SDMX GC

BANCA D'ITALIA
EUROSISTEMA

# How Does It Work?

## LEARNING VTL BY EXAMPLE

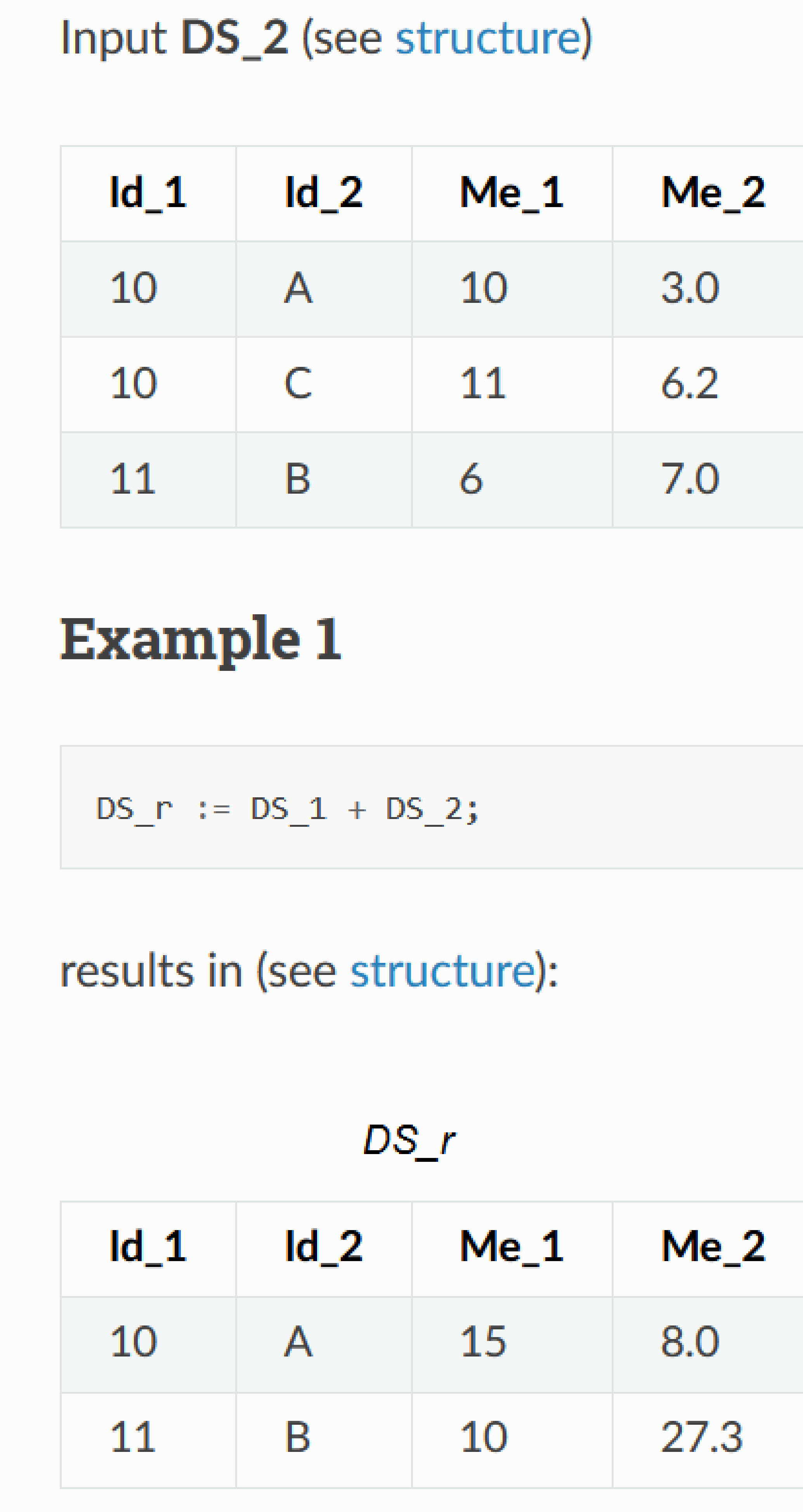
**Attilio Mattiocco**

**Valentino Pinna**

**Rome, Sept. 29 2025**

# The idea

- In 2024 the VTL Task Force worked to a complete revamp of the VTL documentation.
- Today the manuals are built from source files of various types and, more important to us, the operator examples are built from technical files (CSVs, JSON, vtl code).
- One step forward: leverage the examples source files to add a new feature to the VTL Engine & Editor



Input **DS_2** (see structure)

| Id_1 | Id_2 | Me_1 | Me_2 |
|---|---|---|---|
| 10 | A | 10 | 3.0 |
| 10 | C | 11 | 6.2 |
| 11 | B | 6 | 7.0 |

## Example 1

```
DS_r := DS_1 + DS_2;
```

results in (see structure):

*DS_r*

| Id_1 | Id_2 | Me_1 | Me_2 |
|---|---|---|---|
| 10 | A | 15 | 8.0 |
| 11 | B | 10 | 27.3 |

# The VTL E&E “*Demo Mode*”

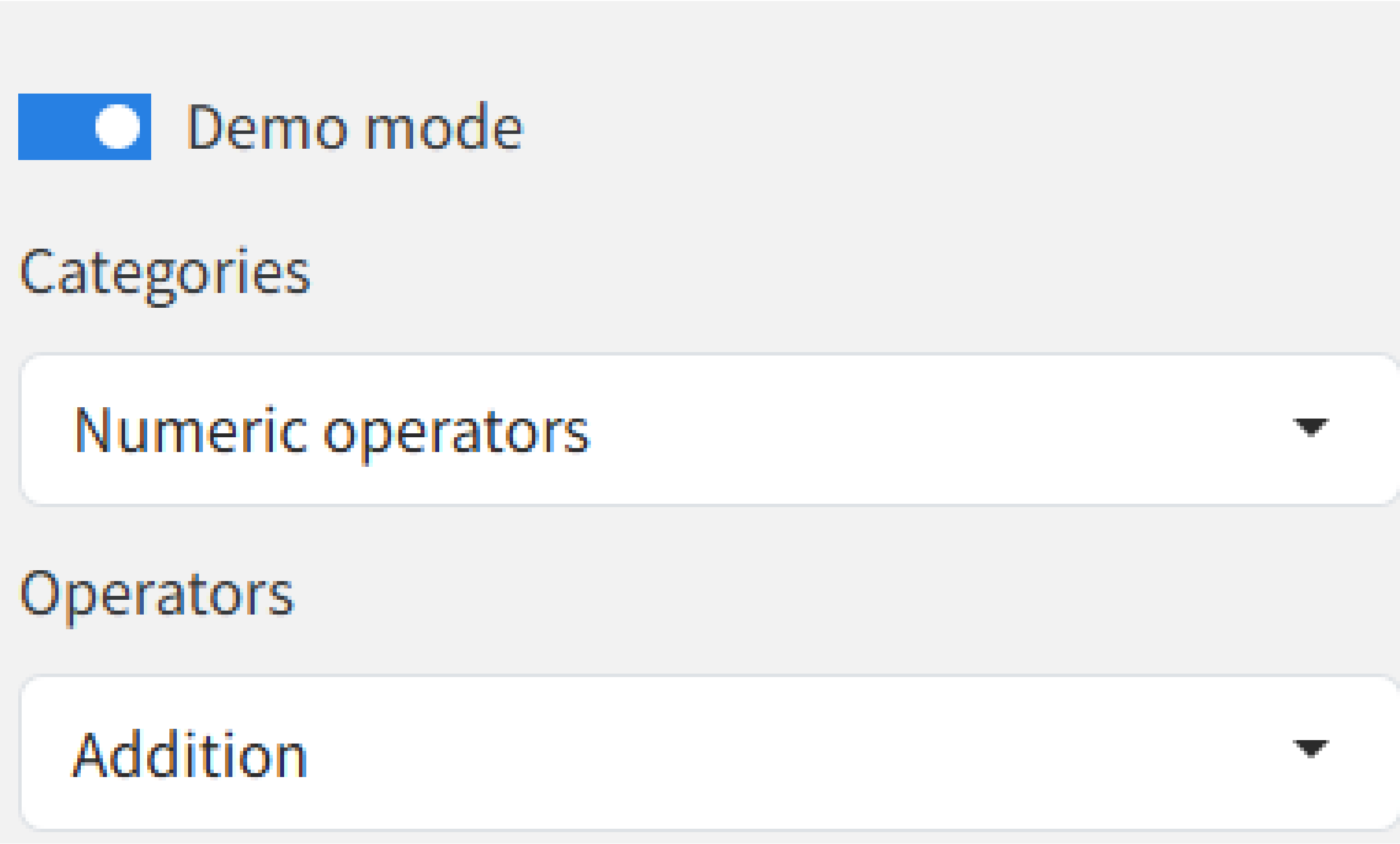
A new feature available in VTL Studio, the development environment of the VTL E&E

The code, data and structures of the examples in the official VTL manual are packed into the distribution for the “*Demo Mode*” feature.

