IBM ILOG CPLEX Optimization Studio SPSS Connector in CPLEX Studio

Version 12 Release 6

Copyright notice

Describes general use restrictions and trademarks related to this document and the software described in this document.

Trademarks

Contents

SPSS Connector in CPLEX Studio

The SPSS Connector connects IBM SPSS Modeler to IBM ILOG CPLEX Studio and enables you to read data from an SPSS stream into an OPL tuple set. The SPSS Connector is available on Windows 32-bit and 64-bit platforms.

Chapter 1. Prerequisites for using this tutorial

It is assumed that SPSS Modeler is installed and you know how to use it.

- IBM SPSS Modeler must be installed on the same machine as the CPLEX Studio IDE.
- You must have a license for SPSS Modeler. A trial license is included with the installation and lasts for 90 days.
- This tutorial assumes you have a working knowledge of SPSS Modeler.

  An overview of SPSS Modeler can be viewed here: http://www-01.ibm.com/software/analytics/spss/products/modeler/.

  Demos of SPSS Modeler can be viewed here: http://www-01.ibm.com/software/analytics/spss/downloads/demos.html.

Chapter 2. Preparatory steps

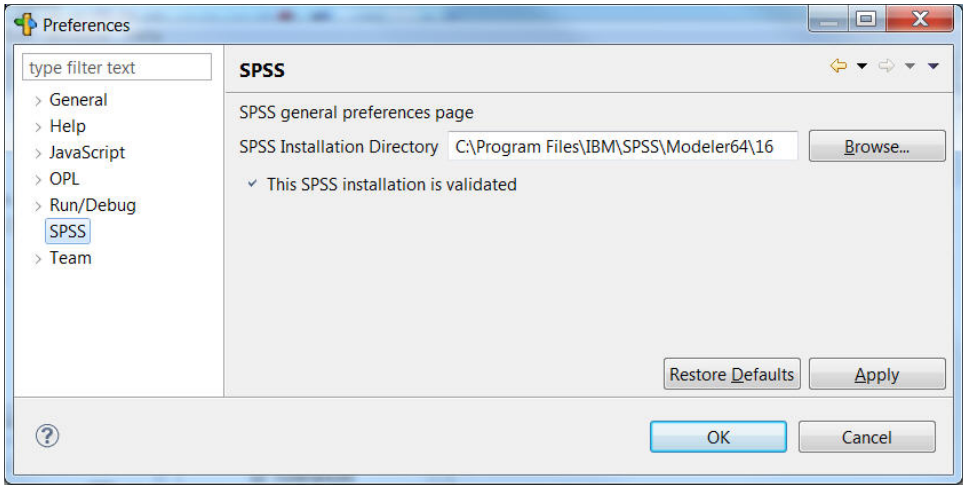
Before you can work with an SPSS stream in CPLEX Studio, you need to take some preparatory steps.

- “Enabling SPSS Modeler” in CPLEX Studio
- “Setting the SPSS Modeler demos environment variable” on page 4
- “Setting the SPSS Modeler environment variable” on page 4
- “Importing an SPSS data stream into the IDE” on page 5
- “Adapting the SPSS stream for CPLEX Studio” on page 5

Enabling SPSS Modeler

1. Launch the CPLEX Studio IDE.
2. Ensure that the IDE preferences indicate the installation path for SPSS Modeler.

   Click **Window** > **Preferences** and select **SPSS**. The installation path below shows the 64-bit version of SPSS Modeler.

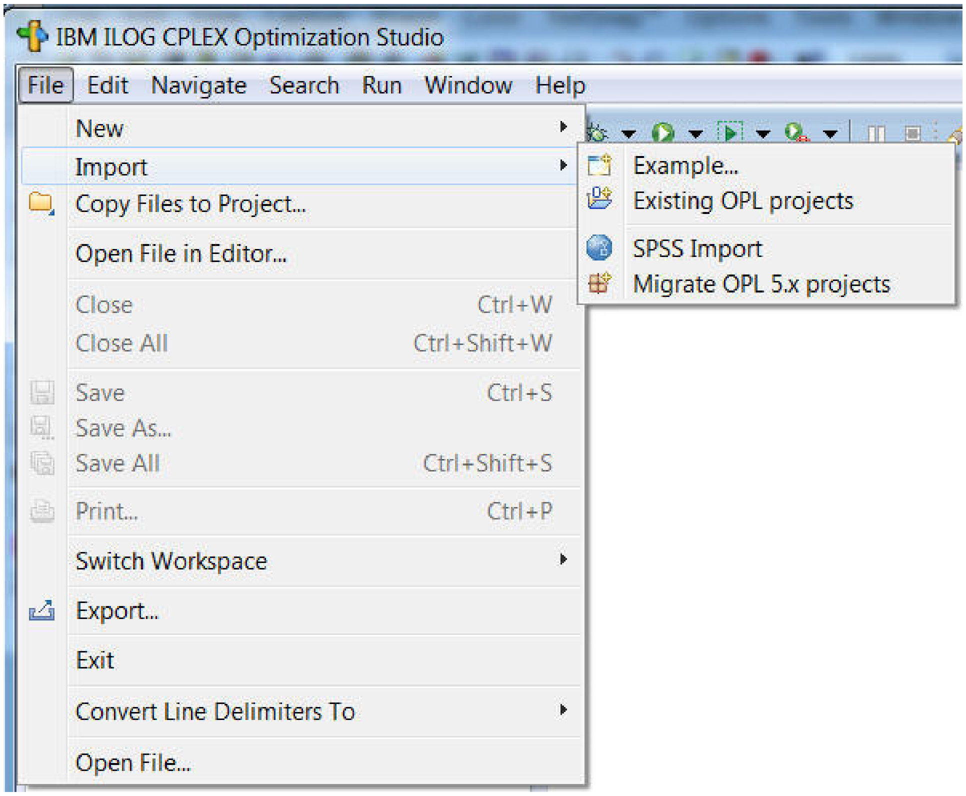


SPSS general preferences page

SPSS Installation Directory C:\Program Files\IBM\SPSS\Modeler64\16

✓ This SPSS installation is validated

3. You should see SPSS in the **Import** list. If you do not see the option, close the IDE and reopen it.



Setting the SPSS Modeler demos environment variable

In order to use the stream examples provided with SPSS Modeler connected to OPL, you must set the CLEO_DEMOS environment variable so that the data files referenced in these streams can be found. For example, on Windows 64-bit:

Start > **Control Panel** > **System** > **Advanced** > **Environment Variables** > **New**

Variable name: `CLEO_DEMOS`

Variable value: `C:\Program Files\IBM\SPSS\Modeler64\16\Demos`

You are advised to restart the IDE after setting the environment variable, to avoid problems when running the optimization model later in this tutorial.

Setting the SPSS Modeler environment variable

The OPL run time software needs to find a valid SPSS installation. If SPSS Modeler is not installed in the default folder (C:\Program Files\IBM\SPSS\Modeler64\16), you need to define the environment variable `OPL_SPSS_HOME` with the alternative installation path as its value.

