Semarchy

xDI DEV

D Process

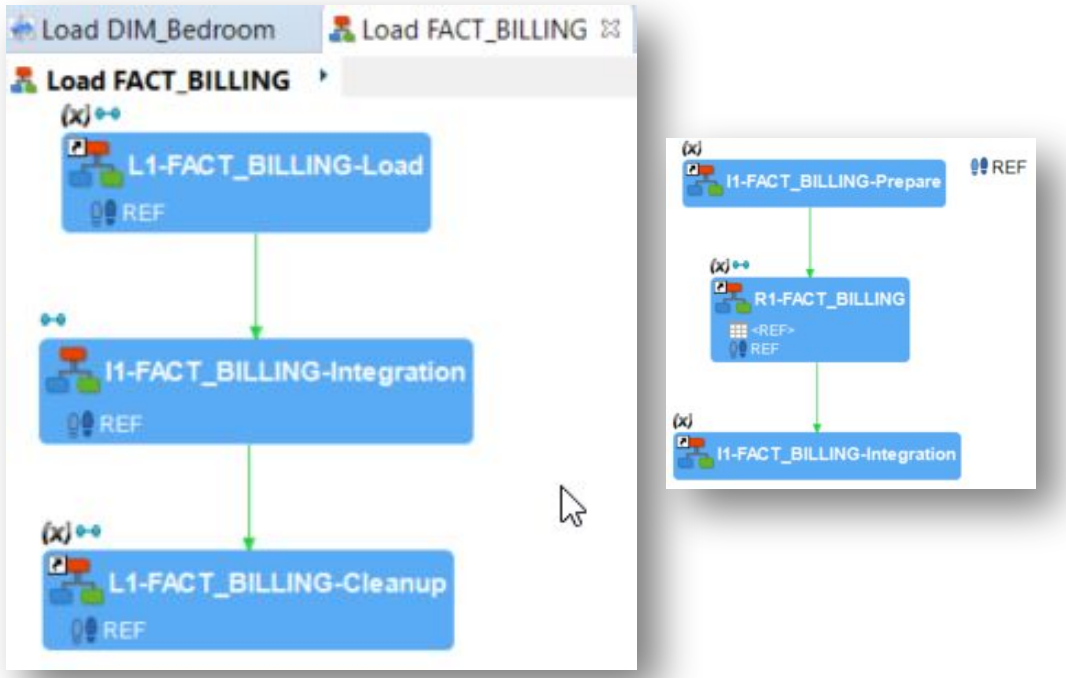
xDI DEV

D Process

D1 - Introduce process

What is a process ?

- A **Process** is a set of **Actions** executed by the Runtime
- **xDI** generates a Process from each Mapping
  - This is the type of Process you were able to consult at each execution of a Mapping

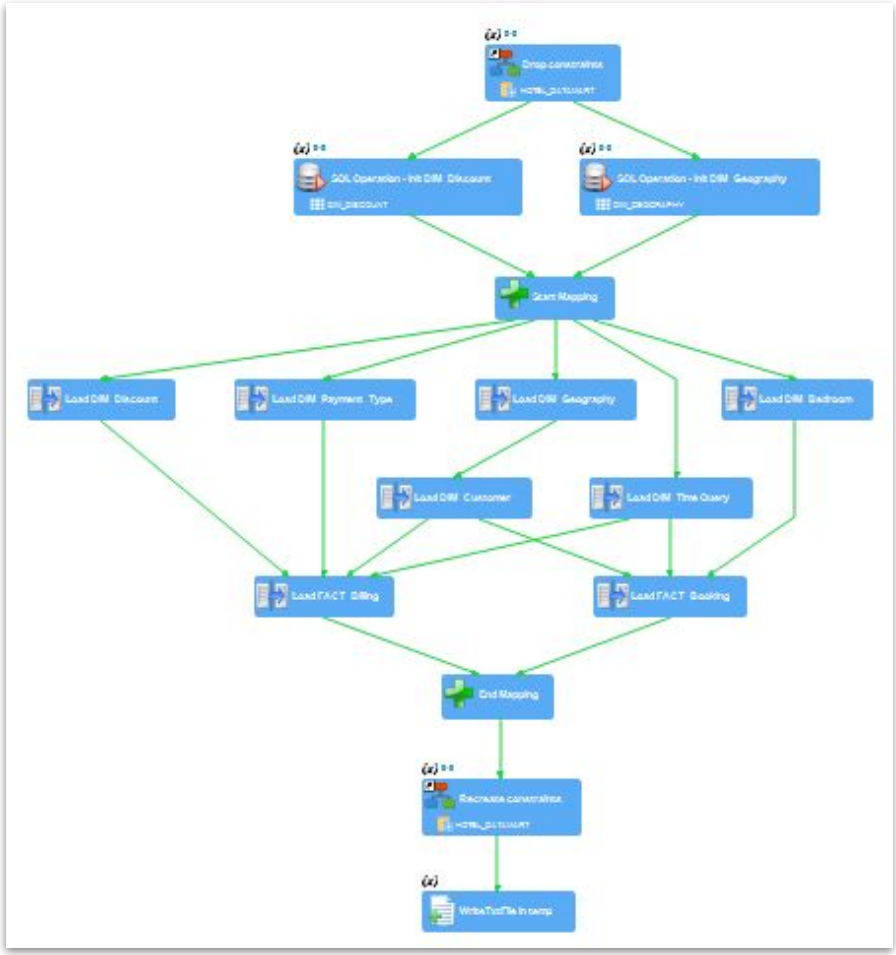


D-Process

What is a process ?

However, you can also create a Process manually

- To orchestrate higher level components
  - Mappings
  - Even other Processes
- As well as unit actions of a lower level
  - SQL commands
  - File handling
  - A lot of other tools are available



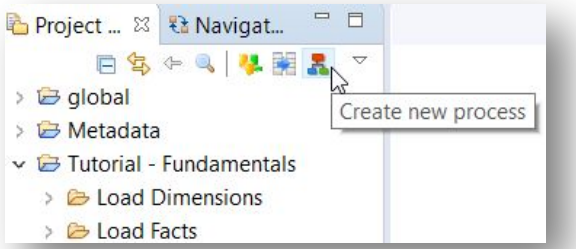
What is a process ?

- The « Process » is an orchestration of
  - Other processes
  - Unitary actions
  - Mappings
- Mappings generate processes
- Processes can call mappings
- Templates are models of processes