All the examples are preloaded, along with all necessary data and metadata: VTL Studio becomes a **live, interactive VTL manual**

Users can **read the VTL code, compile, execute, check the results**

The demo mode **examples are not locked**: they can be modified, allowing users to experiment use cases that are not covered by the manual



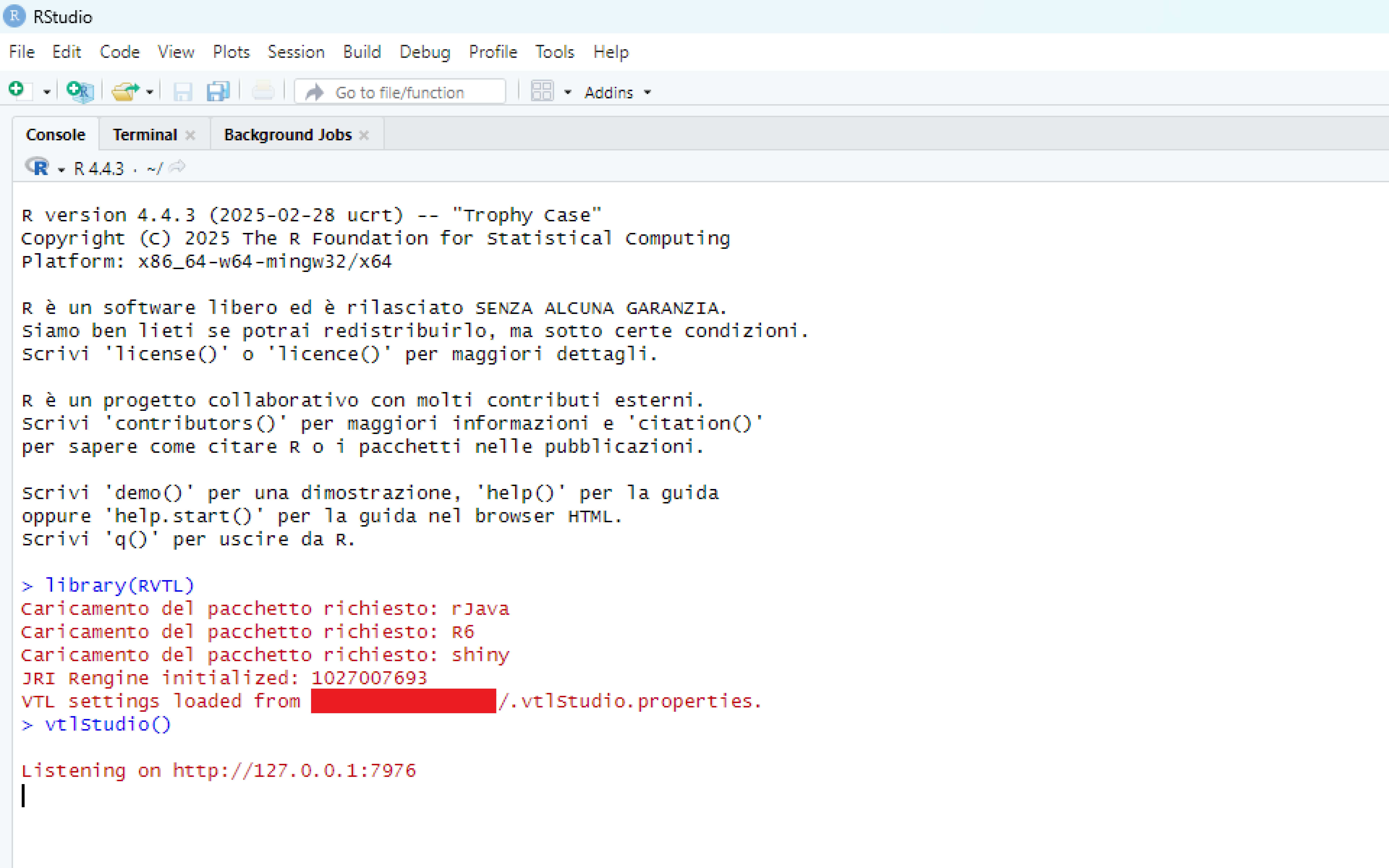
Demo mode

Categories

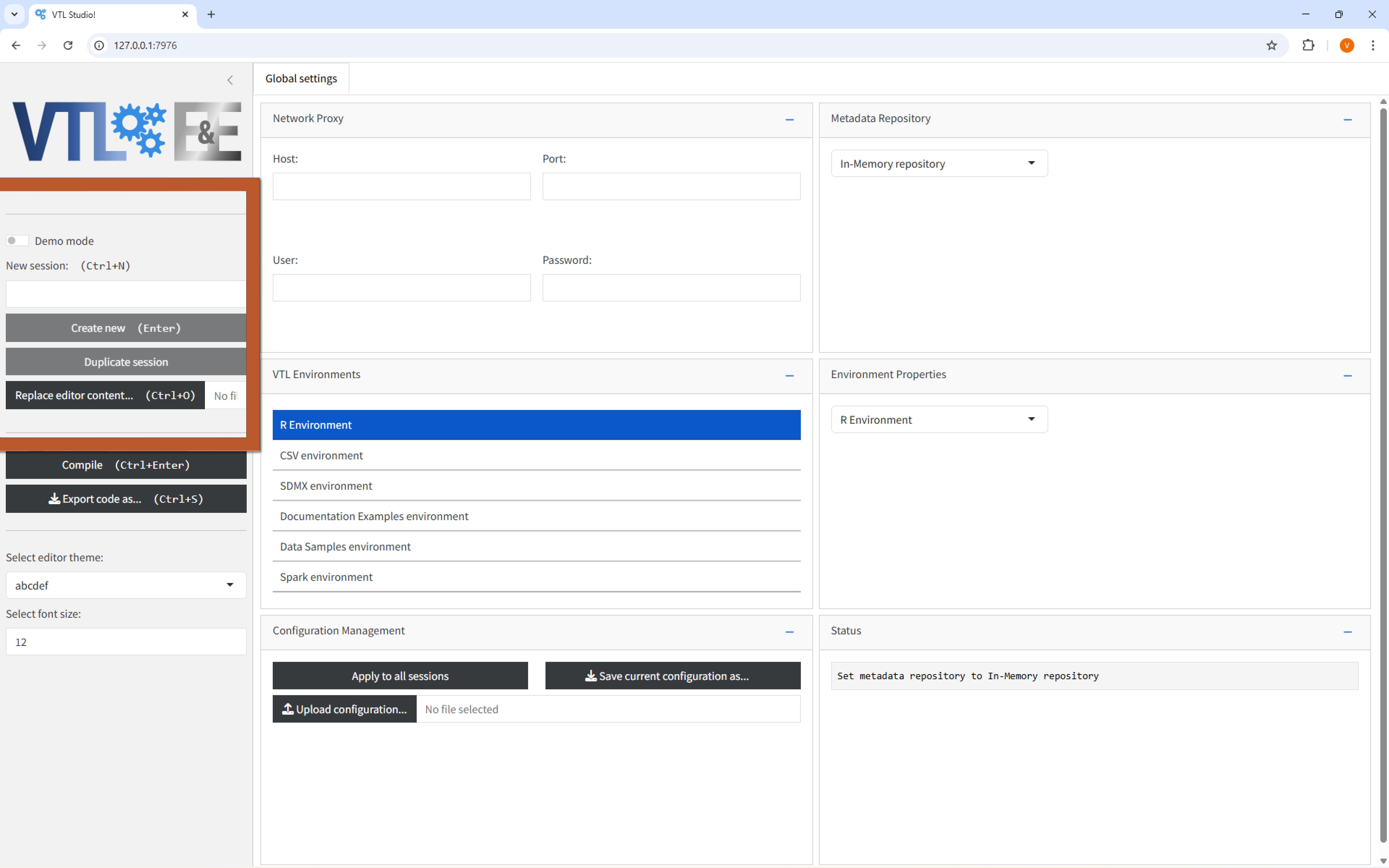
Numeric operators

Operators

Addition



```
R version 4.4.3 (2025-02-28 ucrt) -- "Trophy Case"
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64

R è un software libero ed è rilasciato SENZA ALCUNA GARANZIA.
Siamo ben lieti se potrai redistribuirlo, ma sotto certe condizioni.
Scrivi 'license()' o 'licence()' per maggiori dettagli.

R è un progetto collaborativo con molti contributi esterni.
Scrivi 'contributors()' per maggiori informazioni e 'citation()'
per sapere come citare R o i pacchetti nelle pubblicazioni.

Scrivi 'demo()' per una dimostrazione, 'help()' per la guida
oppure 'help.start()' per la guida nel browser HTML.
Scrivi 'q()' per uscire da R.

> library(RVTL)
Caricamento del pacchetto richiesto: rJava
Caricamento del pacchetto richiesto: R6
Caricamento del pacchetto richiesto: shiny
JRI Rengine initialized: 1027007693
VTL settings loaded from [redacted]/.vtlStudio.properties.
> vtlStudio()

Listening on http://127.0.0.1:7976
```