If this environment variable is absent, OPL will search in the default SPSS installation location.

You are advised to restart the IDE after setting the environment variable, to avoid problems when running the optimization model later in this tutorial.

Importing an SPSS data stream into the IDE

The example stream we use in this tutorial is pm_selflearn.str, and can be found in the Demos directory of your SPSS Modeler installation. For example:

```
C:\Program Files\IBM\SPSS\Modeler64\16\Demos\streams\pm_selflearn.str
```

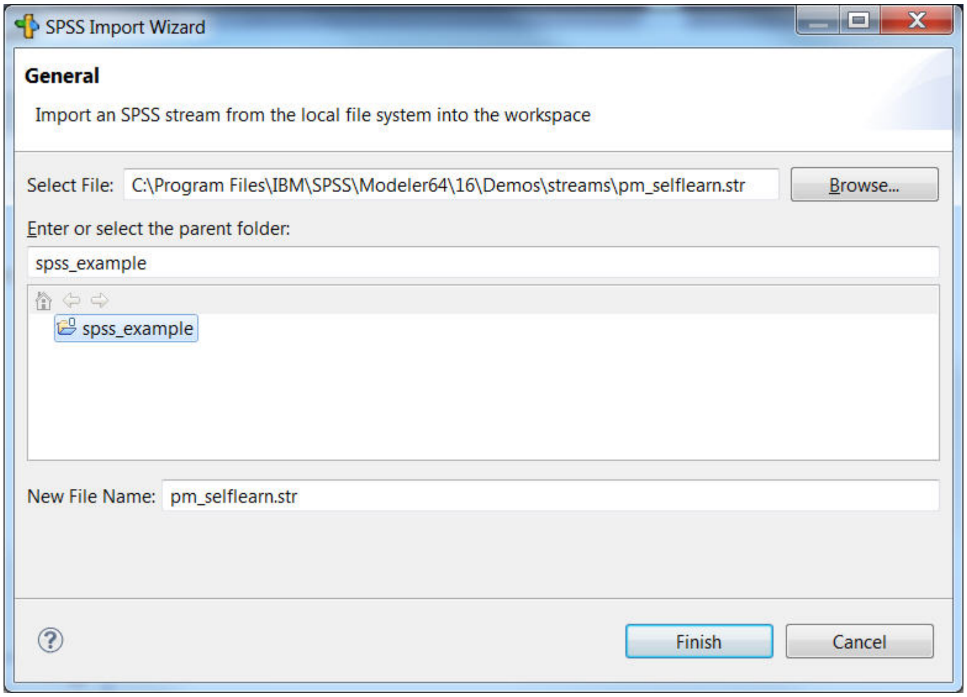
1.

   Create a new, empty OPL project in the IDE (**File** > **New** > **OPL Project**) and name it spss_example.

   Check the box **Add a default Run Configuration**. You will need a default run configuration later in this tutorial.

2.

   Right-click spss_example and import the SPSS stream pm_selflearn.str. (**File** > **Import** > **SPSS Import**).



SPSS Import Wizard

General

Import an SPSS stream from the local file system into the workspace

Select File: C:\Program Files\IBM\SPSS\Modeler64\16\Demos\streams\pm_selflearn.str

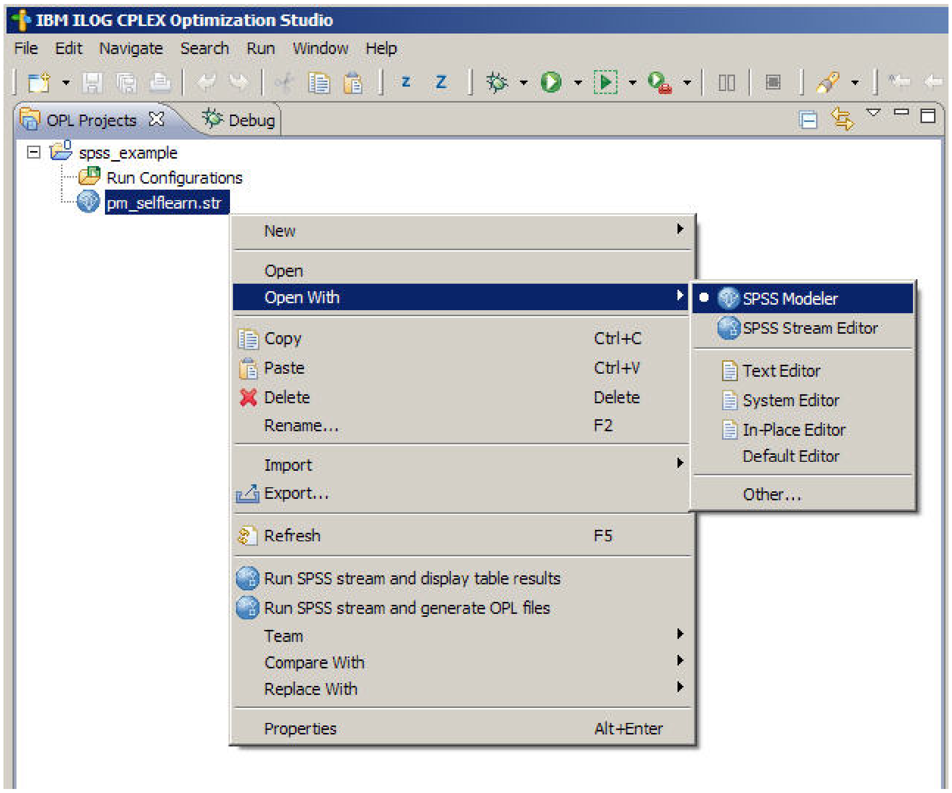
Enter or select the parent folder: spss_example

New File Name: pm_selflearn.str

Adapting the SPSS stream for CPLEX Studio

OPL cannot process SPSS terminal nodes; it processes only operation nodes such as Filter. You must, therefore, modify SPSS streams so that OPL can use them.

