

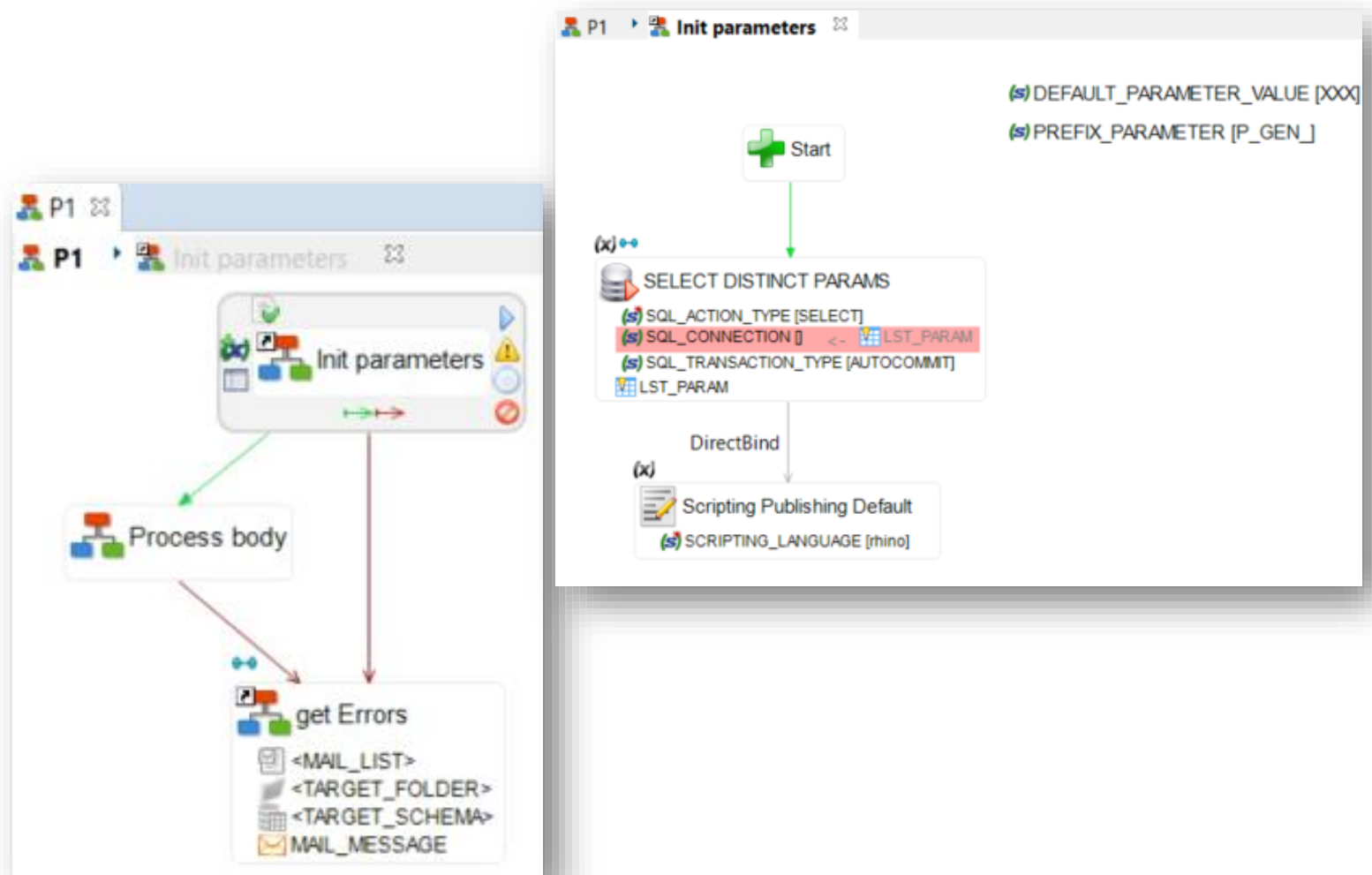
ADV08 Parameters in DB

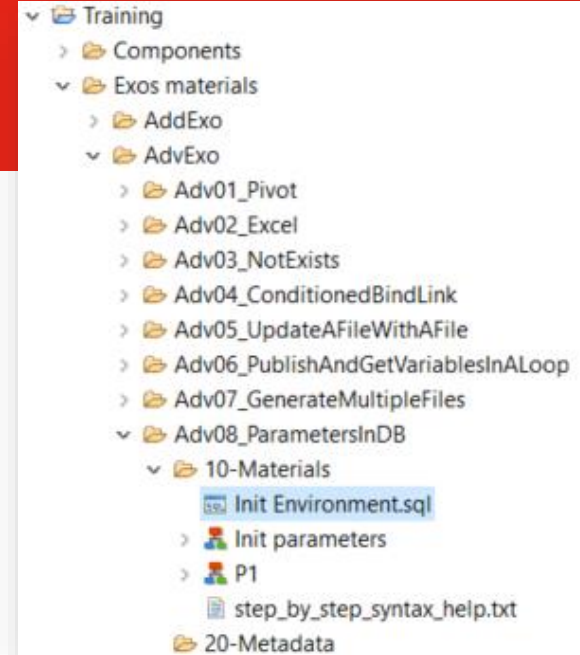


Related articles in Stambia.org

- <https://stambia.org/doc/85-development-hints-and-tips/variables-and-parameters/syntax/214-overview-of-different-syntaxes>
- <https://stambia.org/doc/140-technology-articles/mail/704-how-to-attach-a-list-of-files-to-an-email-based-on-the-content-of-a-folder>

- ❖ Context : Build a treatment (sub-process) to use in each process
 - Read parameters stored in database and publish them
 - Use those parameters in different processes





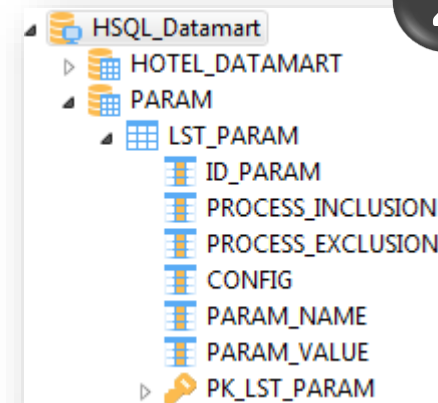
1

❖ 1 - Execute the SQL file (Init Environmentsql) to create/load the required table on a Datamart connection

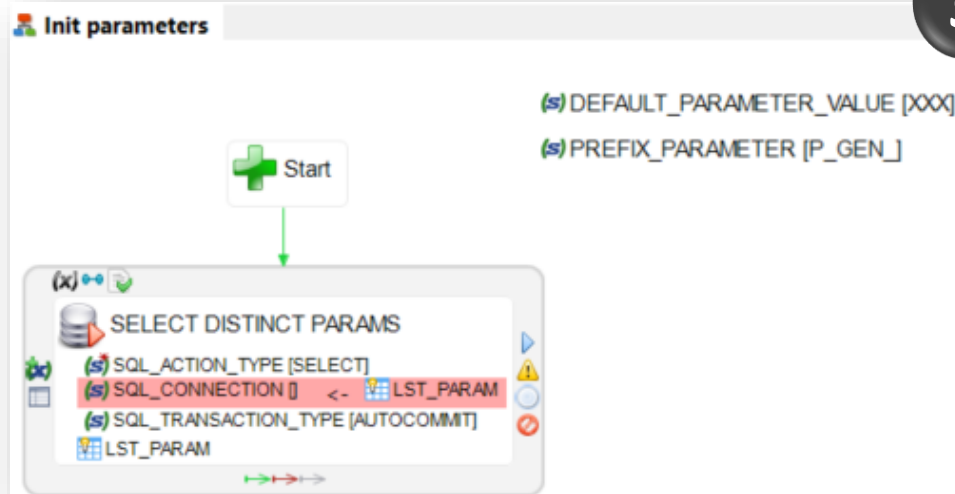
```
Init Environment.sql