Metadata Repository: In-Memory repository

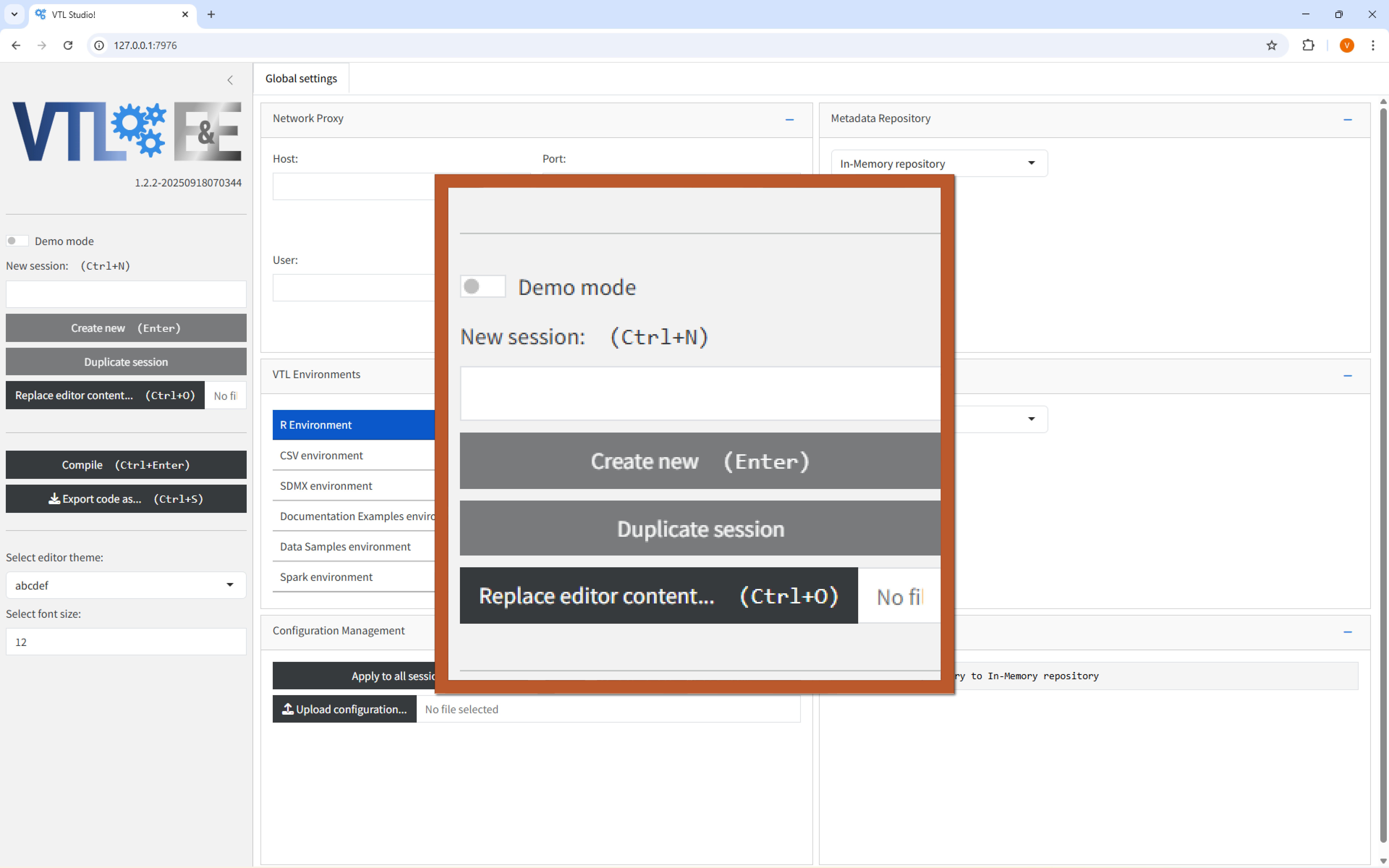
VTL Environments: R Environment

Environment Properties: R Environment

Select editor theme: abcdef

Select font size: 12

```
Set metadata repository to In-Memory repository
```

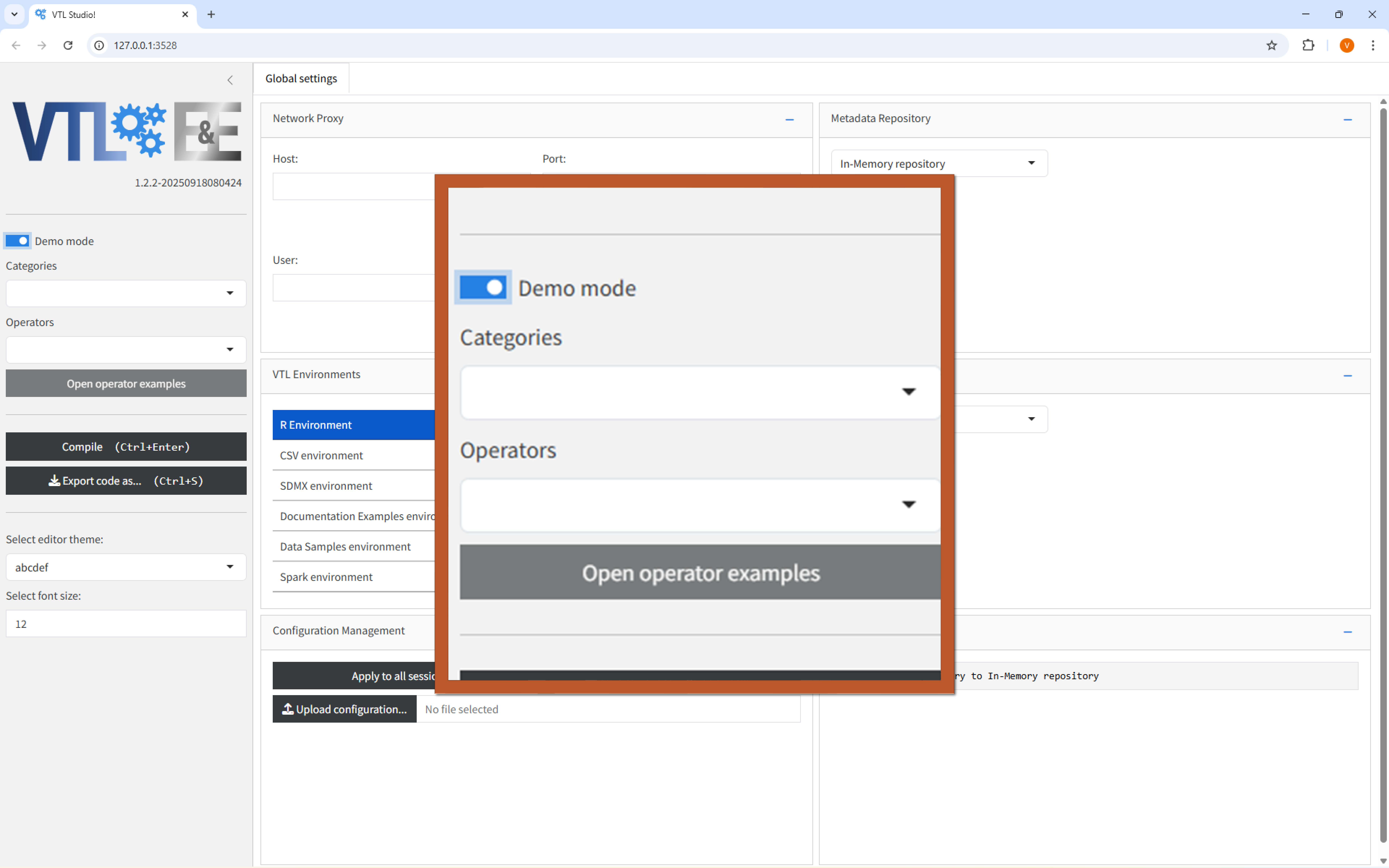


Select editor theme: abcdef

Select font size: 12