1. In the IDE, right-click the stream pm_selflearn.str and select **Open With** > **SPSS Modeler**.



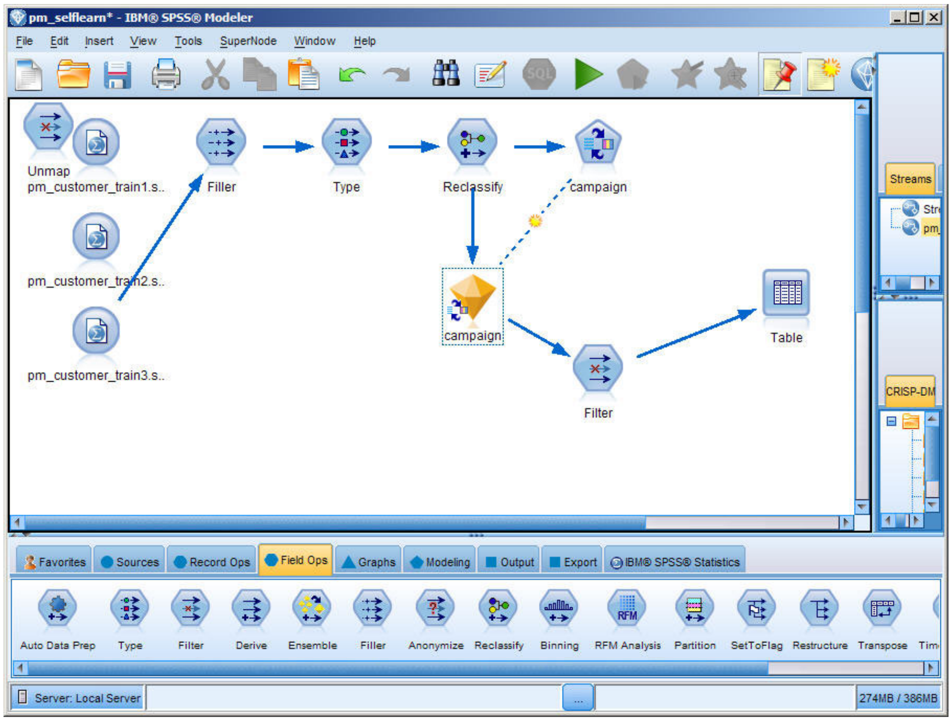
SPSS Modeler opens with the stream `pm_selflearn.str` in the editor.

2. If you run the SPSS stream as it is (run `pm_customer_train3.sav`), it will display a very big result table containing all possible fields. In order to display only the fields of interest for this example, you need to insert an operation node, such as a Filter node, between the campaign node and the result table, and edit the Filter.

   Drag the **Filter** node from the **Field Ops** tab into the editor.

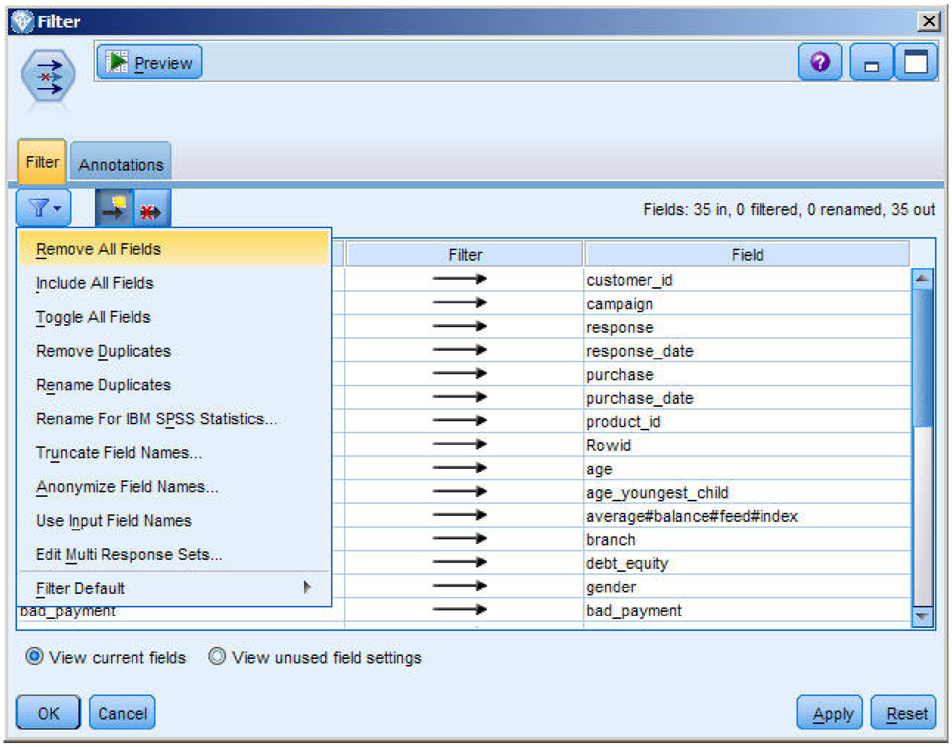
3. Insert the **Filter** node between the **campaign** node and the terminal node, **Table**. (Drag the connecting arrow **campaign --> Table** to **Filter**. The line will split into two connecting arrows, **campaign --> Filter** and **Filter --> Table**.)

   The stream now looks like this in the editor.



4. To edit the **Filter** node, right-click the node and select **Edit**, or double-click the node.

5. Click the **Filter options** button to display the list of options and select **Remove All Fields**.



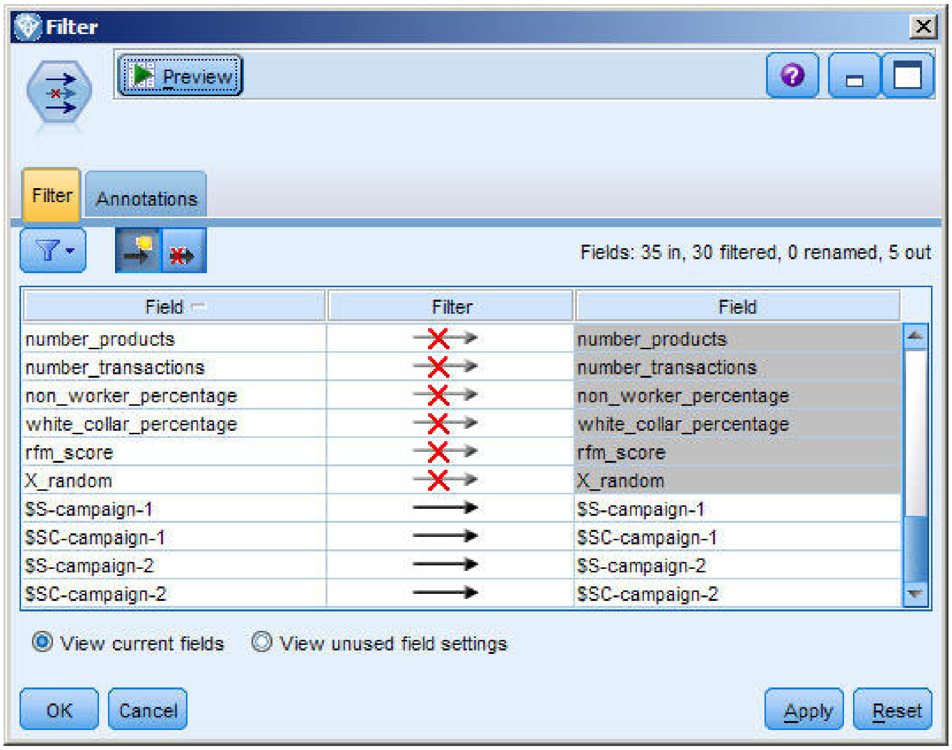
Fields: 35 in, 0 filtered, 0 renamed, 35 out

Filter	Field
→	customer_id
→	campaign
→	response
→	response_date
→	purchase
→	purchase_date
→	product_id
→	Rowid
→	age
→	age_youngest_child
→	average#balance#feed#index
→	branch
→	debt_equity
→	gender
→	bad_payment

The arrows are crossed out. Click a red cross to select each field you want to keep. For this example, you need the following five fields:

`customer_id`, `$S-campaign-1`, `$SC-campaign-1`, `$S-campaign-2`, `$SC-campaign-2`.

