  
sidebar\_label: Overview  
slug: /reference-manual  
custom\_edit\_url: null

---

# String operators

- - -

id: index-string-operators  
title: String operators  
sidebar\_label: Overview  
slug: /reference-manual/string-operators  
custom\_edit\_url: null

---

# String concatenation

---

id: index-string-concatenation

title: String concatenation

sidebar\_label: String concatenation

slug: /reference-manual/string-operators/string-concatenation

custom\_edit\_url: null

---

## Behaviour

The operator has the behaviour of the “Operators applicable on two Scalar Values or Data Sets or Data Set Components” (see the section “Typical behaviours of the ML Operators”).

## Syntax

op1 || op2

## Input parameters type

op1, op2: string or component `_+ }`

## Examples

```
str := "Hello" || " " || "world";  
ds_comp_concat := ds1[calc concat := str1 || " "  
|| str2];  
ds_concat := ds2#str1[rename str1 to concat] || "  
" || ds1#str2[rename str2 to concat];
```

## Results

ds1

**id str1 str2**

1	a	toto
2	b	titi
3	c	tata

ds2

**id str1**

1	Hello
2	Hi
3	Hey

ds\_comp\_concat

**id str1 str2 concat**

1	a	toto	a toto
2	b	titi	b titi
3	c	tata	c tata

ds\_concat

**id concat**

1	Hello toto
---	------------

- 2 Hi titi
- 3 Hey tata

# EBNF

---

id: index-ebnf  
title: EBNF  
sidebar\_label: Overview  
slug: /ebnf  
custom\_edit\_url: null

---

TODO