Creation of a process

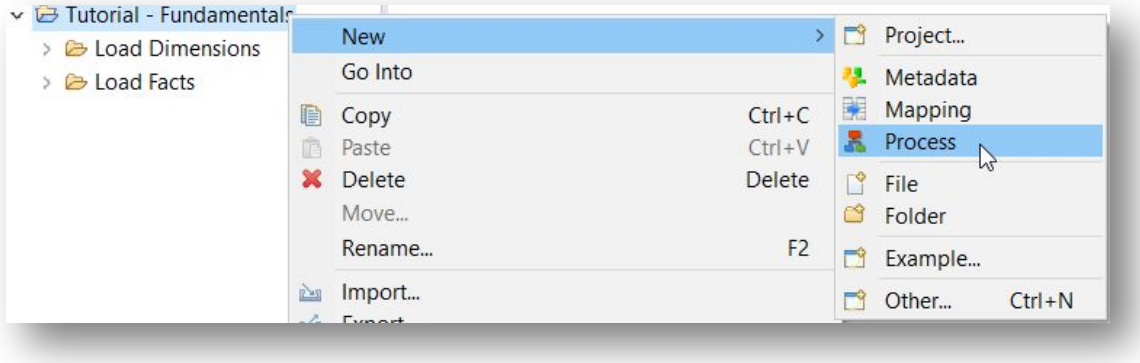
To create a new process file

- In the project explorer, click on the Process icon



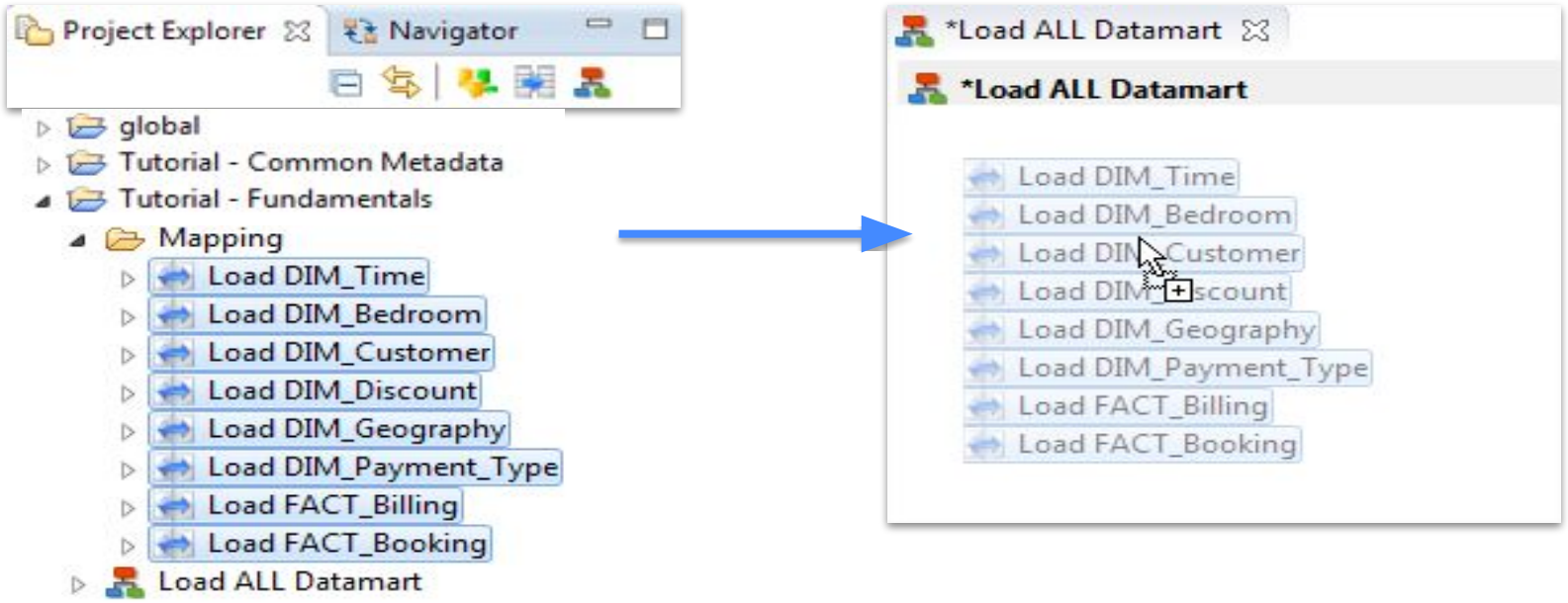
Or

- Right-Click and choose New/Process



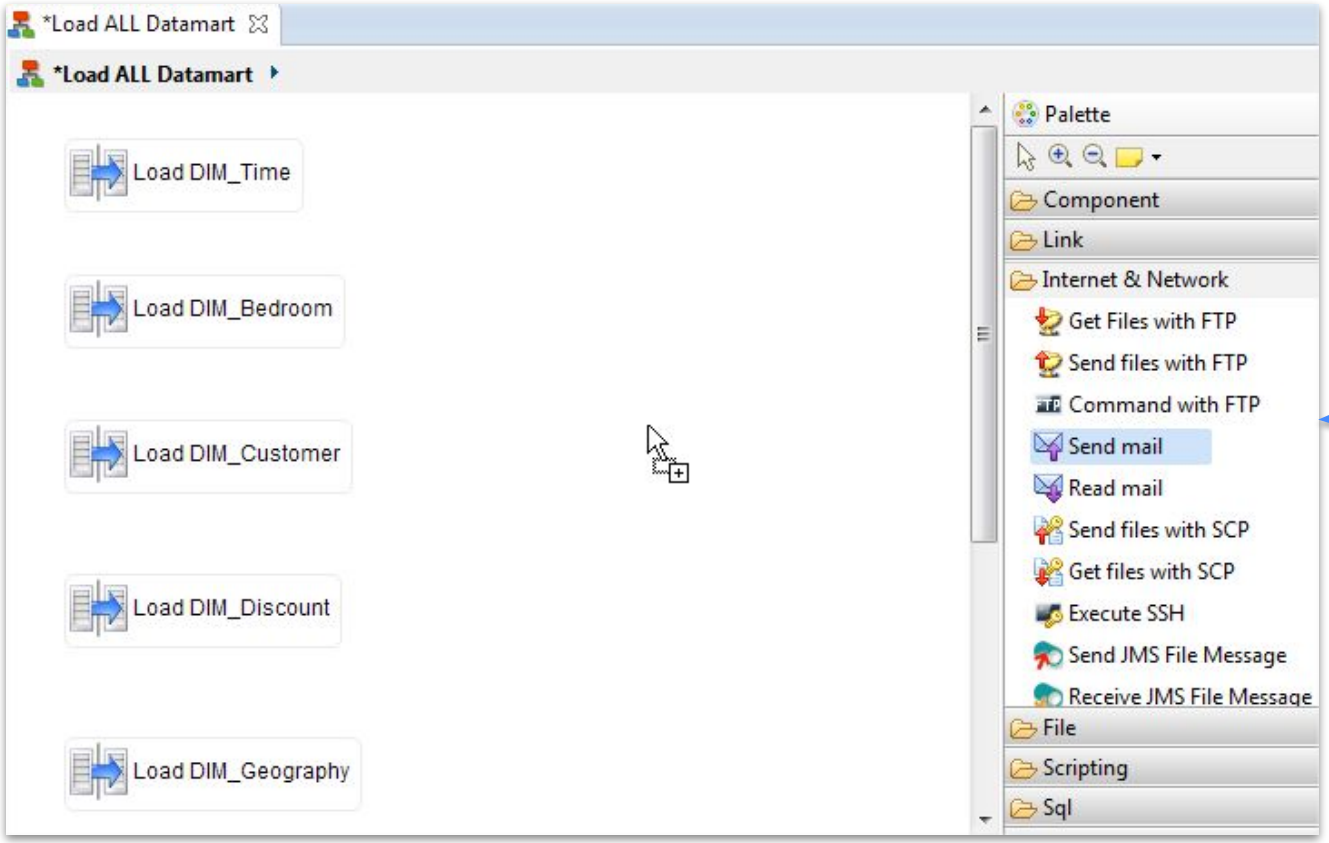
Designing a process – The mappings

You can add Mappings with a simple “Drag & Drop” into the process :



Designing a process - Palette actions creation

You can add a unitary action by using the palette :

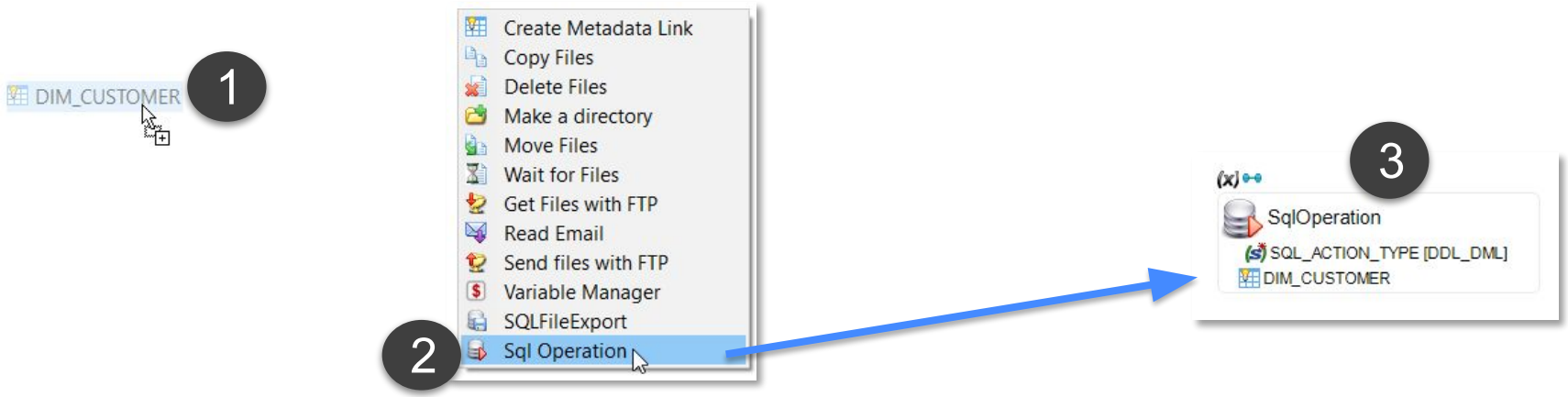


To add an action, double click on the related palette action

Designing a process – Palette actions creation

Another way to create palette action

1. Drag & drop particular Metadata in a process
2. A related window appears linked to the Metadata
3. The action is created

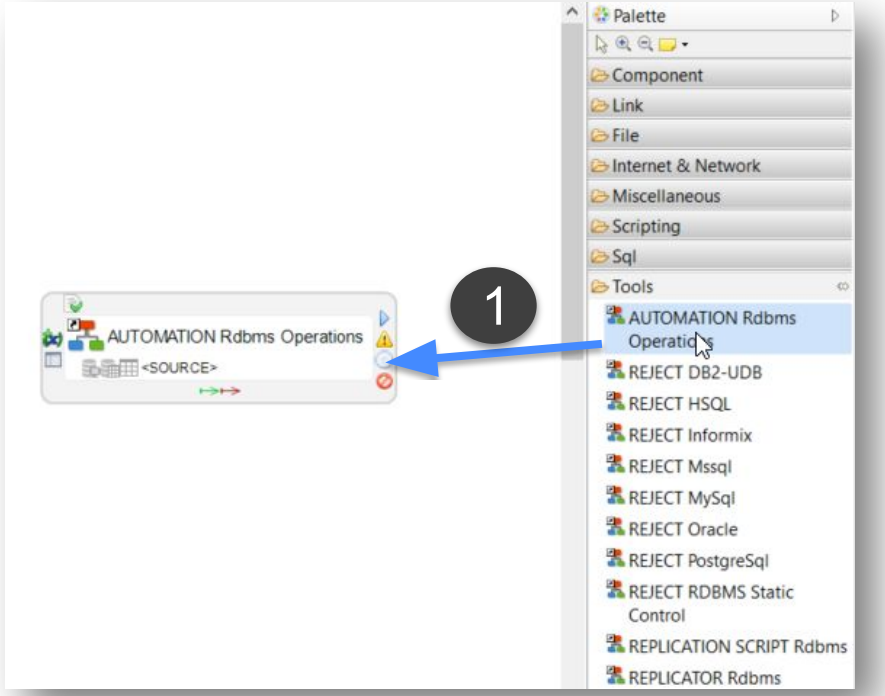


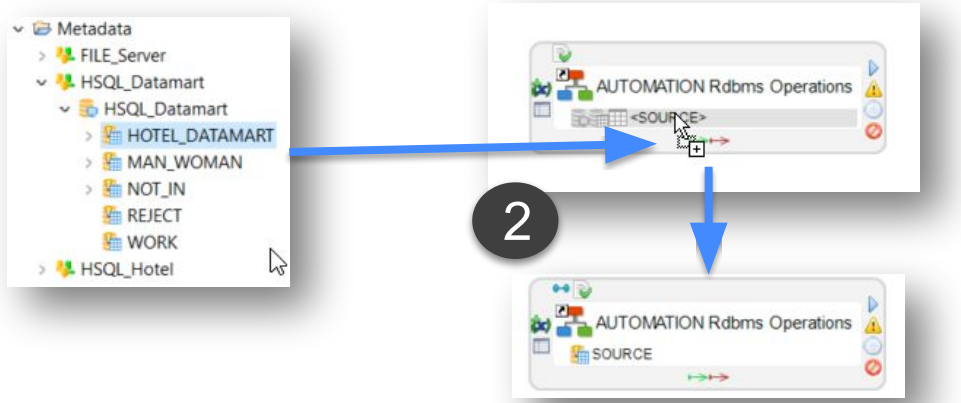
Designing a process - Other process & Templates

You can add a reference to a process or a template by

1. a simple “Drag & Drop” of the Template into the process
2. a drag & drop of metadata links on dedicated locations in the Template

A template will generally need metadata to be able to generate the appropriate code