6. Here we see the last four fields, at the bottom of the table, selected.



Fields: 35 in, 30 filtered, 0 renamed, 5 out

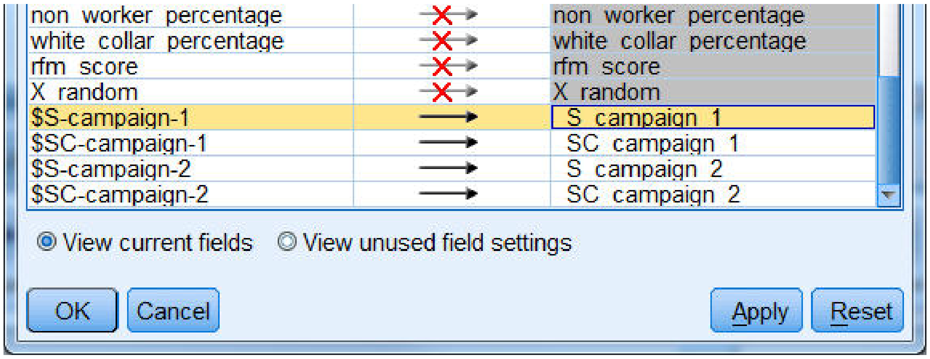
Field	Filter	Field
number_products	✗→	number_products
number_transactions	✗→	number_transactions
non_worker_percentage	✗→	non_worker_percentage
white_collar_percentage	✗→	white_collar_percentage
rfm_score	✗→	rfm_score
X_random	✗→	X_random
$S-campaign-1	→	$S-campaign-1
$SC-campaign-1	→	$SC-campaign-1
$S-campaign-2	→	$S-campaign-2
$SC-campaign-2	→	$SC-campaign-2

● View current fields ○ View unused field settings

Click **Apply** to save your changes.

7. (Optional step). OPL cannot process the characters $, &, #, %, -, ., or an empty string. When you work in CPLEX Studio, OPL automatically replaces these characters with an underscore (_). So, for example, $S-campaign-1 is renamed _S_campaign_1. You can modify the output fields directly in the SPSS filter before running the model in CPLEX Studio, if you prefer.

   Double click in the field on the right to edit the field. Replace the $ sign and the character "-" with the underscore "_". The fields now look like this (the underscore is hidden by the lines of the table).



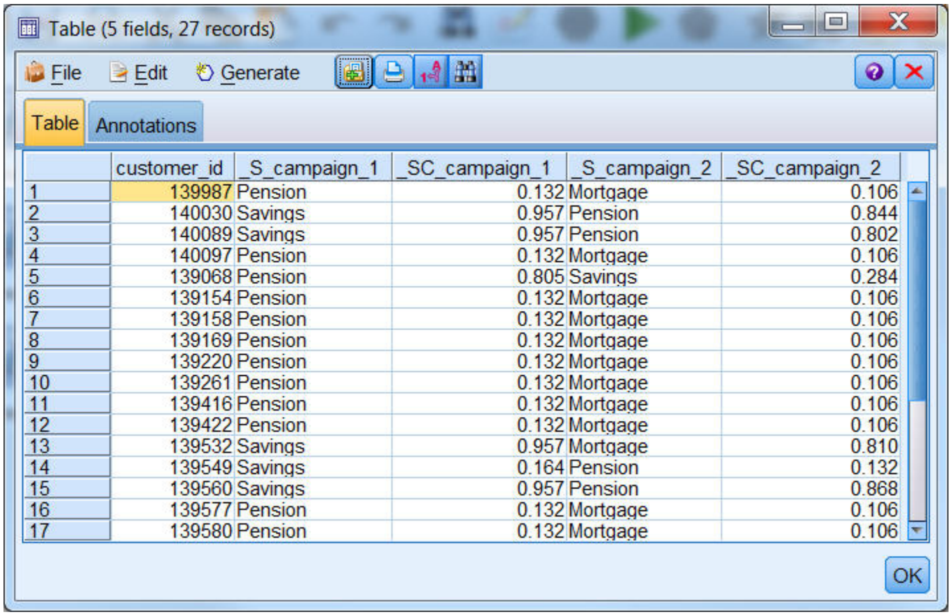
non worker percentage	✗→	non worker percentage
white collar percentage	✗→	white collar percentage
rfm score	✗→	rfm score
X random	✗→	X random
$S-campaign-1	→	S campaign 1
$SC-campaign-1	→	SC campaign 1
$S-campaign-2	→	S campaign 2
$SC-campaign-2	→	SC campaign 2

● View current fields ○ View unused field settings

Click **Apply** then **OK** to close the Filter node.

8. Run the stream in SPSS Modeler to check that the results table contains only the five required fields.

   Right-click `pm_customer_train3.sav` and select **Run From Here**. The table should look like this:



	customer_id	_S_campaign_1	_SC_campaign_1	_S_campaign_2	_SC_campaign_2
1	139987	Pension	0.132	Mortgage	0.106
2	140030	Savings	0.957	Pension	0.844
3	140089	Savings	0.957	Pension	0.802
4	140097	Pension	0.132	Mortgage	0.106
5	139068	Pension	0.805	Savings	0.284
6	139154	Pension	0.132	Mortgage	0.106
7	139158	Pension	0.132	Mortgage	0.106
8	139169	Pension	0.132	Mortgage	0.106
9	139220	Pension	0.132	Mortgage	0.106
10	139261	Pension	0.132	Mortgage	0.106
11	139416	Pension	0.132	Mortgage	0.106
12	139422	Pension	0.132	Mortgage	0.106
13	139532	Savings	0.957	Mortgage	0.810
14	139549	Savings	0.164	Pension	0.132
15	139560	Savings	0.957	Pension	0.868
16	139577	Pension	0.132	Mortgage	0.106
17	139580	Pension	0.132	Mortgage	0.106

9. Save the stream and exit SPSS Modeler.
10. In the OPL IDE, **move the focus to the OPL projects panel and refresh (F5 key)** so that OPL can access the edited stream.