Metadata Repository: In-Memory repository

```
ry to In-Memory repository
```

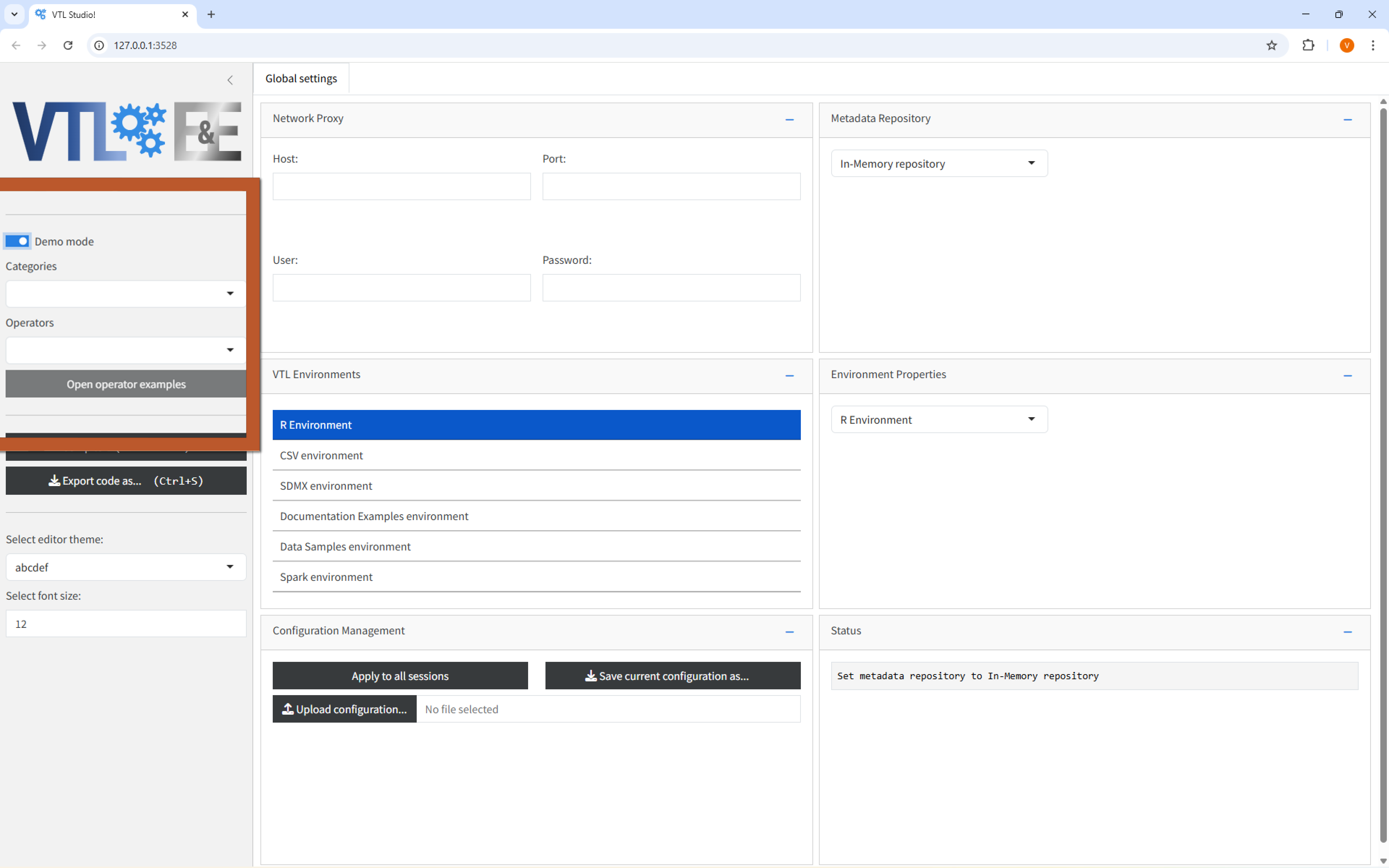


Select editor theme: abcdef

Select font size: 12

Metadata Repository: In-Memory repository

```
ry to In-Memory repository
```



Demo mode

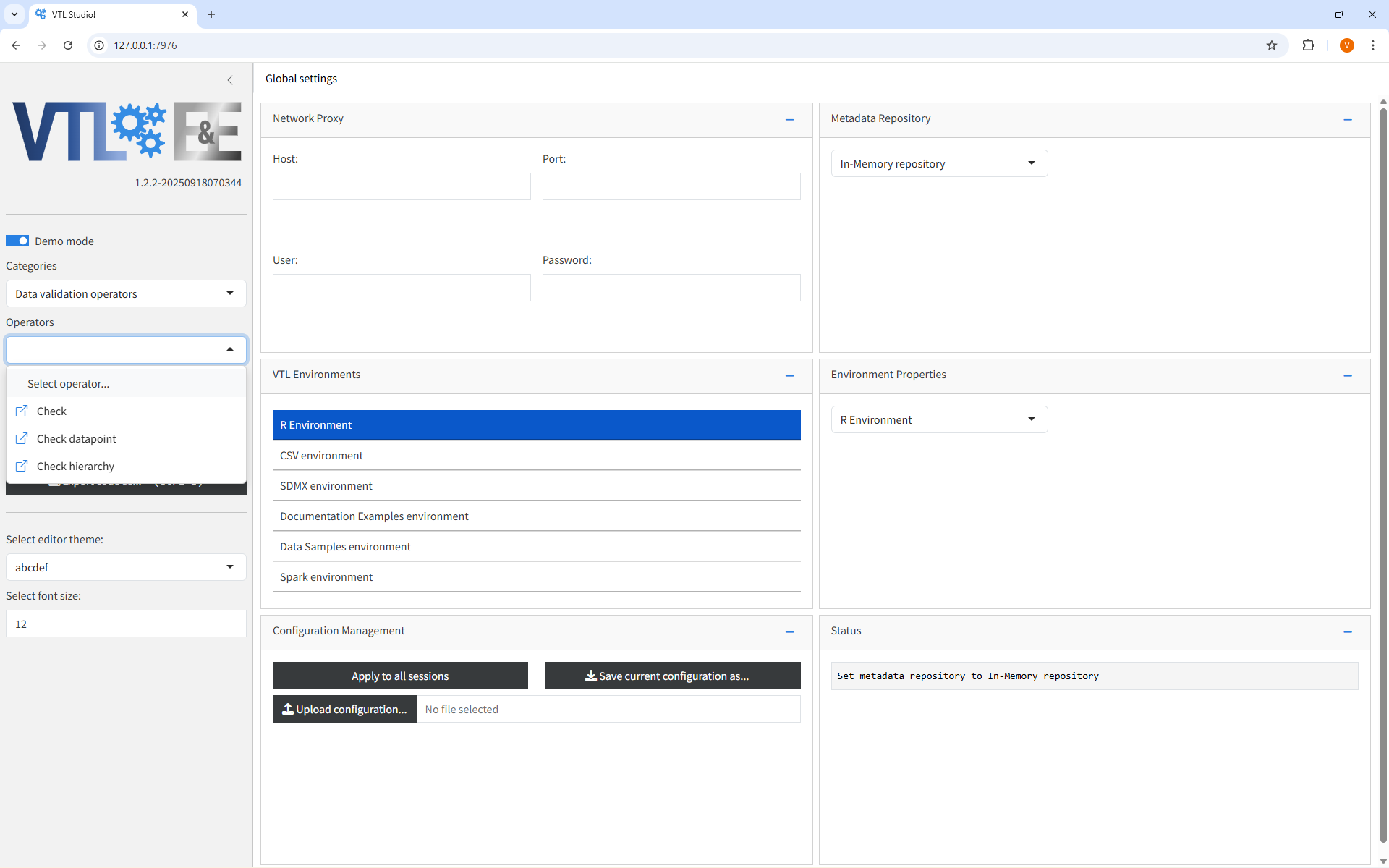
Metadata Repository: In-Memory repository