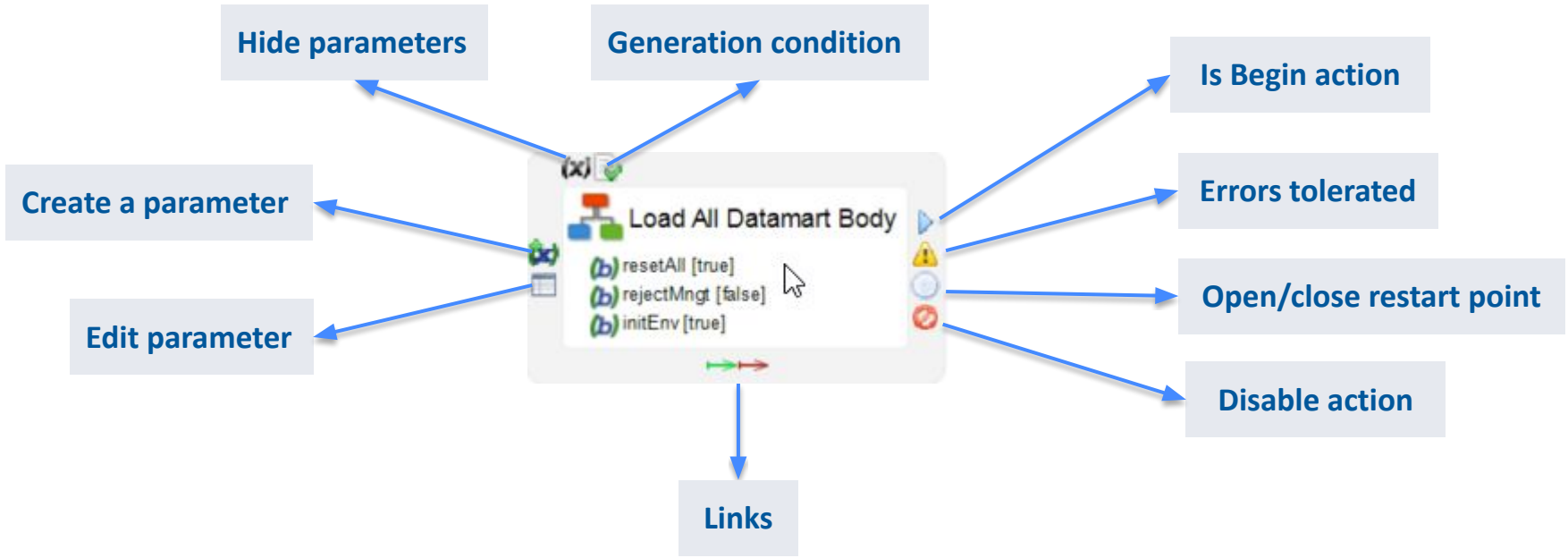




D-Process

Designing a process - Banner around the actions

A banner is available around each action in a process

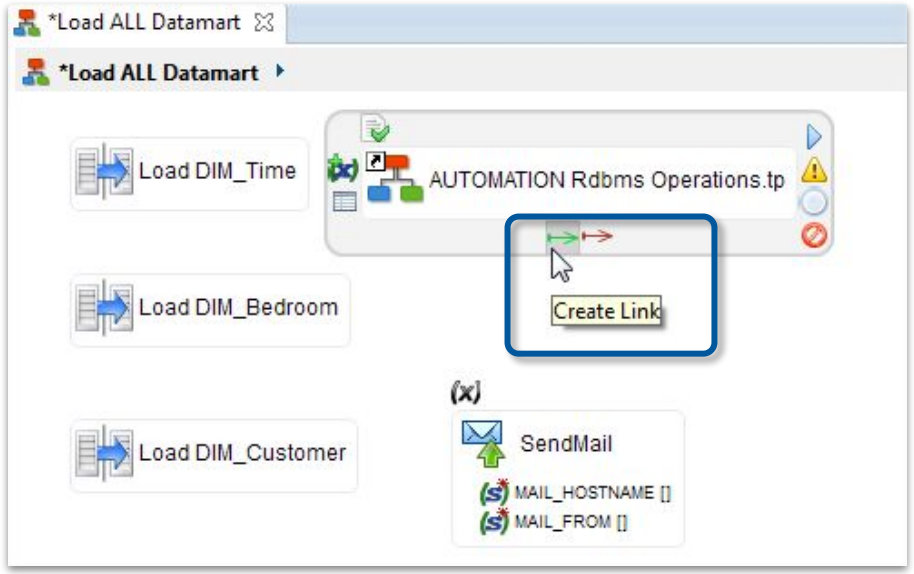


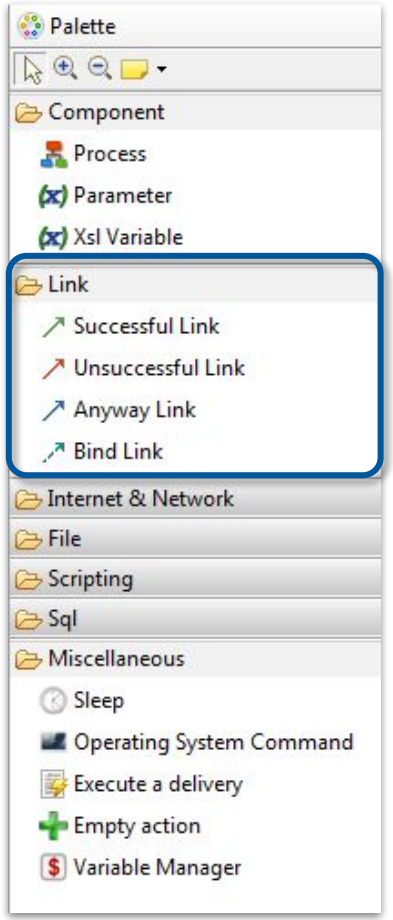
To select properly an action, locate the mouse on the name of the action, not on the properties