Execute current SQL. Current SQL is the selected text or the complete file content if nothing is selected.
CREATE SCHEMA PARAM;
CREATE TABLE PARAM.LST_PARAM (
  ID_PARAM INTEGER,
  PROCESS_INCLUSION VARCHAR(500),
  PROCESS_EXCLUSION VARCHAR(500),
  CONFIG VARCHAR(20),
  PARAM_NAME VARCHAR(100),
  PARAM_VALUE VARCHAR(100),
  CONSTRAINT PK_LST_PARAM PRIMARY KEY (ID_PARAM));
INSERT INTO PARAM.LST_PARAM VALUES (1,'ALL',NULL,'ALL','SLEEP_PERIOD','2');
INSERT INTO PARAM.LST_PARAM VALUES (2,'P1',NULL,'Default','DIRECTION_DECISION','LEFT');
INSERT INTO PARAM.LST_PARAM VALUES (3,'P2',NULL,'Default','DIRECTION_DECISION','RIGHT');
INSERT INTO PARAM.LST_PARAM VALUES (5,'ALL','P2','Default','CONDITIONAL_ACTION','TRUE');
INSERT INTO PARAM.LST_PARAM VALUES (7,'P1,P2',NULL,'ALL','ITERATION','2');
INSERT INTO PARAM.LST_PARAM VALUES (8,'P3',NULL,'ALL','ITERATION','3');
```