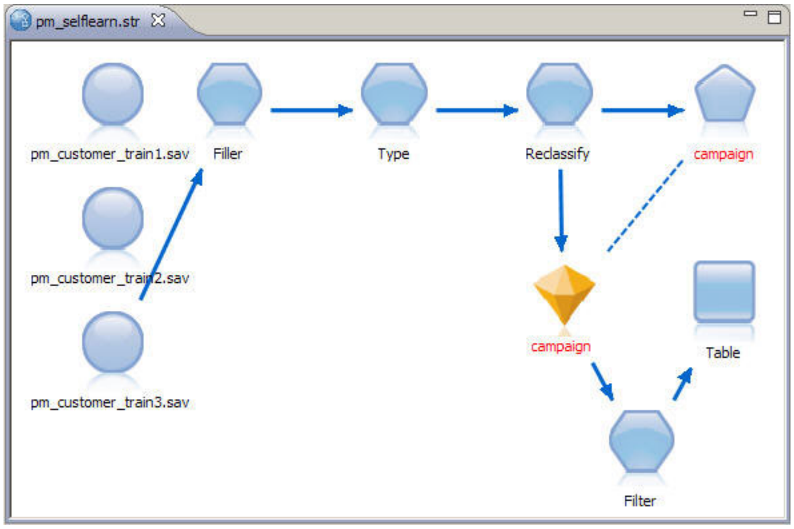
Note:

You cannot modify streams in OPL, so any changes to streams must be made with SPSS Modeler.

Chapter 3. Running an SPSS stream in the IDE

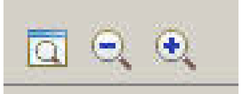
After preparing the environment and adapting the stream, you can run an SPSS data stream in CPLEX Studio.

1. In the IDE, right-click on the stream `pm_selflearn.str` and select **Open With > SPSS Stream Editor** to display the nodes of the stream. If the stream does not display, close the IDE and reopen it.

   Note: If several nodes have the same name, the names of those nodes are displayed in red.

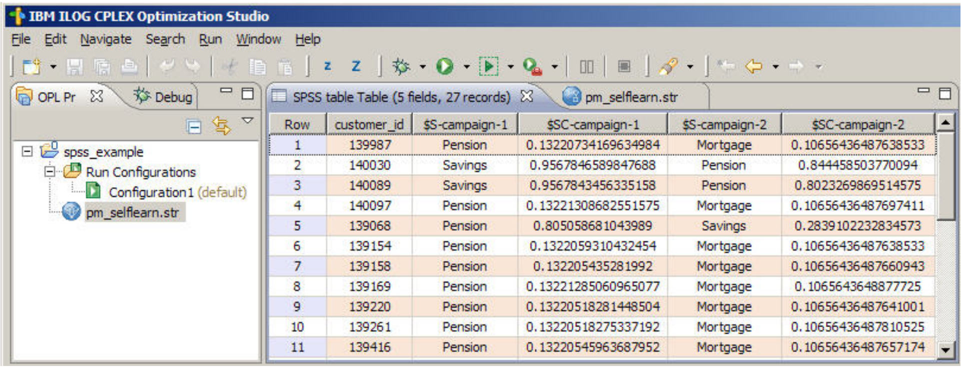
   When the SPSS Stream Editor is open, three zoom buttons become available:

    Zoom Fit, Zoom Out, Zoom In.

2. The following run commands are available in the SPSS Stream Editor when you right-click on a node:
   - **SPSS Run and display results as tables**
   - **SPSS Run and generate OPL files**

   To avoid a warning message in the IDE, you need to have a default configuration in the IDE before executing a command. Earlier in this tutorial we created `Configuration1` as the default configuration.

3. Select the **Table** node and click **SPSS Run and display results as tables for Table** to display a snapshot of the results table in the IDE. This is the results table that was modified in SPSS Modeler.


IBM ILOG CPLEX Optimization Studio
File Edit Navigate Search Run Window Help
OPL Pr
Debug
spss_example
Run Configurations
Configuration1 (default)
pm_selflearn.str
SPSS table Table (5 fields, 27 records)
pm_selflearn.str

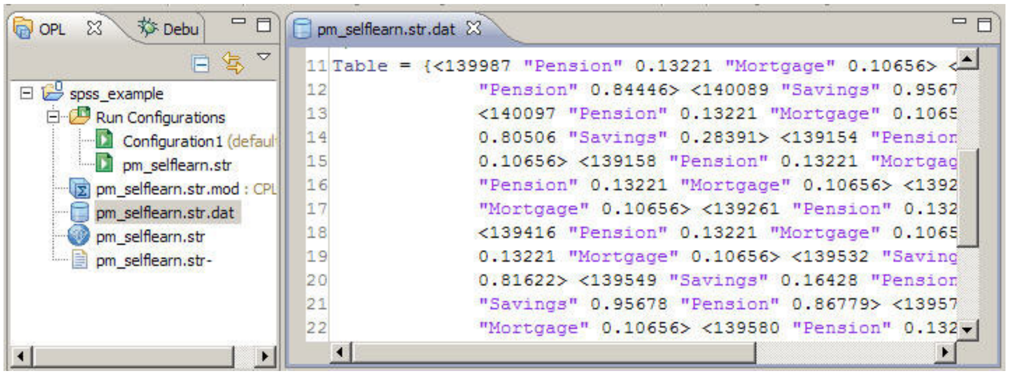
Row	customer_id	$S-campaign-1	$SC-campaign-1	$S-campaign-2	$SC-campaign-2
1	139987	Pension	0.13220734169634984	Mortgage	0.10656436487638533
2	140030	Savings	0.9567846589847688	Pension	0.844458503770094
3	140089	Savings	0.9567843456335158	Pension	0.8023269869514575
4	140097	Pension	0.13221308682551575	Mortgage	0.10656436487697411
5	139068	Pension	0.805058681043989	Savings	0.2839102232834573
6	139154	Pension	0.1322059310432454	Mortgage	0.10656436487638533
7	139158	Pension	0.132205435281992	Mortgage	0.10656436487660943
8	139169	Pension	0.13221285060965077	Mortgage	0.1065643648877725
9	139220	Pension	0.13220518281448504	Mortgage	0.10656436487641001
10	139261	Pension	0.13220518275337192	Mortgage	0.10656436487810525
11	139416	Pension	0.13220545963687952	Mortgage	0.10656436487657174

4. Select the **Table** node and click **SPSS Run and generate OPL files for Table**.

   This command copies a snapshot of the results table into an OPL `.dat` file and creates a skeleton of the optimization model. This allows you to work with data from SPSS Modeler in OPL format. A run configuration that uses the two newly-generated files is created automatically.

OPL data file

The new data file created, `pm_selflearn.str.dat`, contains the output of the table in OPL format.