Environment Properties: R Environment

Select editor theme: abcdef

Select font size: 12

```
Set metadata repository to In-Memory repository
```



1.2.2-20250918070344

Demo mode

Categories: Data validation operators

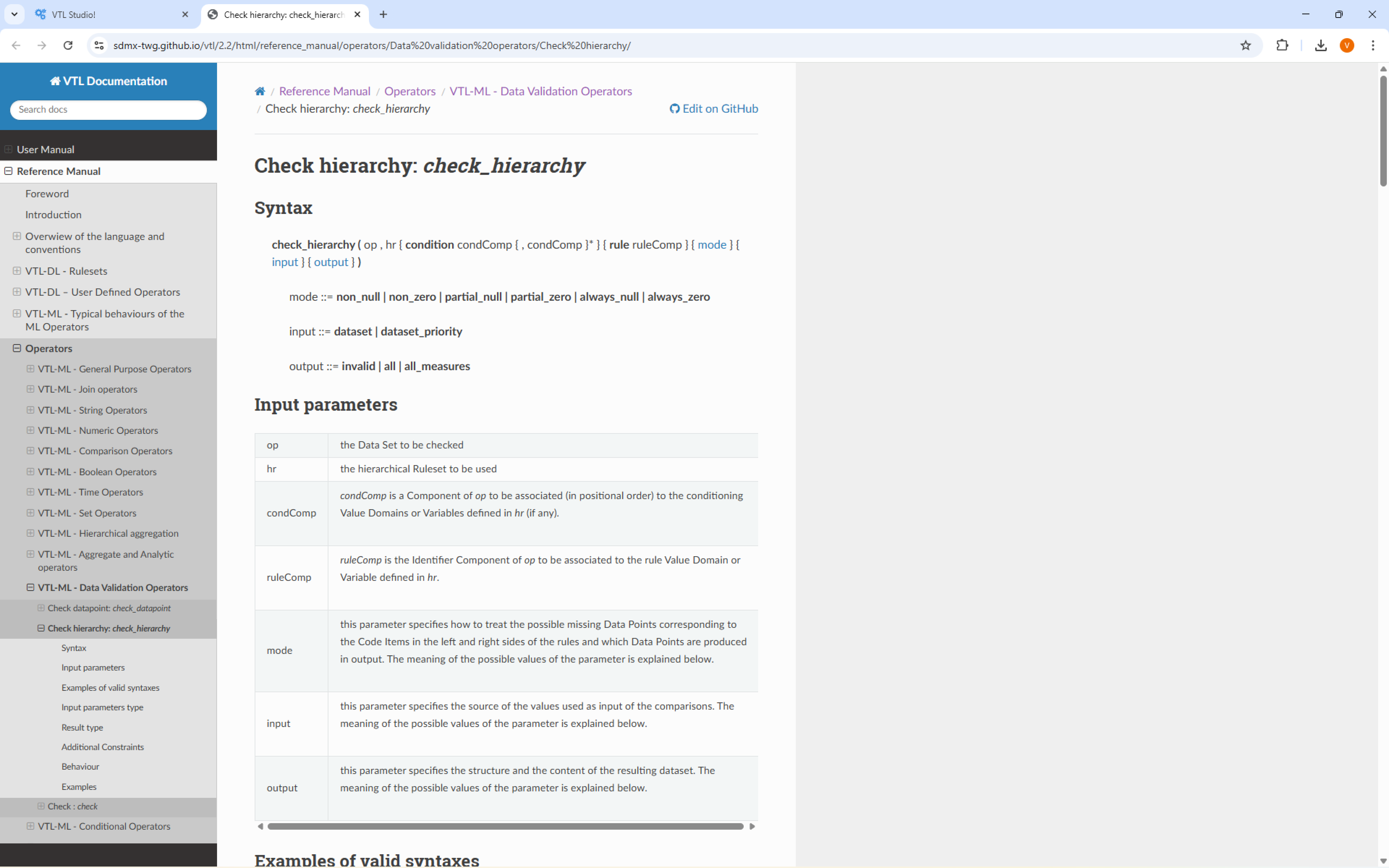
Metadata Repository: In-Memory repository

Environment Properties: R Environment

Select editor theme: abcdef

Select font size: 12

```
Set metadata repository to In-Memory repository
```


VTL Studio!
Check hierarchy: check_hierarch
sdmx-twg.github.io/vtl/2.2/html/reference_manual/operators/Data%20validation%20operators/Check%20hierarchy/
VTL Documentation
Search docs
User Manual
Reference Manual
Foreword
Introduction
Overwiew of the language and conventions
VTL-DL - Rulesets
VTL-DL – User Defined Operators
VTL-ML - Typical behaviours of the ML Operators
Operators
VTL-ML - General Purpose Operators
VTL-ML - Join operators
VTL-ML - String Operators
VTL-ML - Numeric Operators
VTL-ML - Comparison Operators
VTL-ML - Boolean Operators
VTL-ML - Time Operators
VTL-ML - Set Operators
VTL-ML - Hierarchical aggregation
VTL-ML - Aggregate and Analytic operators
VTL-ML - Data Validation Operators
Check datapoint: check_datapoint
Check hierarchy: check_hierarchy
Syntax
Input parameters
Examples of valid syntaxes
Input parameters type
Result type
Additional Constraints
Behaviour
Examples
Check : check
VTL-ML - Conditional Operators
Reference Manual / Operators / VTL-ML - Data Validation Operators / Check hierarchy: check_hierarchy
Edit on GitHub

# Check hierarchy: *check_hierarchy*

## Syntax

**check_hierarchy (** op , hr { **condition** condComp { , condComp }* } { **rule** ruleComp } { mode } { input } { output } **)**

mode ::= **non_null | non_zero | partial_null | partial_zero | always_null | always_zero**

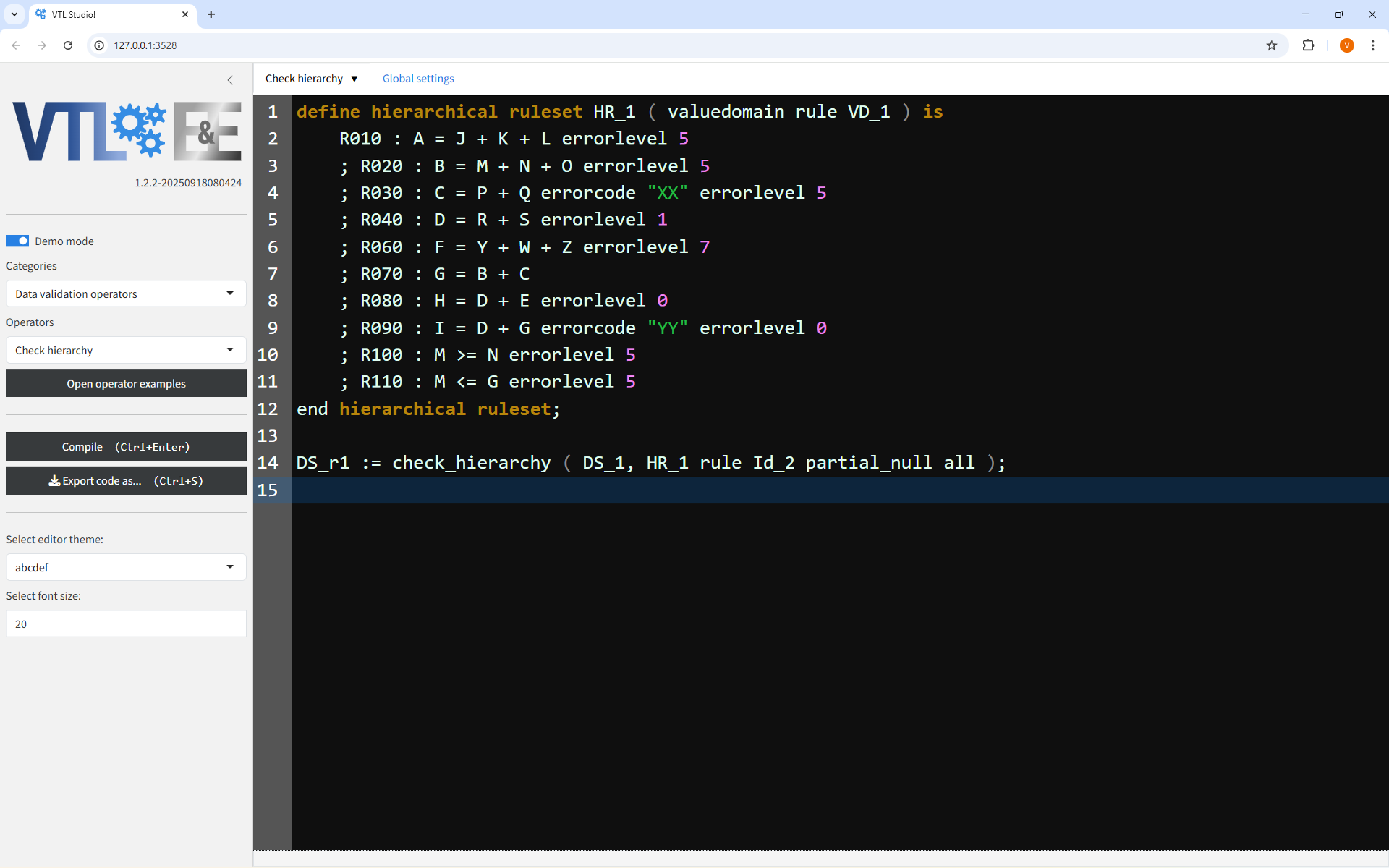
input ::= **dataset | dataset_priority**