2

❖ 2 - Reverse the schema and the table



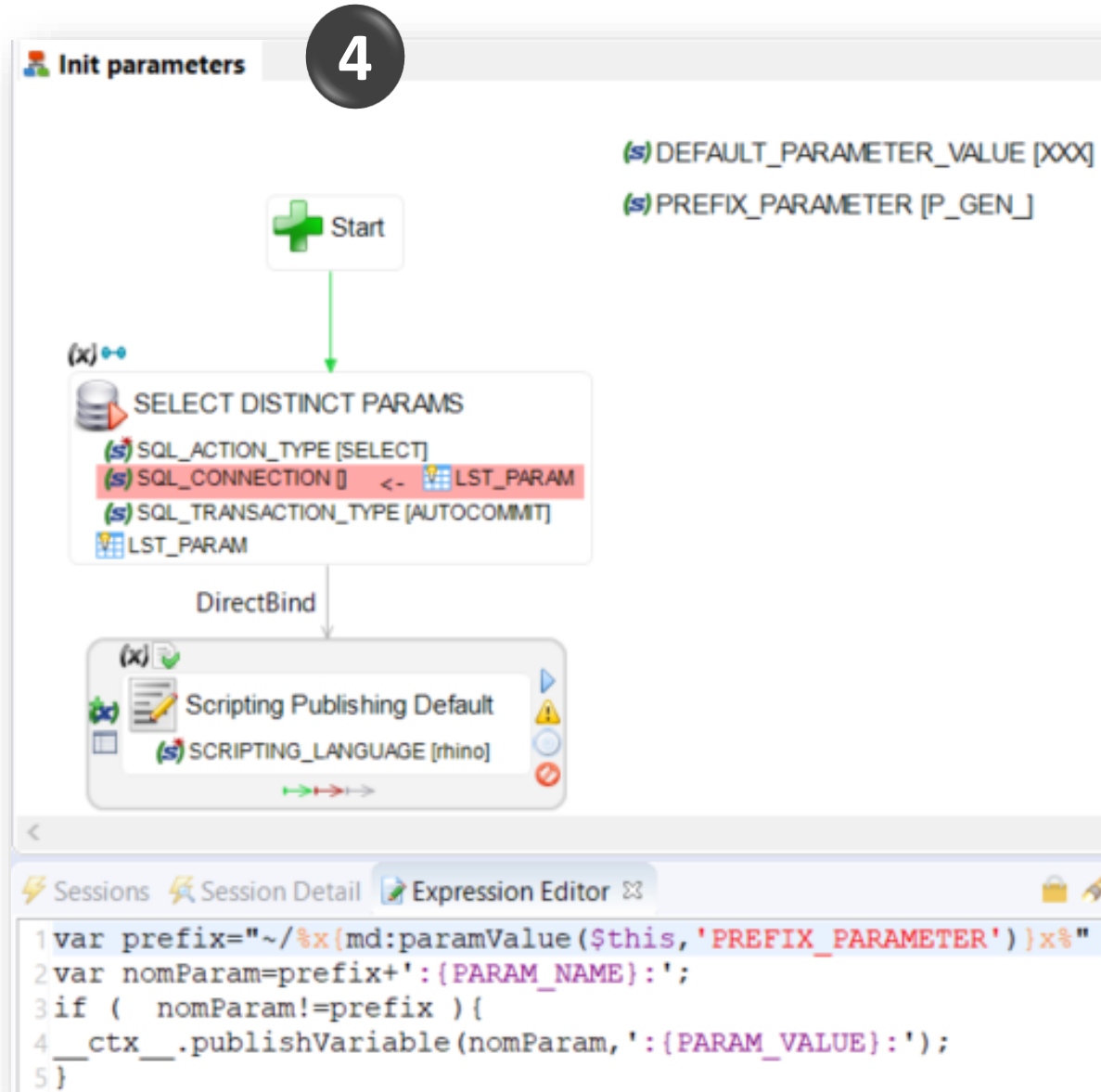
- ❖ 3 – Add in the process, named “init parameters” two parameters and a SQL Operation to select the parameters