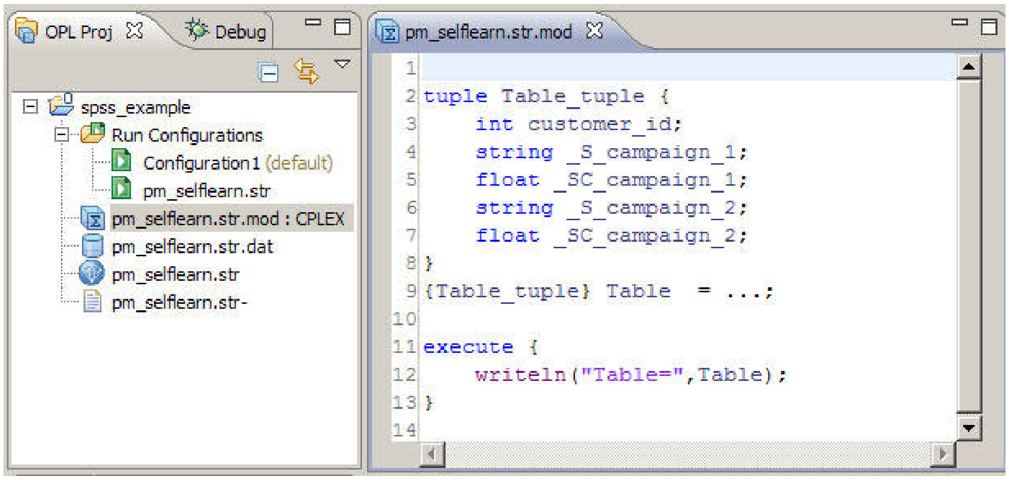


```
Table = {<139987 "Pension" 0.13221 "Mortgage" 0.10656> <
                 "Pension" 0.84446> <140089 "Savings" 0.9567
                 <140097 "Pension" 0.13221 "Mortgage" 0.1065
                 0.80506 "Savings" 0.28391> <139154 "Pension
                 0.10656> <139158 "Pension" 0.13221 "Mortgag
                 "Pension" 0.13221 "Mortgage" 0.10656> <1392
                 "Mortgage" 0.10656> <139261 "Pension" 0.132
                 <139416 "Pension" 0.13221 "Mortgage" 0.1065
                 0.13221 "Mortgage" 0.10656> <139532 "Saving
                 0.81622> <139549 "Savings" 0.16428 "Pension
                 "Savings" 0.95678 "Pension" 0.86779> <13957
                 "Mortgage" 0.10656> <139580 "Pension" 0.132
```

OPL model file

The new model file created, `pm_selflearn.str.mod`, contains the definition of the table using tuple schemas.

As OPL cannot process the characters $, &, #, %, -, ., or an empty string, it automatically replaces them with an underscore (_). So, for example, $S-campaign-1 is renamed _S_campaign_1. If you did not rename the characters in the preparatory steps, OPL renames them at this point.



```

tuple Table_tuple {
    int customer_id;
    string _S_campaign_1;
    float _SC_campaign_1;
    string _S_campaign_2;
    float _SC_campaign_2;
}
{Table_tuple} Table  = ...;

execute {
    writeln("Table=",Table);
}

```

With the SPSS Connector, you can create and modify an optimization model in the IDE, and manipulate data from an SPSS stream, without repeatedly launching the stream in SPSS Modeler.

For example, if the SPSS analyst is not the same person as the OR expert, this method provides a way for them to work together. The SPSS analyst continues to work on the stream in SPSS Modeler, while the optimization specialist builds a run configuration in the IDE based on a small fixed data set.

Next step

When the model is ready to go to production, the optimization specialist needs to create a new run configuration with the `.mod` file, and a new `.dat` file containing a connection to the SPSS stream.

The next section explains how to establish the connection with the stream.

Chapter 4. Running an optimization model in the IDE and reading data from SPSS Modeler

Activate the SPSS Connector to run an optimization model in the IDE and read data from an SPSS stream.

1. Create a new run configuration in the IDE, with the skeleton model `pm_selflearn.str.mod` and an empty `.dat` file. For this example, use the default run configuration `Configuration1` created earlier.

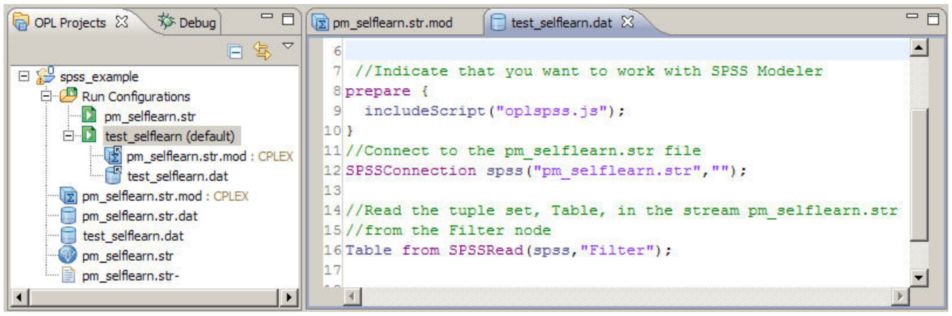
   Rename the configuration `test_selflearn` for clarity.
2. Create a new data file, `test_selflearn.dat` and add it to the run configuration `test_selflearn`. In this data file, enter OPL code to declare that you want to:
   - work with SPSS Modeler
   - create a connection between OPL and an SPSS stream
   - read data from the stream into an OPL tuple set
3. To activate the connection with SPSS Modeler, enter the following OPL code at the beginning of the `.dat` file:

   ```
   prepare {includeScript("oplspss.js");}
   ```
4. Then add the SPSSConnection keyword in the `.dat` file to connect to the stream `pm_selflearn.str` in SPSS Modeler.

   ```
   SPSSConnection spss("pm_selflearn.str","");
   ```

   The second argument to `SPSSConnection`, the empty string `""`, is for future extensions of the SPSS connector.
5. Add the SPSSRead keyword in the `.dat` file to read the tuple set `Table` in the stream, from the `Filter` node.

   ```
   Table from SPSSRead(spss,"Filter");
   ```
6. The OPL data file, `test_selflearn.dat`, now looks like this:

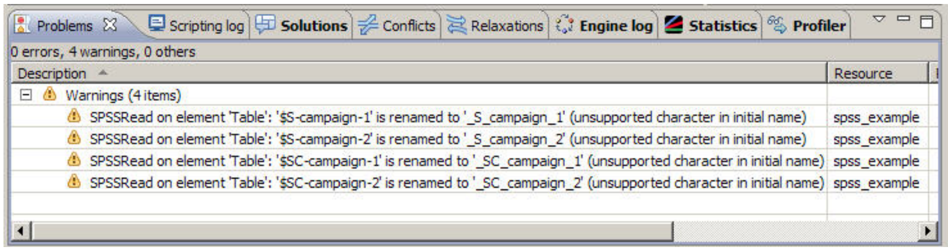
   ```

    //Indicate that you want to work with SPSS Modeler
   prepare {
     includeScript("oplspss.js");
   }
   //Connect to the pm_selflearn.str file
   SPSSConnection spss("pm_selflearn.str","");

   //Read the tuple set, Table, in the stream pm_selflearn.str
   //from the Filter node
   Table from SPSSRead(spss,"Filter");

   ```