Designing a process - Links

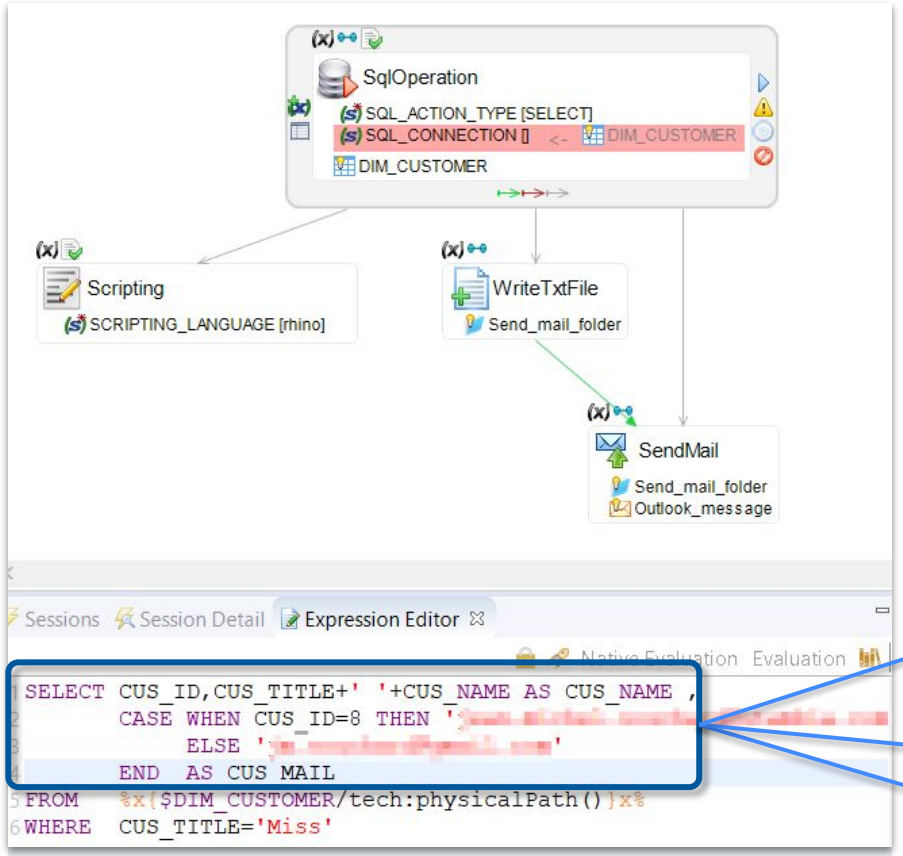
You can link two actions

- Using the corresponding link in the palette
- Directly under each action

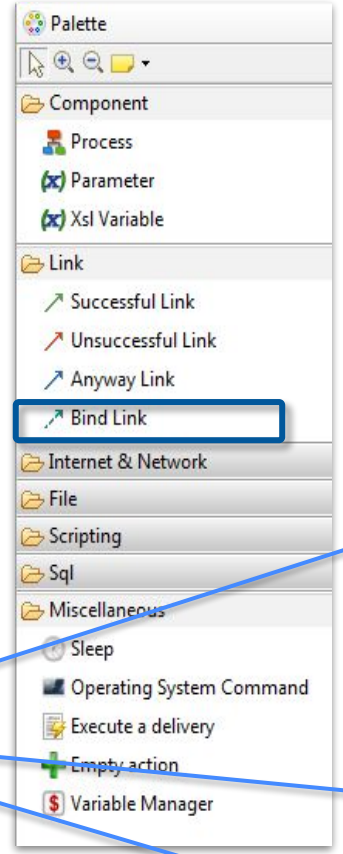


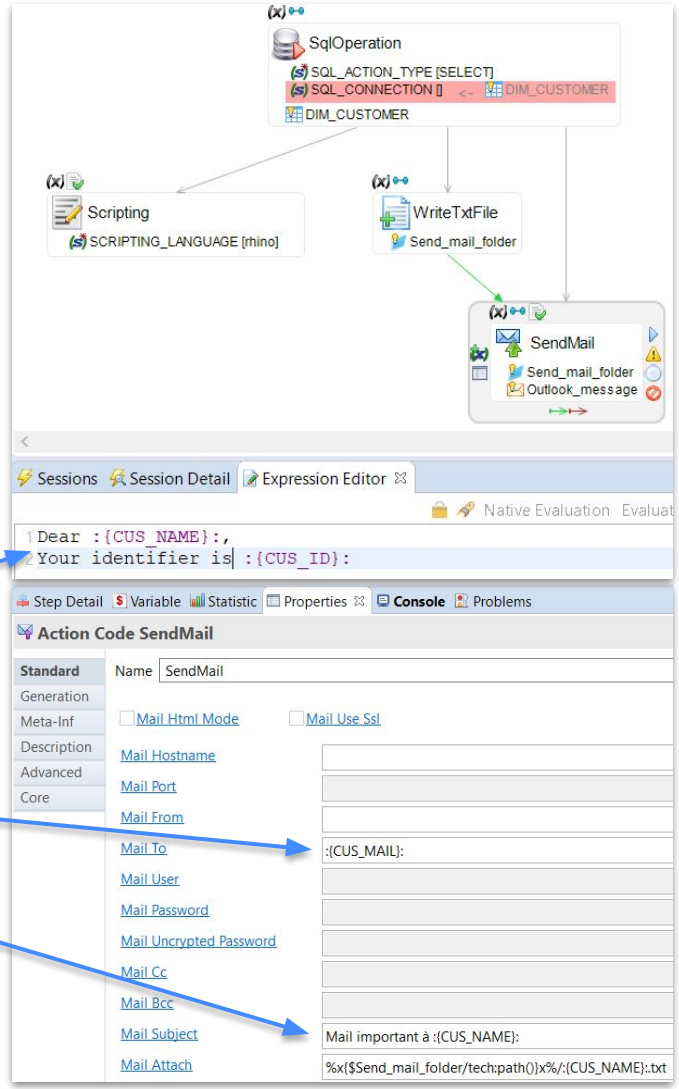


Designing a process - Bind link and variable



Possible to use columns mentioned in the "From" clause of the SQL order or alias if they exist





Bind variables in the expression editor of the action

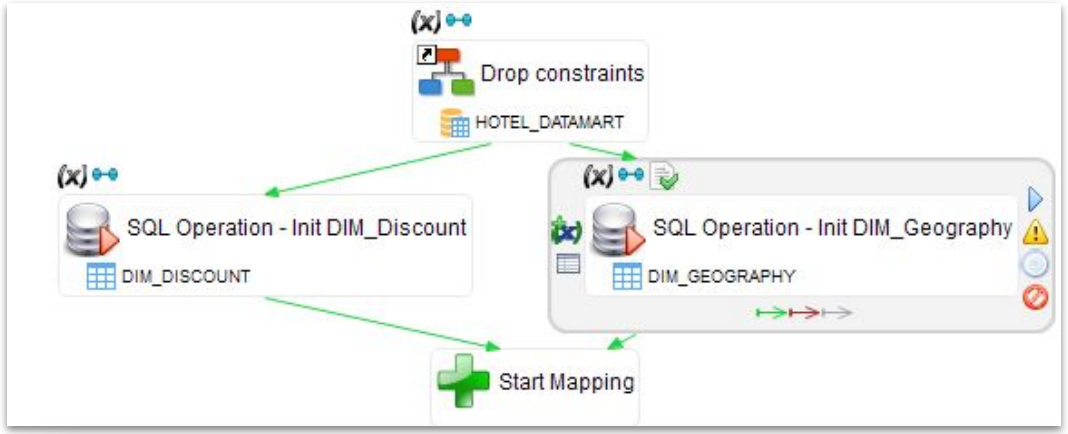
Bind variables in the properties of the action

D-Process

Designing a process - Text of unitary actions

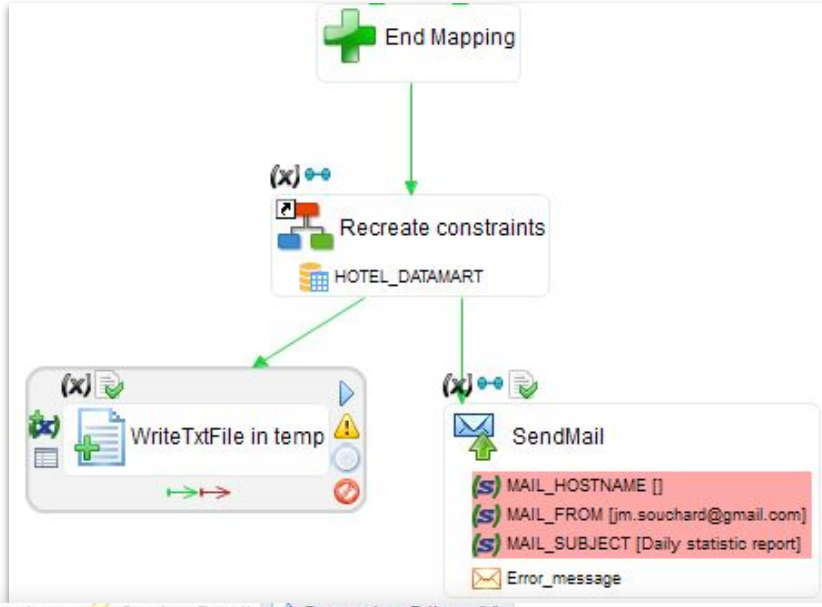
Use the Expression Editor view, to define the text of some actions

- SQL operation,
- Writing of a file,
- Send a mail, etc.



Sessions | Session Detail | Expression Editor

```
INSERT INTO %x{$DIM_GEOGRAPHY/tech:physicalPath()}x%
            (GEO_KEY_ID, GEO_ZIP_CODE, GEO_CITY, GEO_STATE_CODE, GEO_STATE)
       VALUES (0, NULL, 'No Address', NULL, NULL);
INSERT INTO %x{$DIM_GEOGRAPHY/tech:physicalPath()}x%
            (GEO_KEY_ID, GEO_ZIP_CODE, GEO_CITY, GEO_STATE_CODE, GEO_STATE)
       VAKUES (1, '?', 'Unknown Zip Code', '?', '?');
```



Sessions | Session Detail | Expression Editor

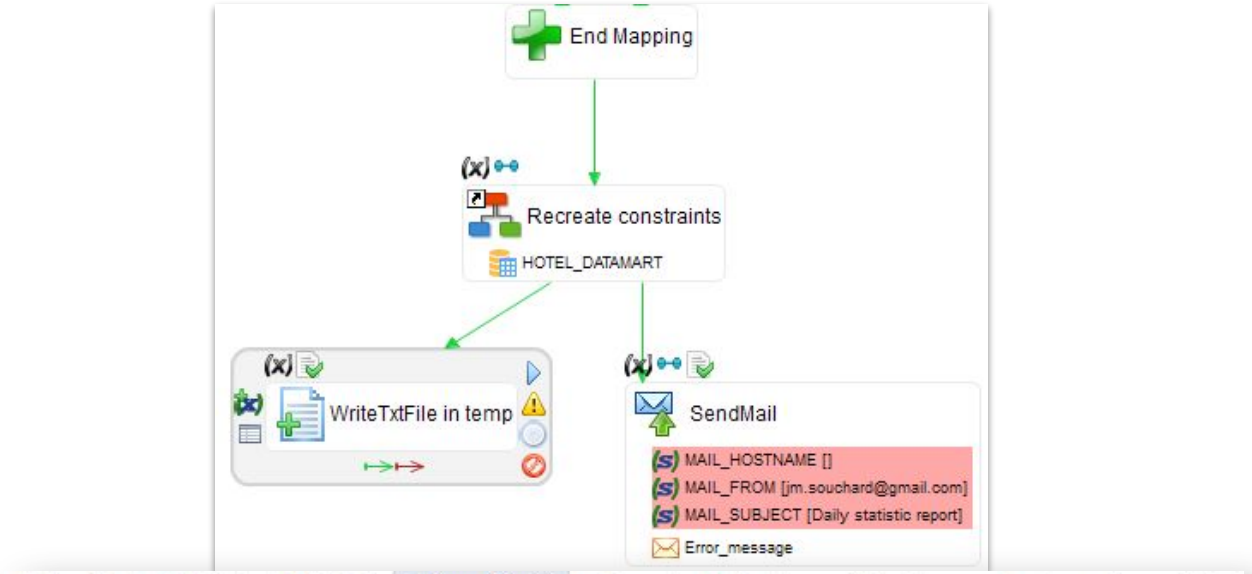
```
Session started on: ${/CORE_BEGIN_DATE}$

Steps duration:
    Load FACT_BILLING: ${../Load FACT_Billing/CORE_DURATION}$
    Load FACT_BOOKING: ${../Load FACT_Booking/CORE_DURATION}$
    Recreate Fk : ${../Recreate constraints/Create FK/CORE_DURATION}$

Total Rows: %e(rhino){__ctx__.sumVariable("SQL_NB_ROWS")}e(rhino)%
Total Inserts: %e(rhino){__ctx__.sumVariable("SQL_STAT_INSERT")}e(rhino)%
```

Designing a process - The step parameters

Each step (process, action, etc.) can be configured in the Properties view:



D-Process


Semarchy
Demo
Process

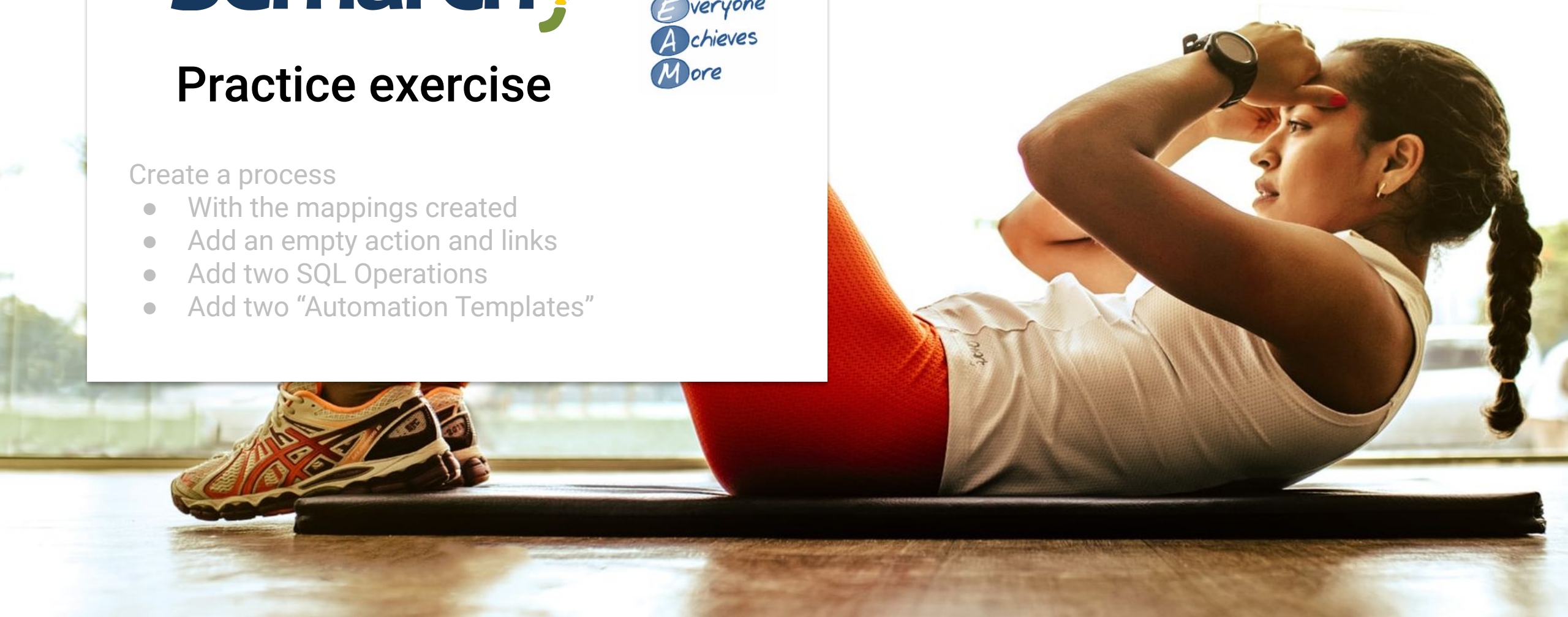
Semarchy

Together Everyone Achieves More

Practice exercise

Create a process

- With the mappings created
- Add an empty action and links
- Add two SQL Operations
- Add two “Automation Templates”

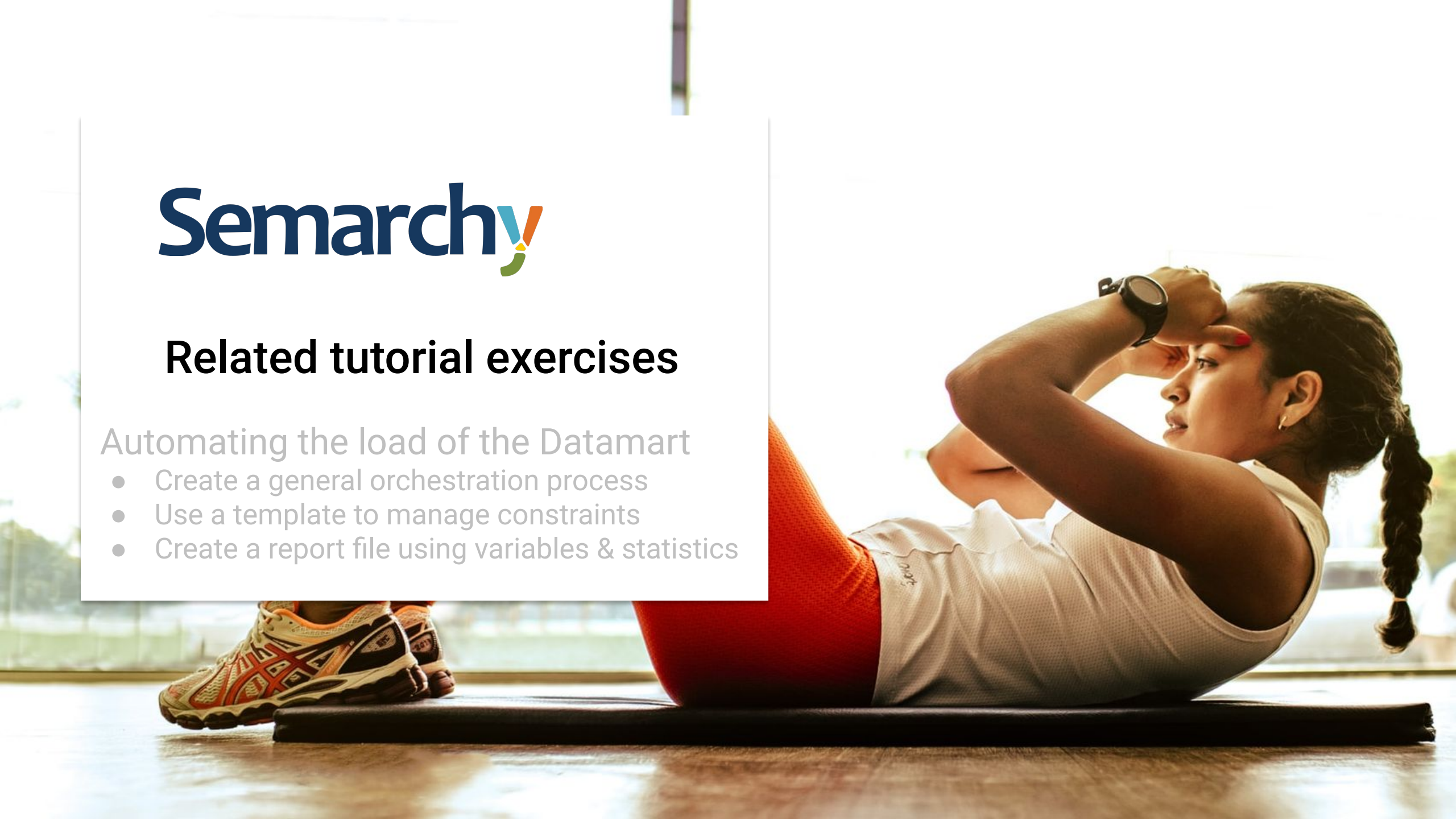


Semarchy

Related tutorial exercises

Automating the load of the Datamart

- Create a general orchestration process
- Use a template to manage constraints
- Create a report file using variables & statistics