output ::= **invalid | all | all_measures**

## Input parameters

| | |
|---|---|
| op | the Data Set to be checked |
| hr | the hierarchical Ruleset to be used |
| condComp | *condComp* is a Component of *op* to be associated (in positional order) to the conditioning Value Domains or Variables defined in *hr* (if any). |
| ruleComp | *ruleComp* is the Identifier Component of *op* to be associated to the rule Value Domain or Variable defined in *hr*. |
| mode | this parameter specifies how to treat the possible missing Data Points corresponding to the Code Items in the left and right sides of the rules and which Data Points are produced in output. The meaning of the possible values of the parameter is explained below. |
| input | this parameter specifies the source of the values used as input of the comparisons. The meaning of the possible values of the parameter is explained below. |
| output | this parameter specifies the structure and the content of the resulting dataset. The meaning of the possible values of the parameter is explained below. |

## Examples of valid syntaxes



1.2.2-20250918080424

Demo mode

Categories

Data validation operators

Operators

Check hierarchy

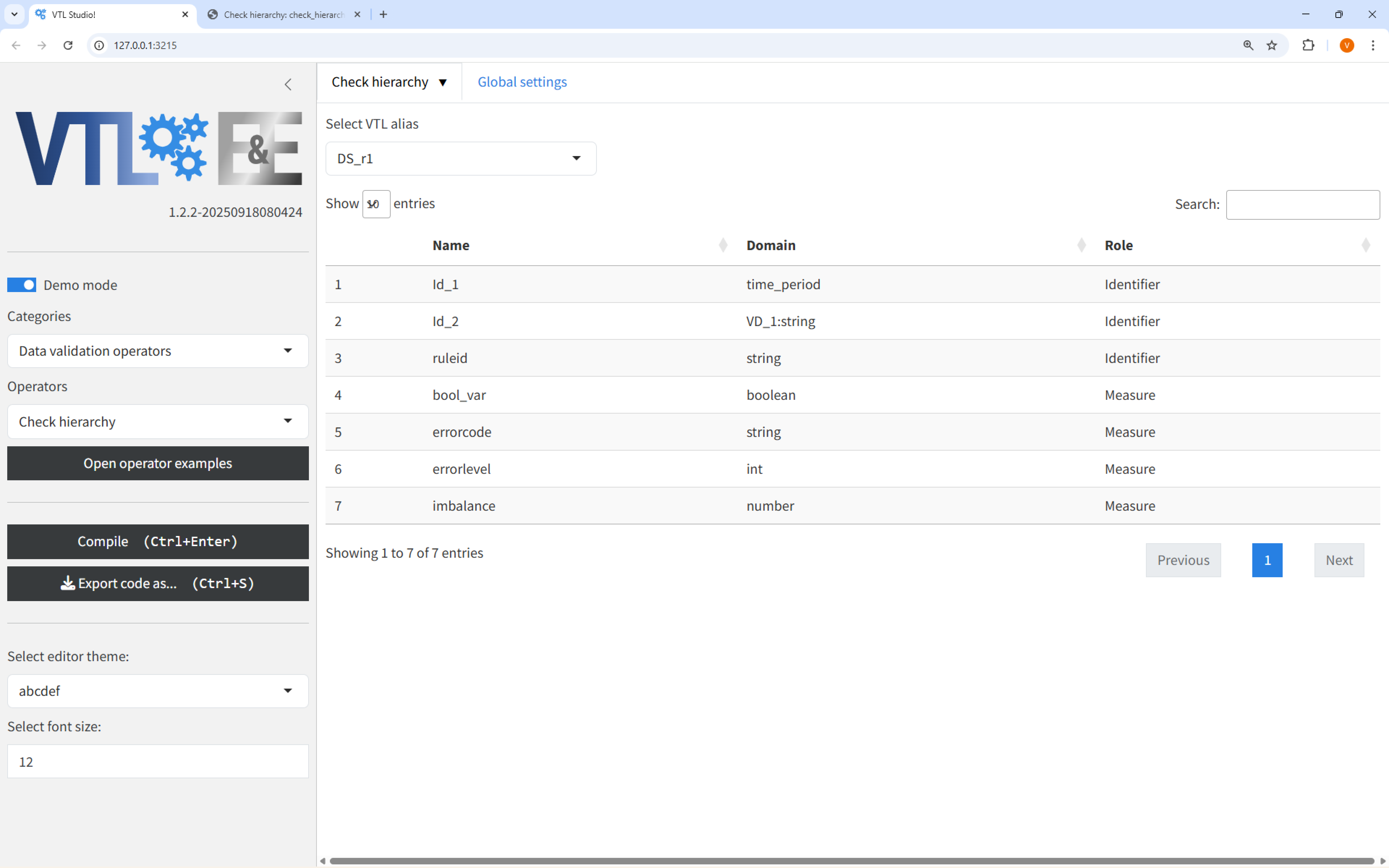
Select editor theme:

abcdef

Select font size:

20

```
define hierarchical ruleset HR_1 ( valuedomain rule VD_1 ) is
    R010 : A = J + K + L errorlevel 5
    ; R020 : B = M + N + O errorlevel 5
    ; R030 : C = P + Q errorcode "XX" errorlevel 5
    ; R040 : D = R + S errorlevel 1
    ; R060 : F = Y + W + Z errorlevel 7
    ; R070 : G = B + C
    ; R080 : H = D + E errorlevel 0
    ; R090 : I = D + G errorcode "YY" errorlevel 0
    ; R100 : M >= N errorlevel 5
    ; R110 : M <= G errorlevel 5
end hierarchical ruleset;

DS_r1 := check_hierarchy ( DS_1, HR_1 rule Id_2 partial_null all );
```