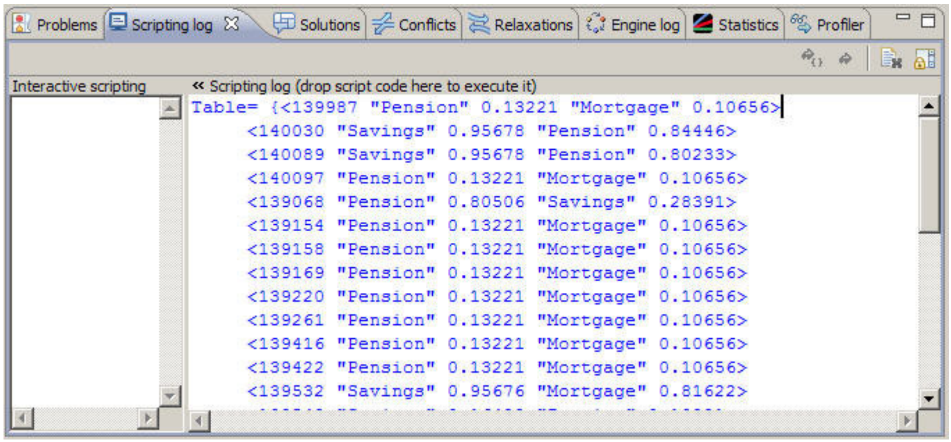
   Save the file `test_selflearn.dat`.
7. Add the model `pm_selflearn.str.mod` and the new data file `test_selflearn.dat` to the renamed configuration. Run the configuration `test_selflearn`.
8. The Problems tab displays warnings about the renamed characters, if you haven't already renamed them. (You can rename characters that are not supported by OPL, directly in SPSS Modeler, before running the OPL model in the IDE. See “Adapting the SPSS stream for CPLEX Studio” on page 5.)



0 errors, 4 warnings, 0 others

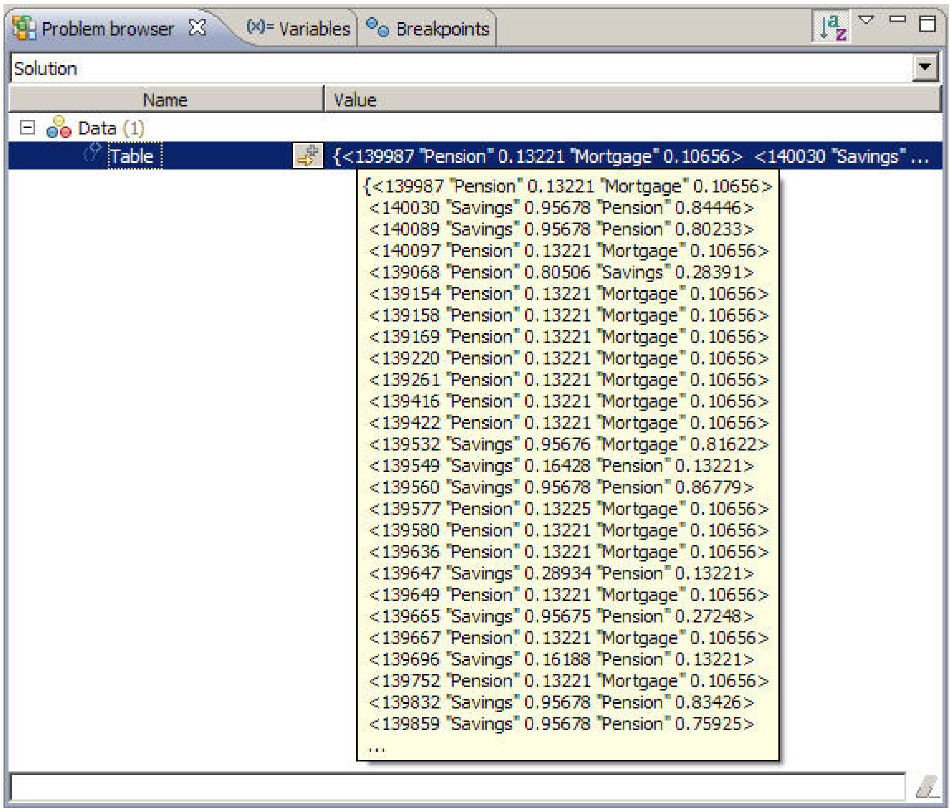
Description	Resource
⊟ Warnings (4 items)	
SPSSRead on element 'Table': '$S-campaign-1' is renamed to '_S_campaign_1' (unsupported character in initial name)	spss_example
SPSSRead on element 'Table': '$S-campaign-2' is renamed to '_S_campaign_2' (unsupported character in initial name)	spss_example
SPSSRead on element 'Table': '$SC-campaign-1' is renamed to '_SC_campaign_1' (unsupported character in initial name)	spss_example
SPSSRead on element 'Table': '$SC-campaign-2' is renamed to '_SC_campaign_2' (unsupported character in initial name)	spss_example

9. The results table is displayed in the Scripting log.



```
Table= {<139987 "Pension" 0.13221 "Mortgage" 0.10656>
     <140030 "Savings" 0.95678 "Pension" 0.84446>
     <140089 "Savings" 0.95678 "Pension" 0.80233>
     <140097 "Pension" 0.13221 "Mortgage" 0.10656>
     <139068 "Pension" 0.80506 "Savings" 0.28391>
     <139154 "Pension" 0.13221 "Mortgage" 0.10656>
     <139158 "Pension" 0.13221 "Mortgage" 0.10656>
     <139169 "Pension" 0.13221 "Mortgage" 0.10656>
     <139220 "Pension" 0.13221 "Mortgage" 0.10656>
     <139261 "Pension" 0.13221 "Mortgage" 0.10656>
     <139416 "Pension" 0.13221 "Mortgage" 0.10656>
     <139422 "Pension" 0.13221 "Mortgage" 0.10656>
     <139532 "Savings" 0.95676 "Mortgage" 0.81622>
```

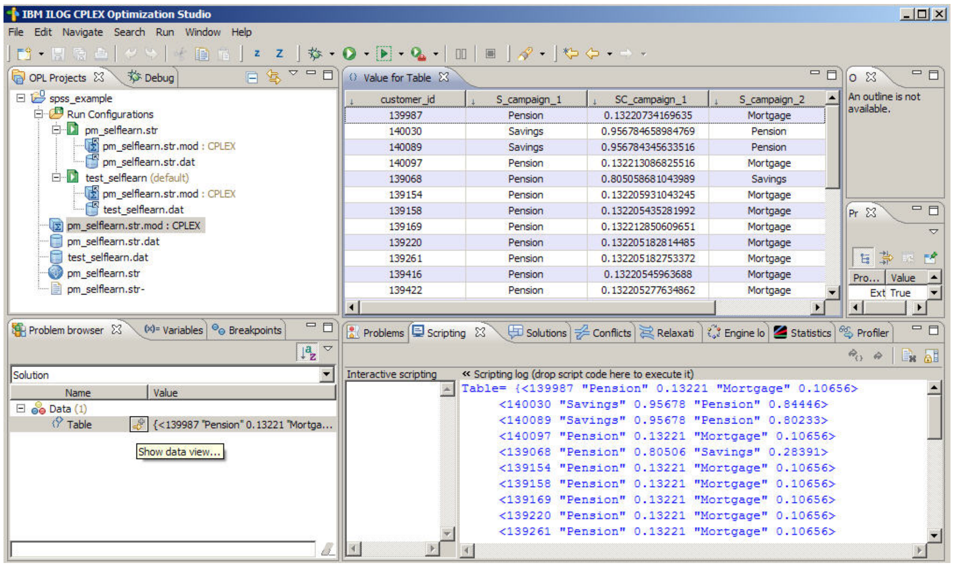
10. In the Problem browser, the results set can be displayed if you hover the mouse over the value cell of the `Table` data item.