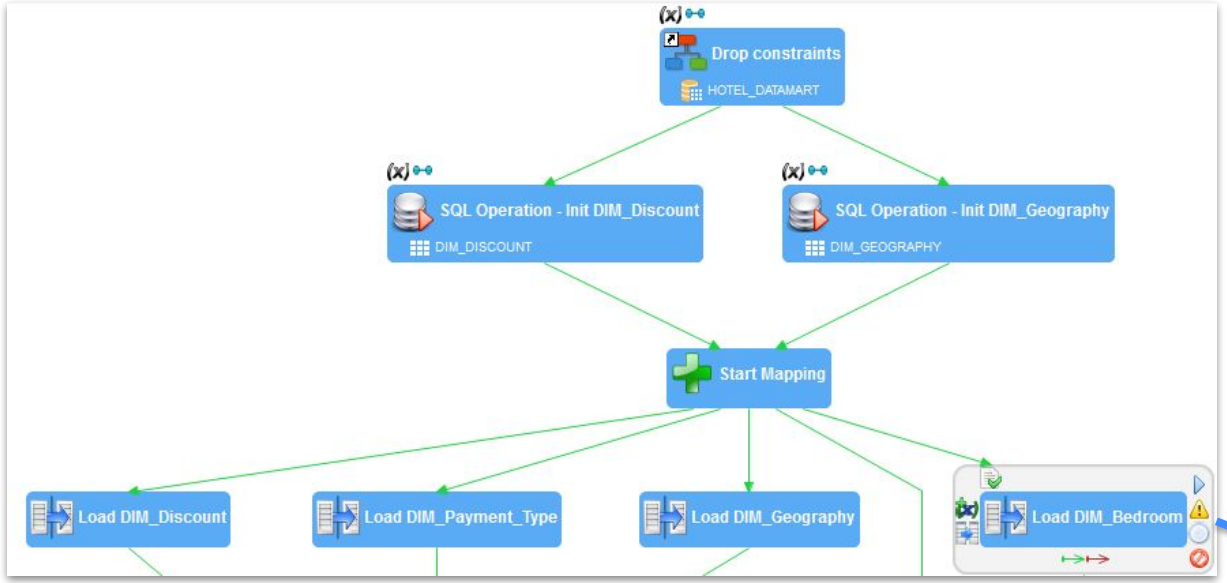
xDI DEV

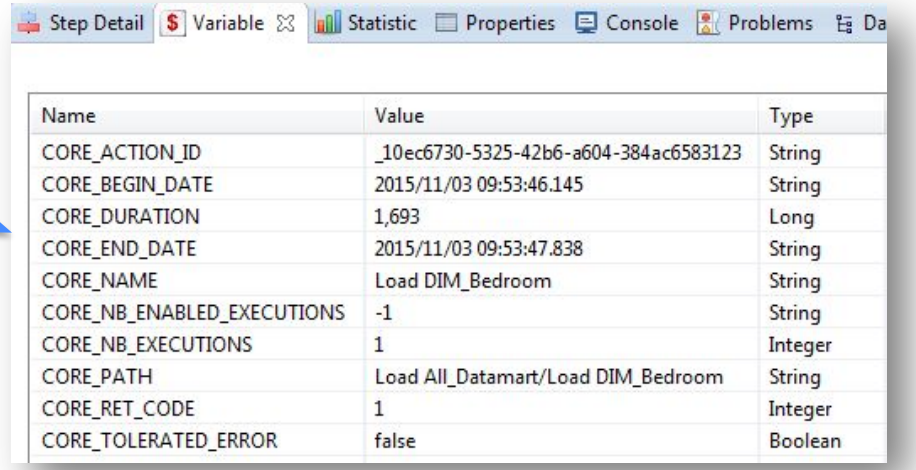
D Process

D2 - Variable, scripting & xPath

Managing variables and statistics

During execution, a process will generate variables for each step

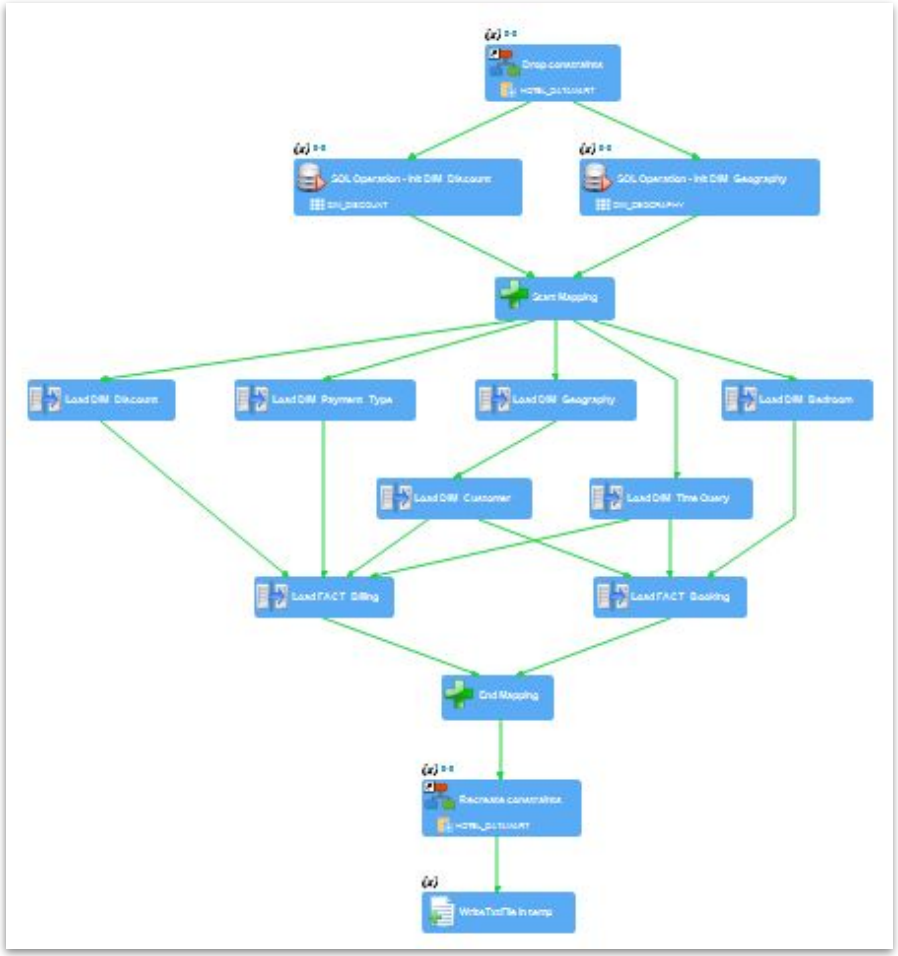


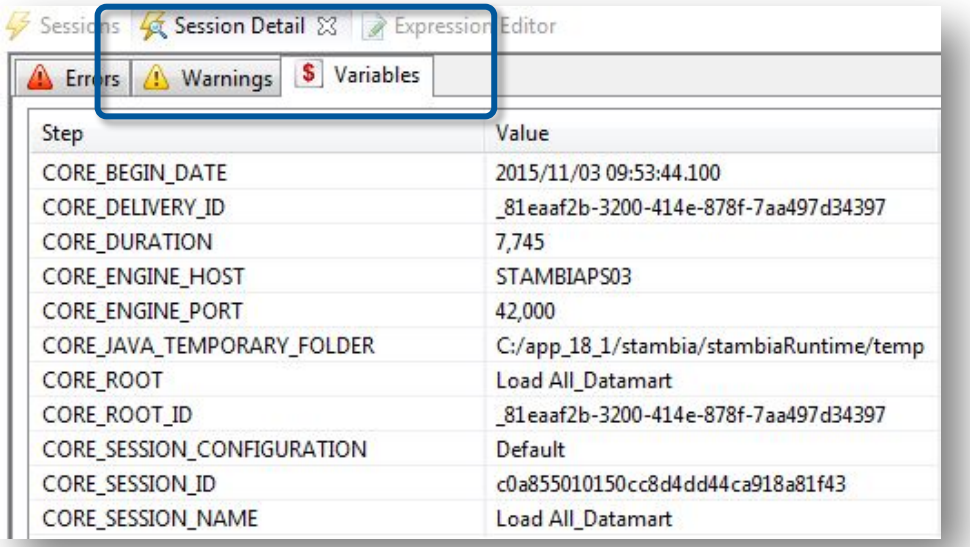


Name	Value	Type
CORE_ACTION_ID	_10ec6730-5325-42b6-a604-384ac6583123	String
CORE_BEGIN_DATE	2015/11/03 09:53:46.145	String
CORE_DURATION	1,693	Long
CORE_END_DATE	2015/11/03 09:53:47.838	String
CORE_NAME	Load DIM_Bedroom	String
CORE_NB_ENABLED_EXECUTIONS	-1	String
CORE_NB_EXECUTIONS	1	Integer
CORE_PATH	Load All_Datamart/Load DIM_Bedroom	String
CORE_RET_CODE	1	Integer
CORE_TOLERATED_ERROR	false	Boolean

Managing variables and statistics

When executing a process, a session will be created and generate variables



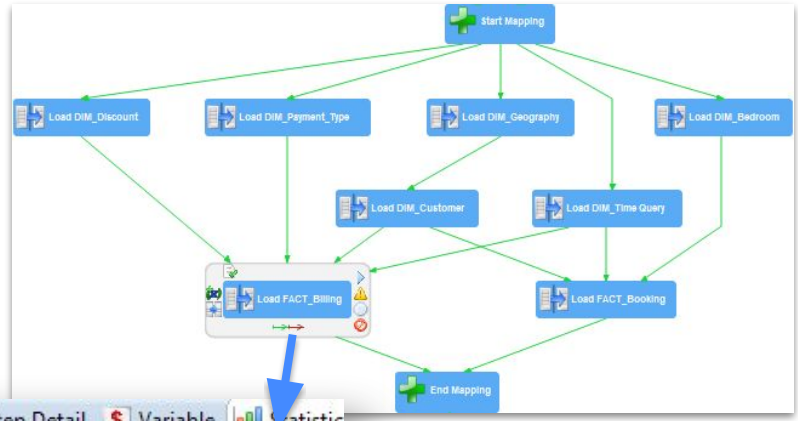


Step	Value
CORE_BEGIN_DATE	2015/11/03 09:53:44.100
CORE_DELIVERY_ID	_81eaaf2b-3200-414e-878f-7aa497d34397
CORE_DURATION	7,745
CORE_ENGINE_HOST	STAMBIAPS03
CORE_ENGINE_PORT	42,000
CORE_JAVA_TEMPORARY_FOLDER	C:/app_18_1/stambia/stambiaRuntime/temp
CORE_ROOT	Load All_Datamart
CORE_ROOT_ID	_81eaaf2b-3200-414e-878f-7aa497d34397
CORE_SESSION_CONFIGURATION	Default
CORE_SESSION_ID	c0a855010150cc8d4dd44ca918a81f43
CORE_SESSION_NAME	Load All_Datamart

D-Process

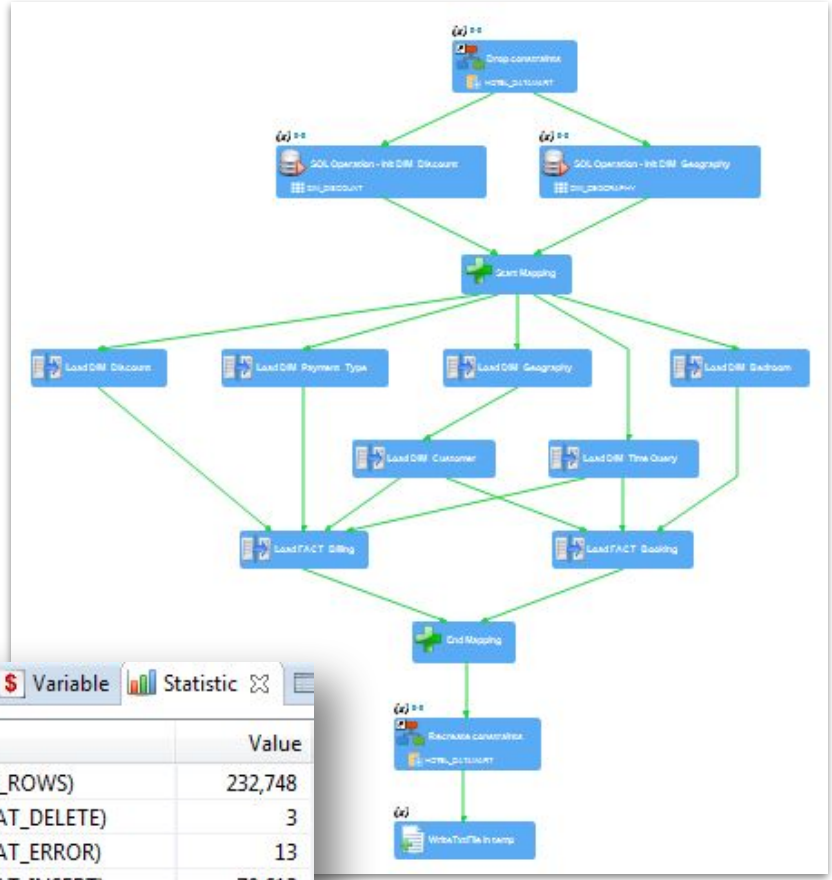
Managing variables and statistics

Some variables can be aggregated and seen in the Statistic view



Step Detail | Variable | Statistic

Name	Value
SUM(SQL_NB_ROWS)	53,316
SUM(SQL_STAT_ERROR)	2
SUM(SQL_STAT_INSERT)	12,105
SUM(SQL_STAT_UPDATE)	0



Step Detail | Variable | Statistic

Name	Value
SUM(SQL_NB_ROWS)	232,748
SUM(SQL_STAT_DELETE)	3
SUM(SQL_STAT_ERROR)	13
SUM(SQL_STAT_INSERT)	70,613
SUM(SQL_STAT_UPDATE)	0