1.2.2-20250918080424

Demo mode

Categories: Data validation operators

Operators: Check hierarchy

Select editor theme: abcdef

Select font size: 12

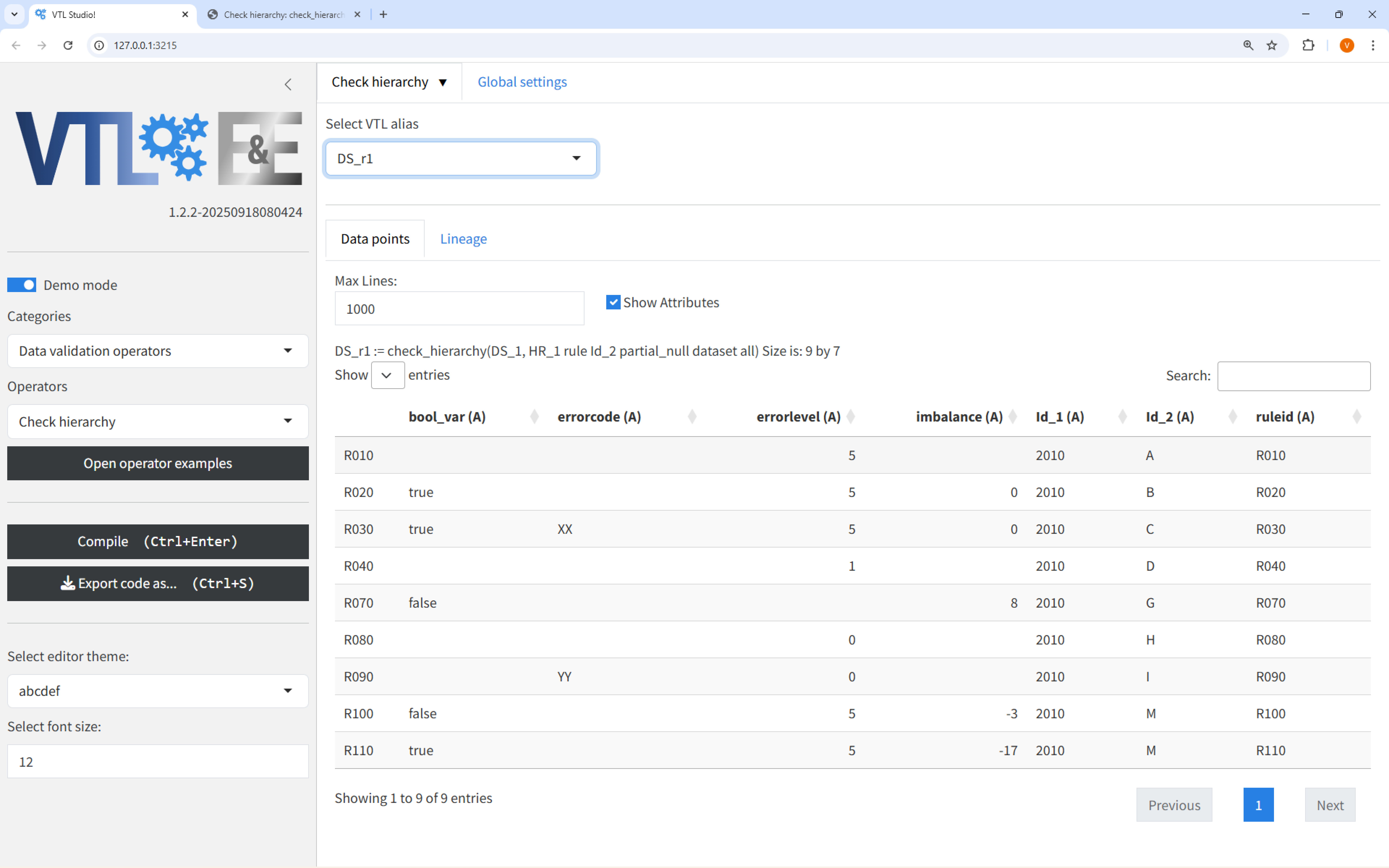
Select VTL alias: DS_r1

Show 10 entries

Search:

| | Name | Domain | Role |
|---|---|---|---|
| 1 | Id_1 | time_period | Identifier |
| 2 | Id_2 | VD_1:string | Identifier |
| 3 | ruleid | string | Identifier |
| 4 | bool_var | boolean | Measure |
| 5 | errorcode | string | Measure |
| 6 | errorlevel | int | Measure |
| 7 | imbalance | number | Measure |

Showing 1 to 7 of 7 entries



1.2.2-20250918080424

Demo mode

Categories: Data validation operators

Operators: Check hierarchy

Select editor theme: abcdef

Select font size: 12

Select VTL alias: DS_r1

Max Lines: 1000

☑ Show Attributes

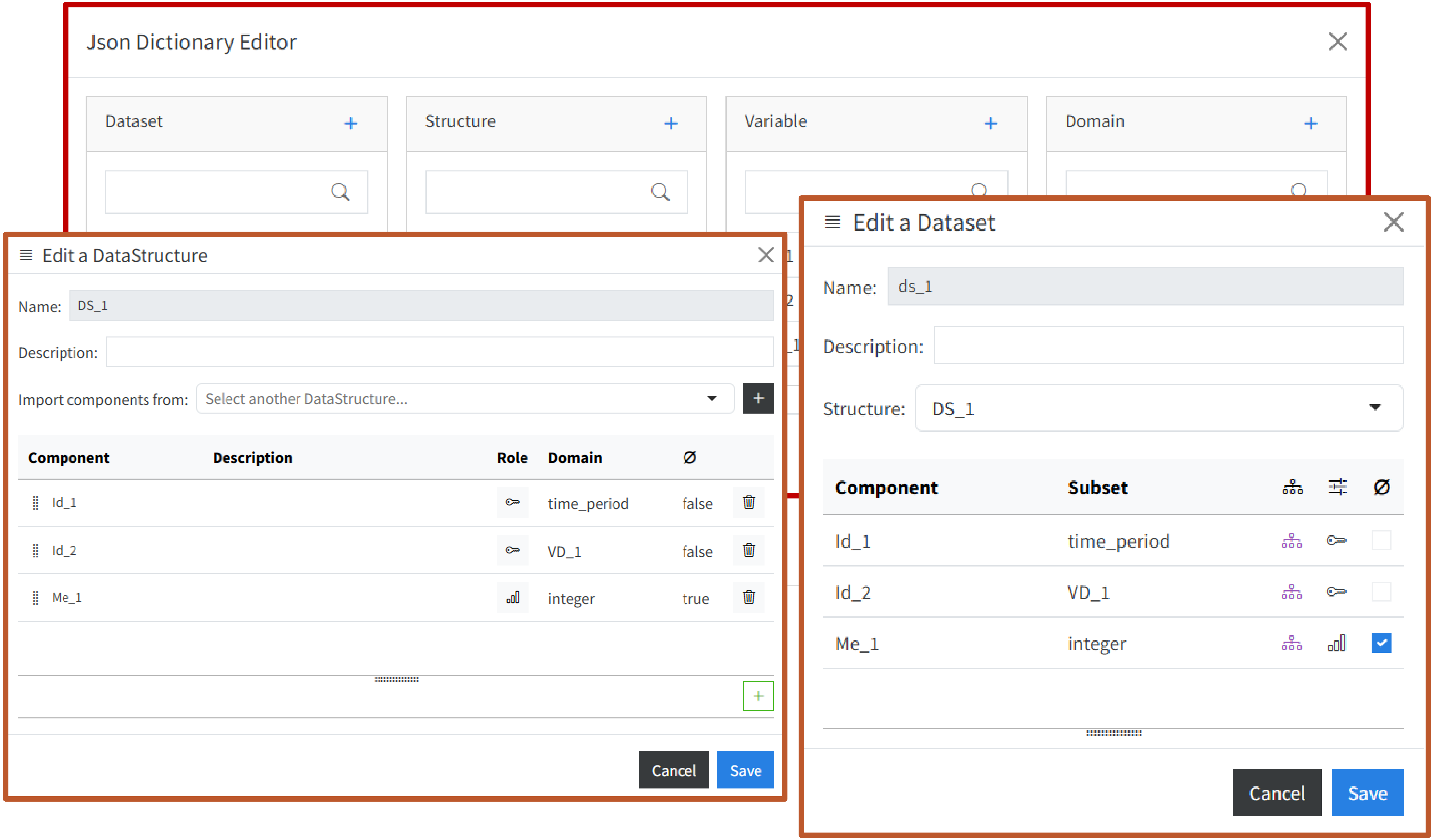
DS_r1 := check_hierarchy(DS_1, HR_1 rule Id_2 partial_null dataset all) Size is: 9 by 7

Show entries

Search:

| | bool_var (A) | errorcode (A) | errorlevel (A) | imbalance (A) | Id_1 (A) | Id_2 (A) | ruleid (A) |
|---|---|---|---|---|---|---|---|
| R010 | | | 5 | | 2010 | A | R010 |
| R020 | true | | 5 | 0 | 2010 | B | R020 |
| R030 | true | XX | 5 | 0 | 2010 | C | R030 |
| R040 | | | 1 | | 2010 | D | R040 |
| R070 | false | | | 8 | 2010 | G | R070 |
| R080 | | | 0 | | 2010 | H | R080 |
| R090 | | YY | 0 | | 2010 | I | R090 |
| R100 | false | | 5 | -3 | 2010 | M | R100 |
| R110 | true | | 5 | -17 | 2010 | M | R110 |

Showing 1 to 9 of 9 entries



Name: DS_1

Description:

Import components from: Select another DataStructure...