Name	Value
⊟ Data (1)	
Table	{<139987 "Pension" 0.13221 "Mortgage" 0.10656> <140030 "Savings" ...

```
{<139987 "Pension" 0.13221 "Mortgage" 0.10656>
 <140030 "Savings" 0.95678 "Pension" 0.84446>
 <140089 "Savings" 0.95678 "Pension" 0.80233>
 <140097 "Pension" 0.13221 "Mortgage" 0.10656>
 <139068 "Pension" 0.80506 "Savings" 0.28391>
 <139154 "Pension" 0.13221 "Mortgage" 0.10656>
 <139158 "Pension" 0.13221 "Mortgage" 0.10656>
 <139169 "Pension" 0.13221 "Mortgage" 0.10656>
 <139220 "Pension" 0.13221 "Mortgage" 0.10656>
 <139261 "Pension" 0.13221 "Mortgage" 0.10656>
 <139416 "Pension" 0.13221 "Mortgage" 0.10656>
 <139422 "Pension" 0.13221 "Mortgage" 0.10656>
 <139532 "Savings" 0.95676 "Mortgage" 0.81622>
 <139549 "Savings" 0.16428 "Pension" 0.13221>
 <139560 "Savings" 0.95678 "Pension" 0.86779>
 <139577 "Pension" 0.13225 "Mortgage" 0.10656>
 <139580 "Pension" 0.13221 "Mortgage" 0.10656>
 <139636 "Pension" 0.13221 "Mortgage" 0.10656>
 <139647 "Savings" 0.28934 "Pension" 0.13221>
 <139649 "Pension" 0.13221 "Mortgage" 0.10656>
 <139665 "Savings" 0.95675 "Pension" 0.27248>
 <139667 "Pension" 0.13221 "Mortgage" 0.10656>
 <139696 "Savings" 0.16188 "Pension" 0.13221>
 <139752 "Pension" 0.13221 "Mortgage" 0.10656>
 <139832 "Savings" 0.95678 "Pension" 0.83426>
 <139859 "Savings" 0.95678 "Pension" 0.75925>
 ...
```

11. Finally, you can display the results table in the IDE editor by clicking the **Show data view** button.



customer_id	S_campaign_1	SC_campaign_1	S_campaign_2
139987	Pension	0.13220734169635	Mortgage
140030	Savings	0.956784658984769	Pension
140089	Savings	0.956784345633516	Pension
140097	Pension	0.132213086825516	Mortgage
139068	Pension	0.805058681043989	Savings
139154	Pension	0.132205931043245	Mortgage
139158	Pension	0.132205435281992	Mortgage
139169	Pension	0.132212850609651	Mortgage
139220	Pension	0.132205182814485	Mortgage
139261	Pension	0.132205182753372	Mortgage
139416	Pension	0.13220545963688	Mortgage
139422	Pension	0.132205277634862	Mortgage

```
Table= {<139987 "Pension" 0.13221 "Mortgage" 0.10656>
     <140030 "Savings" 0.95678 "Pension" 0.84446>
     <140089 "Savings" 0.95678 "Pension" 0.80233>
     <140097 "Pension" 0.13221 "Mortgage" 0.10656>
     <139068 "Pension" 0.80506 "Savings" 0.28391>
     <139154 "Pension" 0.13221 "Mortgage" 0.10656>
     <139158 "Pension" 0.13221 "Mortgage" 0.10656>
     <139169 "Pension" 0.13221 "Mortgage" 0.10656>
     <139220 "Pension" 0.13221 "Mortgage" 0.10656>
     <139261 "Pension" 0.13221 "Mortgage" 0.10656>
```

From this point you can continue working with your optimization model.

Chapter 5. Getting more information on SPSS stream execution

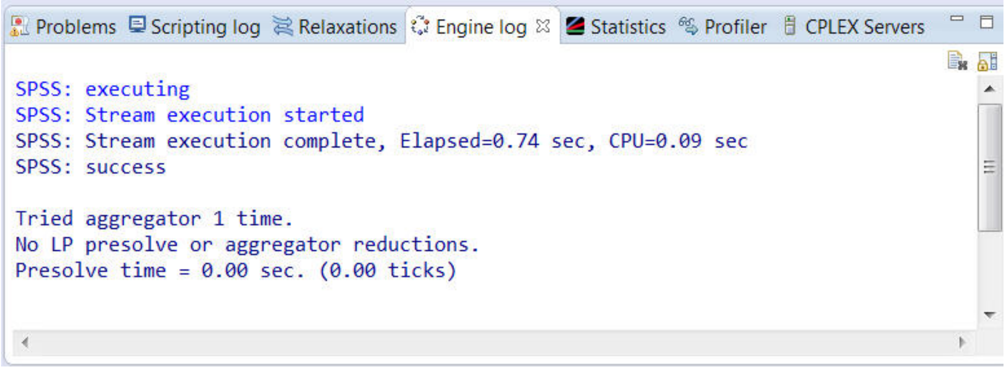
After stream execution, the log from SPSS can be displayed in the engine log of the IDE.

If you execute an SPSS stream in verbose mode, the log information from SPSS Modeler is displayed in the engine log of the CPLEX Studio IDE. To enable the verbose mode, add the statement `SetVerboseSPSS` to the .dat file.

For example, the file `test_selflearn.dat` now contains:

```
prepare {
   includeScript("oplspss.js");
   SetVerboseSPSS();
}
SPSSConnection spss("pm_selflearn.str","");
Table from SPSSRead(spss,"Filter");
```

Save the modified data file and run the configuration `test_selflearn`. The log information from SPSS Modeler is displayed in the engine log of the IDE.



```
SPSS: executing
SPSS: Stream execution started
SPSS: Stream execution complete, Elapsed=0.74 sec, CPU=0.09 sec
SPSS: success

Tried aggregator 1 time.
No LP presolve or aggregator reductions.
Presolve time = 0.00 sec. (0.00 ticks)
```

Index

Printed in USA