D-Process

Managing variables and statistics

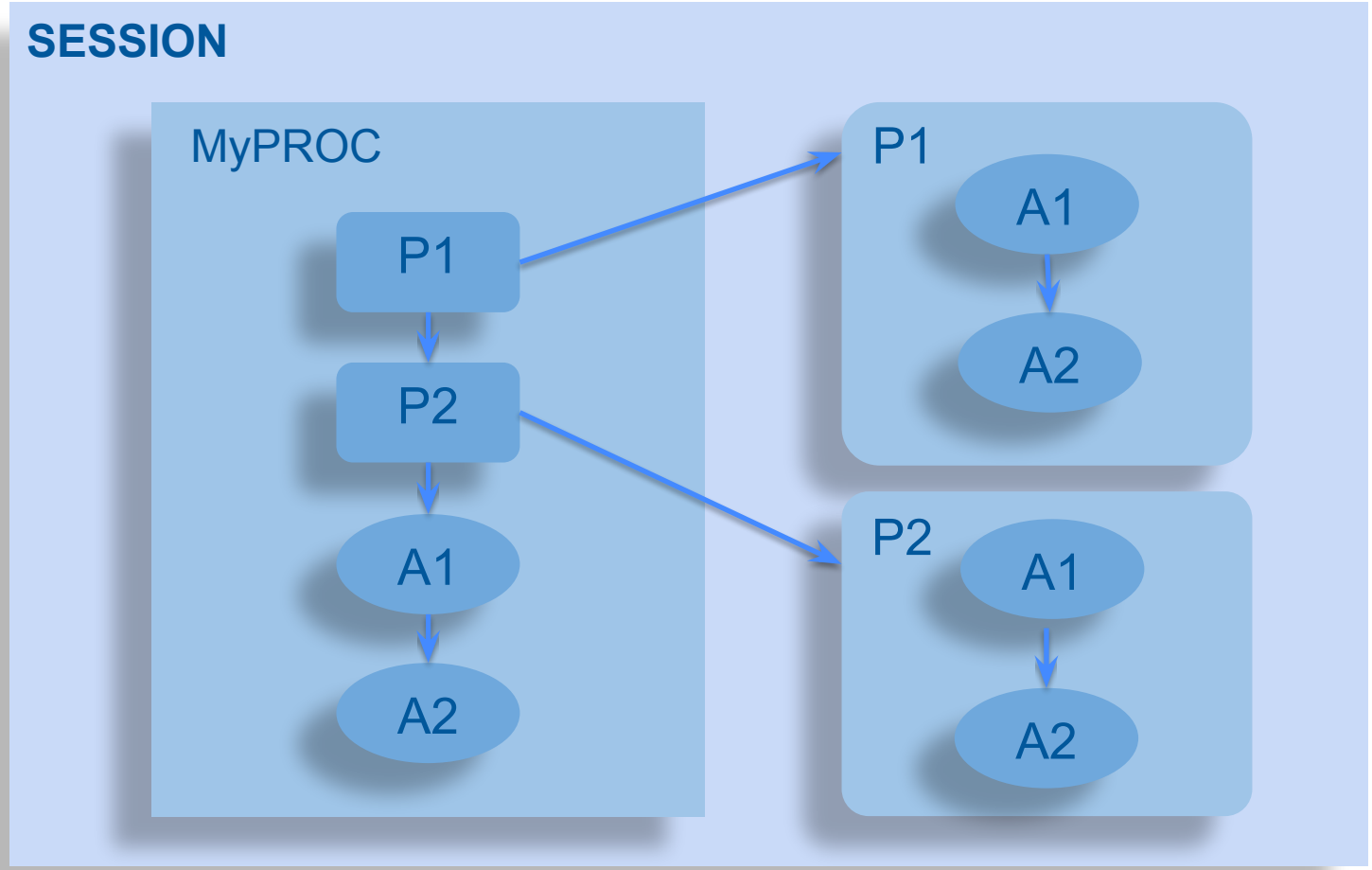
It is possible to get the value of the variables during the execution, with the following syntax:

${*<PATH_OF_THE_VARIABLE>*}$

- where *<PATH_OF_THE_VARIABLE> is the complete path of the variable*
- including the name of the variable

Managing variables and statistics

Consider the following process



SESSION is an execution instance of the process named '*MyPROC*'

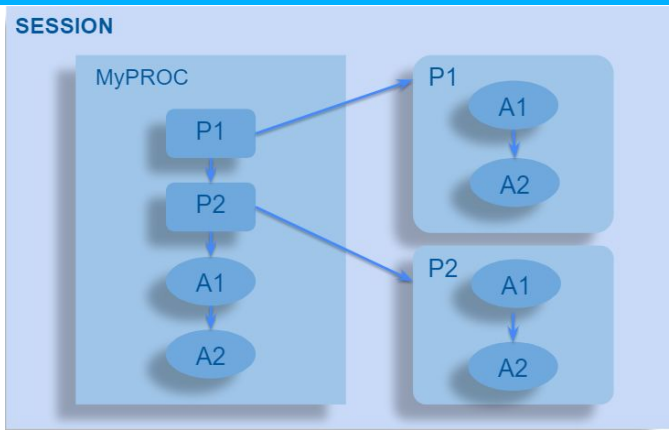
The *MyPROC* process calls two sub processes P1 and P2

When executing *MyPROC*, a session will be created (and visible in the Session and Session Detail views)

Each step (session, process, action, sub-process, …) will generate variables

Managing variables and statistics

Example (1/2) :



- ${**/MyPROC/**A1/CORE_DURATION}$

  Gives the duration of the action A1 of the MyPROC process (main process)

- ${**/MyPROC/P1/**A1/CORE_DURATION}$

  Gives the duration of the action A1 of the MyPROC/P1 process (sub process)

- ${**~/**A1/CORE_DURATION}$ *: ~/ gets the path of the main process*

  Gives the duration of the action A1 of the MyPROC process.

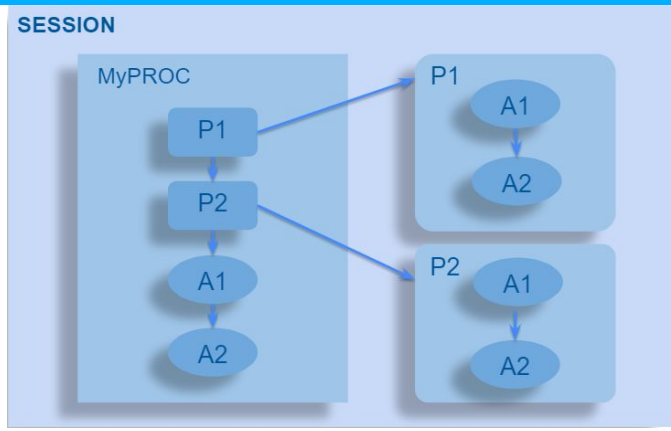
- ${**../**A1/CORE_DURATION}$

  Gives the duration of the action A1 of the parent process

  Used on the MyPROC/A2 action, this will give the duration of MyPROC/A1

  Used on the MyPROC/P2/A2 action, this will give the duration of MyPROC/P2/A1

Managing variables and statistics



Example (2/2) :

- ${/CORE_TEMPORARY_FOLDER}$

  Gives the variable CORE_TEMPORARY_FOLDER on the session level (Session Detail view)

When you are on a link, it's like you are on the parent process !

- ${./A1/CORE_DURATION}$ > 300

  This expression is an execution condition on a link in a process containing an A1 action

  During the execution, there will be an evaluation of the duration of A1 and the destination action of the link will be executed only if the condition is true

Managing variables and statistics

To get the statistics of a specific variable, it is possible to use some scripting

The syntax is the following :

%e(**language**){ *<language code>*}e(**language**)%

Where the **language** can be:

- Rhino (javascript),
- Jython,
- Groovy,
- …

Sample: %e(**rhino**){__ctx__.sumVariable("SQL_STAT_INSERT")}e(**rhino**)%

Managing variables and statistics - Samples

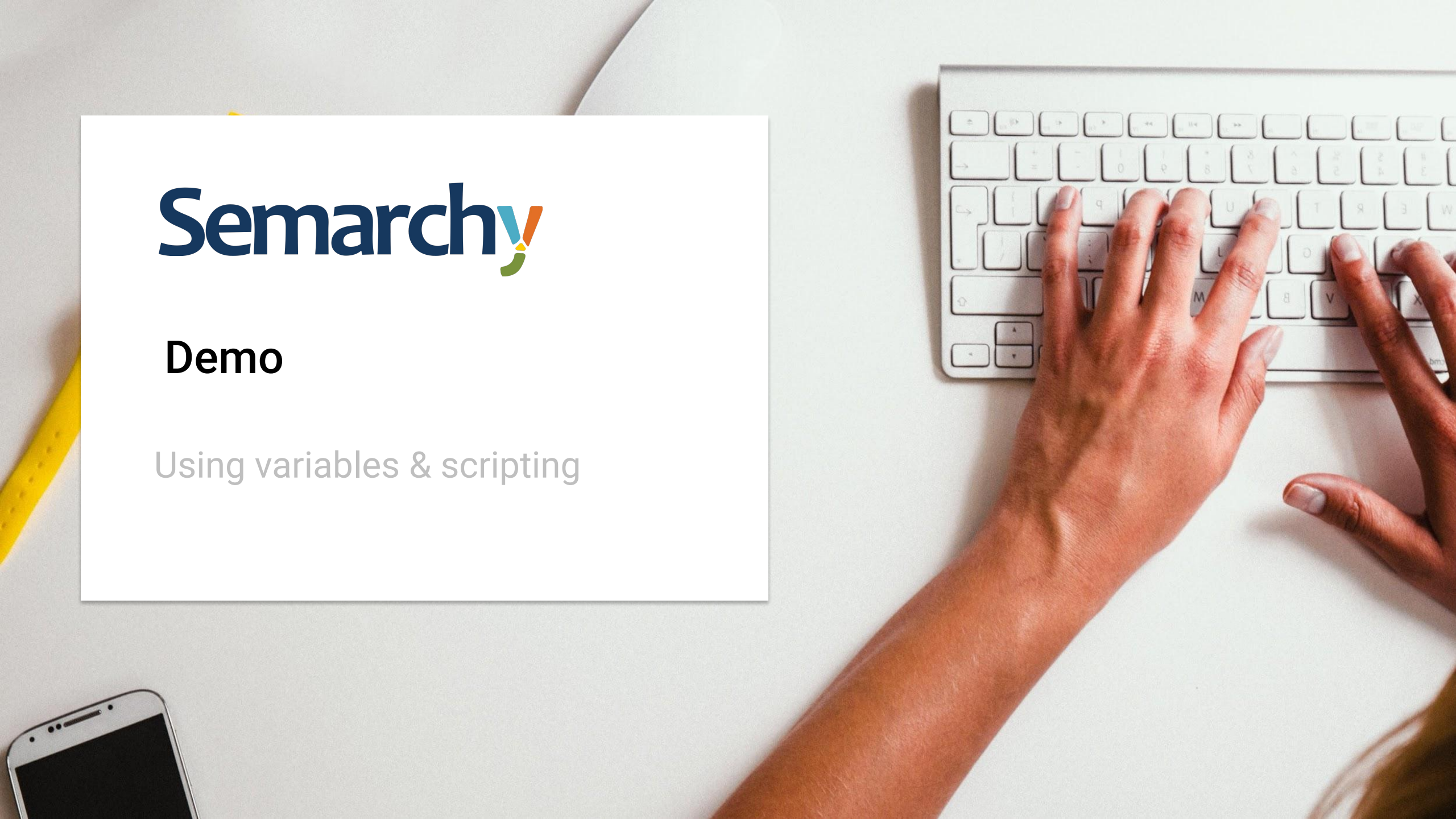
- %e(**rhino**){__ctx__.sumVariable("SQL_STAT_INSERT")}e(**rhino**)%
  Gives the sum of the SQL_STAT_INSERT variables for all the process