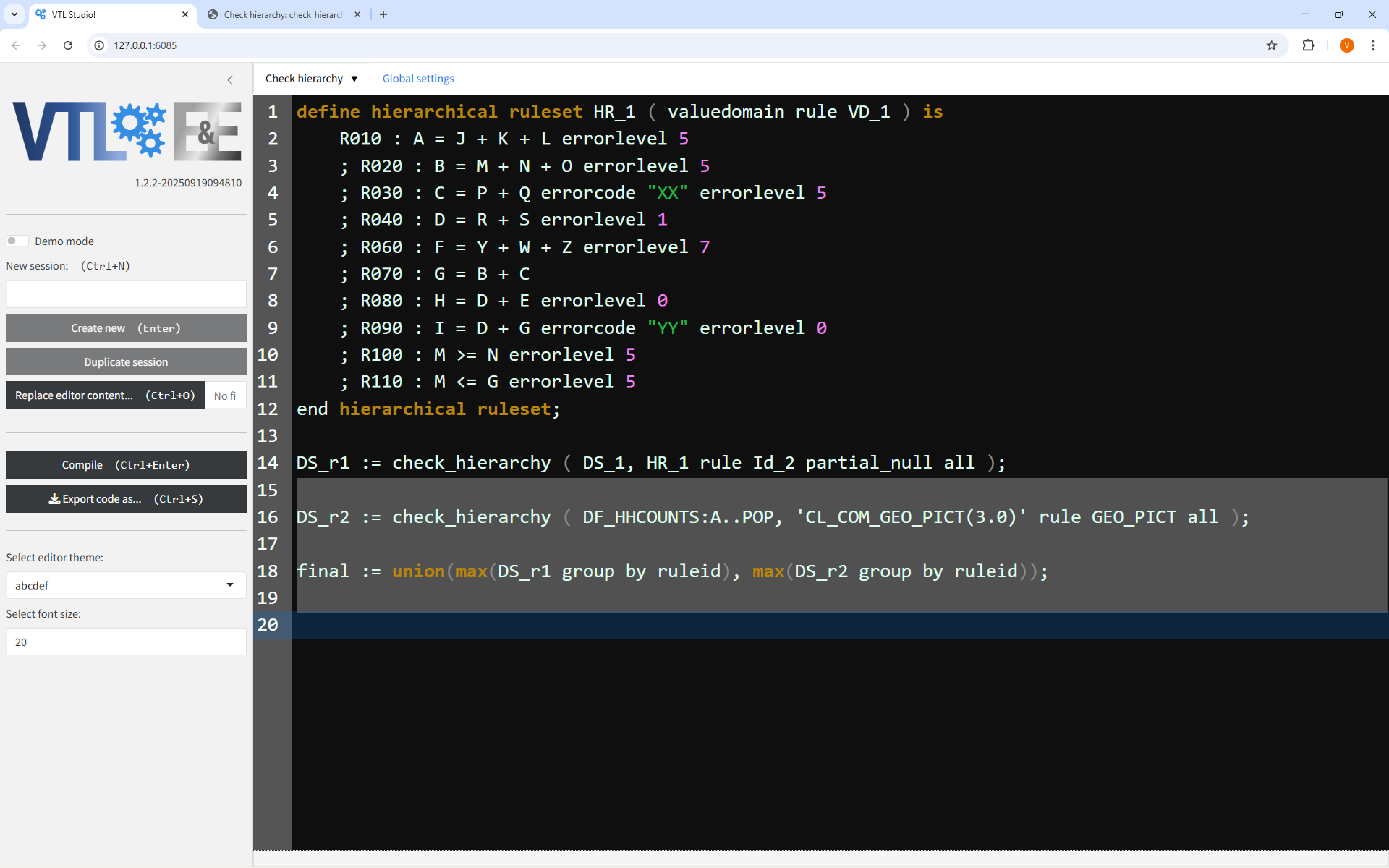
| Component | Description | Role | Domain | Ø |
|---|---|---|---|---|
| Id_1 | | 🔑 | time_period | false |
| Id_2 | | 🔑 | VD_1 | false |
| Me_1 | | 📊 | integer | true |

Name: ds_1

Description:

Structure: DS_1

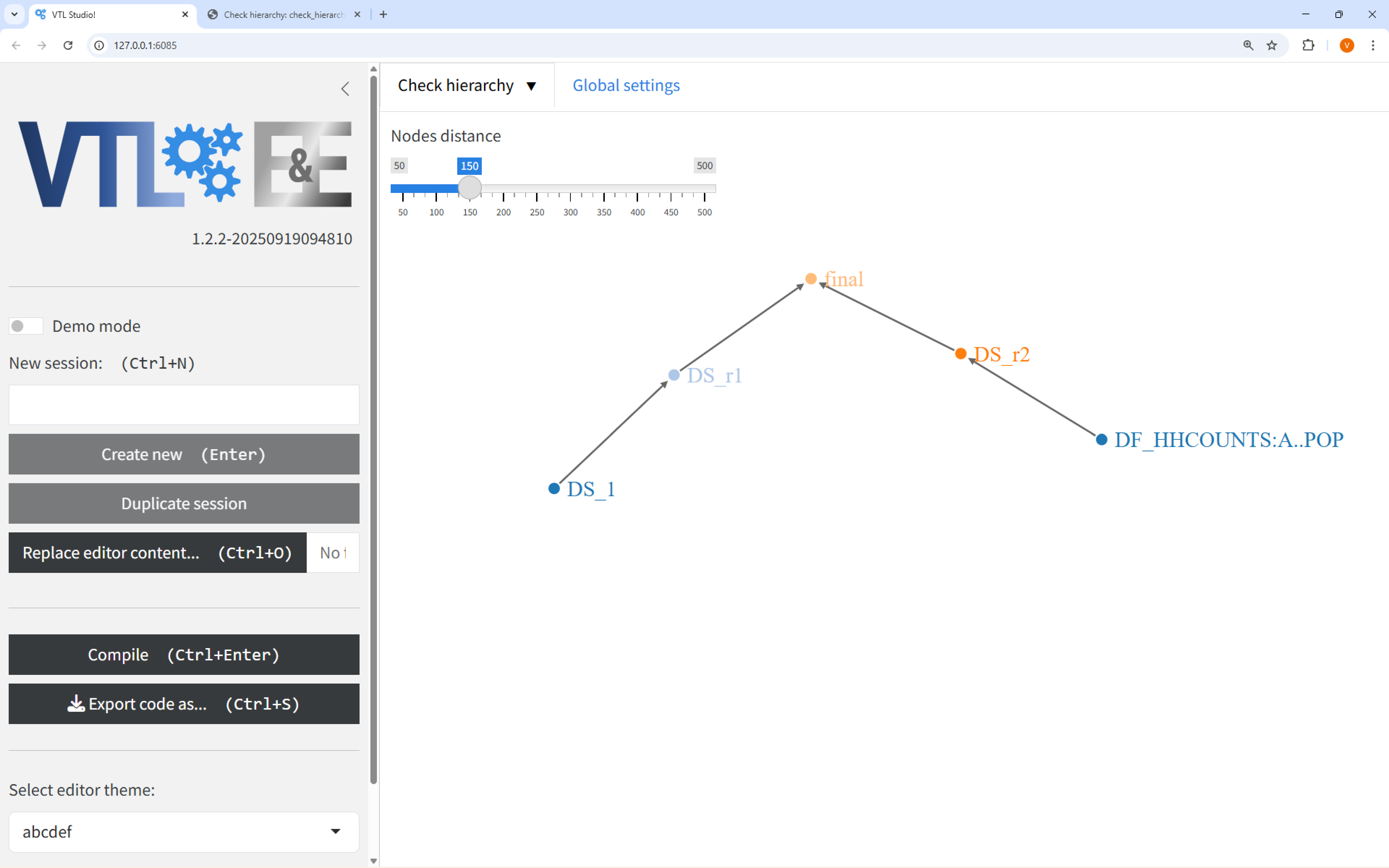
| Component | Subset | | | Ø |
|---|---|---|---|---|
| Id_1 | time_period | | 🔑 | ☐ |
| Id_2 | VD_1 | | 🔑 | ☐ |
| Me_1 | integer | | 📊 | ☑ |



Select editor theme: abcdef

Select font size: 20

```
define hierarchical ruleset HR_1 ( valuedomain rule VD_1 ) is
    R010 : A = J + K + L errorlevel 5
    ; R020 : B = M + N + O errorlevel 5
    ; R030 : C = P + Q errorcode "XX" errorlevel 5
    ; R040 : D = R + S errorlevel 1
    ; R060 : F = Y + W + Z errorlevel 7
    ; R070 : G = B + C
    ; R080 : H = D + E errorlevel 0
    ; R090 : I = D + G errorcode "YY" errorlevel 0
    ; R100 : M >= N errorlevel 5
    ; R110 : M <= G errorlevel 5
end hierarchical ruleset;

DS_r1 := check_hierarchy ( DS_1, HR_1 rule Id_2 partial_null all );

DS_r2 := check_hierarchy ( DF_HHCOUNTS:A..POP, 'CL_COM_GEO_PICT(3.0)' rule GEO_PICT all );

final := union(max(DS_r1 group by ruleid), max(DS_r2 group by ruleid));

```



1.2.2-20250919094810

New session: (Ctrl+N)

Nodes distance: 150 (50–500)

Select editor theme: abcdef

# Thanks