- %e(**rhino**){__ctx__.sumVariable("SQL_STAT_UPDATE“,”../P1”)}e(**rhino**)%
  Gives the sum of the SQL_STAT_UPDATE variables for the sub-process P1

- %e(**jython**){import time; __ctx__.retValue =time.strftime("%H_%M_%S")}e(**jython**)%
  Gives the current time

- %e(**groovy**){__ctx__.retValue = new Date().format('yyyyMMdd')}e(**groovy**)%
  Gives the current date

__ctx__ is a java object. For more information on this object and the existing public functions, consult the documentation of the product



Demo

Using variables & scripting

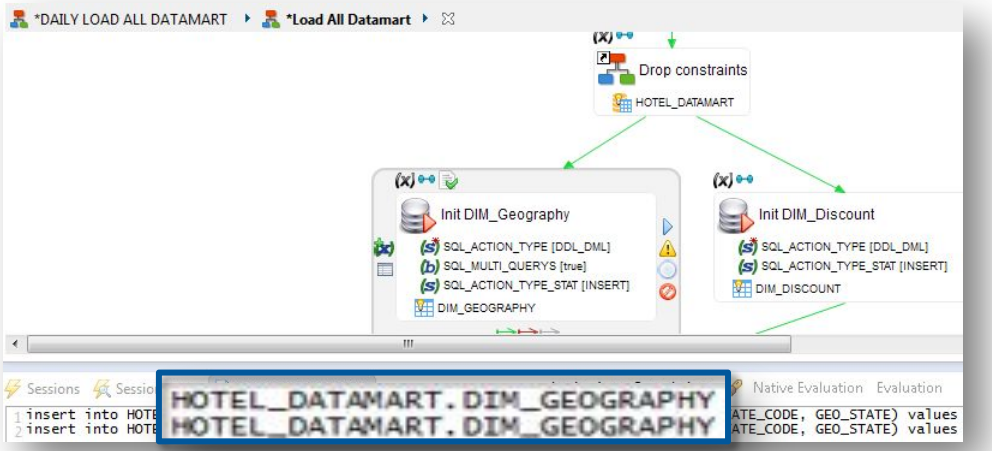
Semarchy

Related tutorial exercises

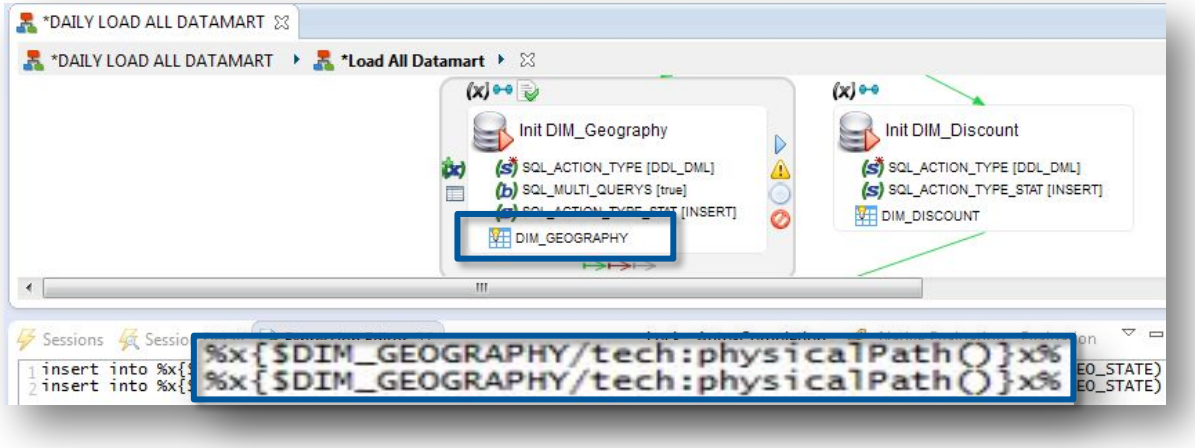
Scripting

Use xPath expression

Don’t hard code the reference of a table, as the schema could change on another environment



Use an xPath expression referencing the metadata link



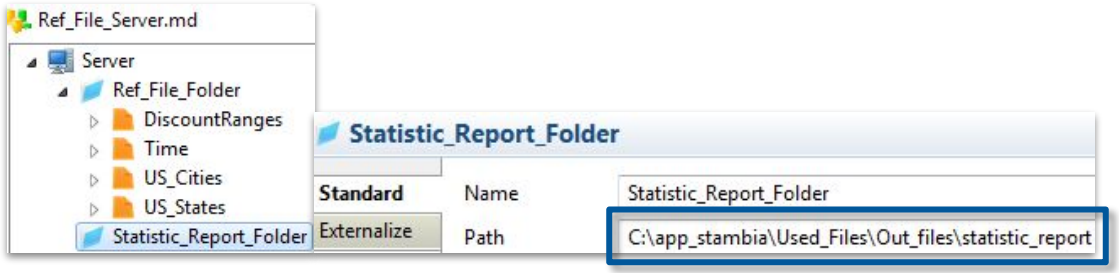
In a Process, XPath expressions can be enclosed by **%x{ … }x%** tags

During the Preparation phase the expression is evaluated and the resulting String is substituted

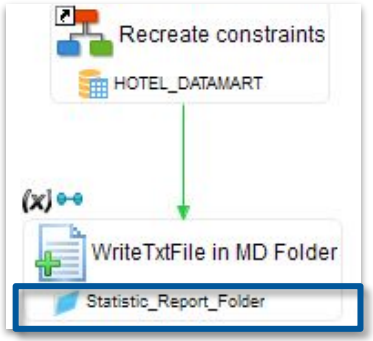
Use xPath expression

Never hard code a path : it could be different in another environment !

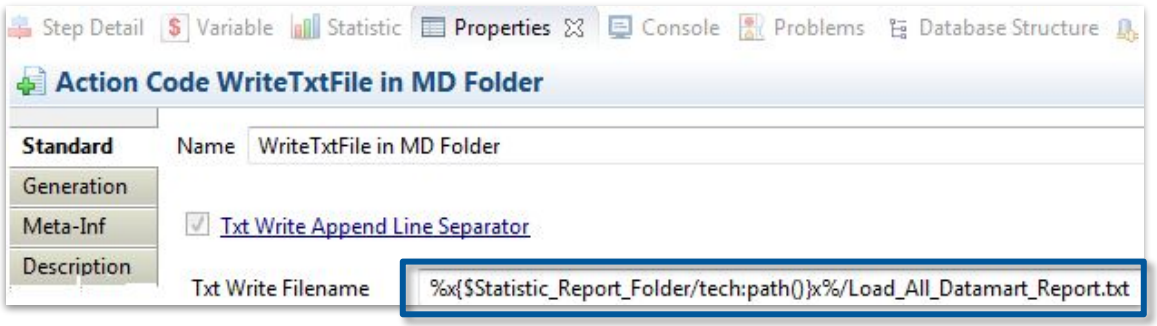
1. Create a folder in a “File Server” Metadata



2. Drag & Drop the Folder Metadata link in the process or in an action



3. Refer the folder using an xPath expression to retrieve the path property



To go further

Document Type	Link
Stambia.org article Using file wait to check files presence	https://stambia.org/doc/102-development-hints-and-tips/actions/file/wait-for-files/159-using-file-wait-to-check-files-presence-and-act-in-consequence
Stambia.org article SQL operation and SQL action type	https://stambia.org/doc/104-development-hints-and-tips/actions/sql/sql-operation/163-sql-operation-and-sql-action-type
Stambia.org article Using transaction with SQL operation	https://stambia.org/doc/104-development-hints-and-tips/actions/sql/sql-operation/191-using-transactions-with-sql-operation
Stambia.org article Operating system command	https://stambia.org/doc/110-development-hints-and-tips/actions/miscellaneous/operating-system-command/197-getting-the-response-of-an-operating-system-command
Stambia.org article How to bind variables to a sub-process	https://stambia.org/doc/80-development-hints-and-tips/how-to/192-how-to-bind-variables-to-a-sub-process
Stambia.org article How to force a process to end with errors	https://stambia.org/doc/80-development-hints-and-tips/how-to/88-how-to-force-a-process-to-end-with-errors
Stambia.org article How to execute a process only when file exists	https://stambia.org/doc/80-development-hints-and-tips/how-to/67-how-to-execute-a-process-only-when-the-input-file-exists
Stambia.org article How to iterate a process on a list of files	https://stambia.org/doc/80-development-hints-and-tips/how-to/19-how-to-iterate-a-process-on-a-list-of-files
Stambia.org article Limit the number of delivery parallel execution	https://stambia.org/doc/80-development-hints-and-tips/how-to/21-how-to-limit-the-number-of-parallel-executions-for-a-delivery
Stambia.org article Variables and parameters – Different syntaxes	https://stambia.org/doc/85-development-hints-and-tips/variables-and-parameters/syntax/214-overview-of-different-syntaxes
Stambia.org article Re-using variables and parameters from an action	https://stambia.org/doc/87-development-hints-and-tips/variables-and-parameters/how-to/91-re-using-variables-and-parameters-from-a-previous-action
Stambia.org article Variable and parameters in a mapping	https://stambia.org/doc/87-development-hints-and-tips/variables-and-parameters/how-to/2-how-to-use-a-process-parameter-in-a-mapping


Semarchy
Questions?