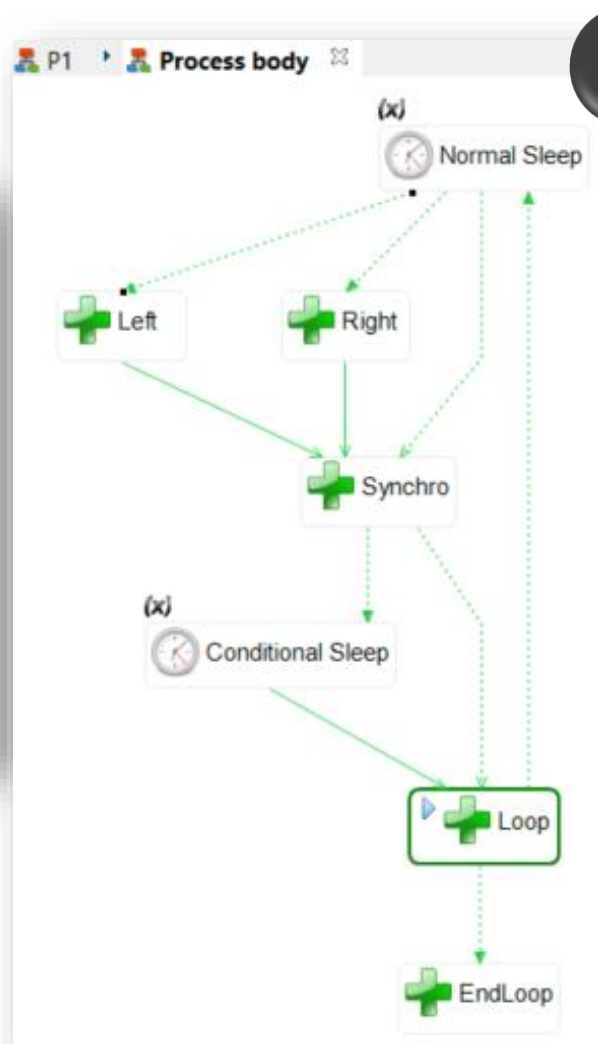
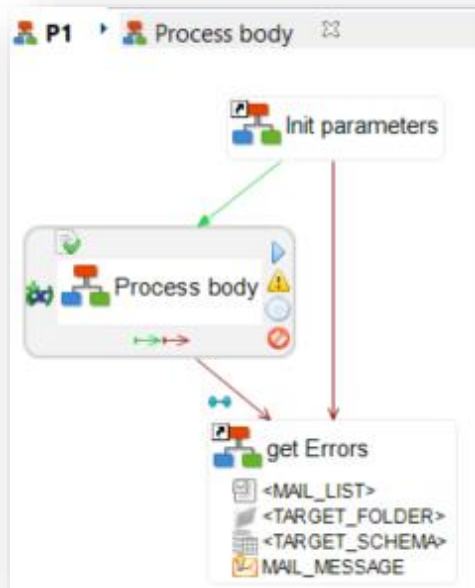
```

1 SELECT COALESCE(PARAM_REQ_NAME,PARAM_ALL_NAME) PARAM_NAME,
2        COALESCE(PARAM_REQ_VALUE,PARAM_ALL_VALUE) PARAM_VALUE
3 FROM   (SELECT DISTINCT PARAM_NAME PARAM_ALL_NAME,
4             '%x{md:paramValue($this,'PREFIX_PARAMETER')}x%' PARAM_ALL_VALUE
5         FROM   PARAM.LST_PARAM)
6 LEFT  OUTER JOIN
7       (SELECT PARAM_NAME PARAM_REQ_NAME,
8             PARAM_VALUE PARAM_REQ_VALUE
9         FROM   %x{$LST_PARAM/tech:physicalPath()}x%
10        WHERE  ( PROCESS_INCLUSION='ALL'
11                OR PROCESS_INCLUSION LIKE '%${/CORE_SESSION_NAME}$%')
12        AND    ( PROCESS_EXCLUSION IS NULL
13                OR PROCESS_EXCLUSION NOT LIKE '%${/CORE_SESSION_NAME}$%')
14        AND    ( CONFIG='ALL'
15                OR CONFIG='${/CORE_SESSION_CONFIGURATION}$%'))
16 ON PARAM_ALL_NAME=PARAM_REQ_NAME
    
```

- ❖ 4 - Add a Java Native scripting action
 - Add a bind link between the two actions



- ❖ 5 - Study P1 process to understand the use of the parameters :



5

Execution	Execution
Generation	Execution Type Successful
Meta-Inf	Triggering Behavior Mandatory
Description	Condition
Core	'\$~/P_GEN_DIRECTION_DECISION)\$' == 'LEFT'

- ❖ 6 - Duplicate P1 process twice and name the new processes “P2” and “P3”
 - Execute each, check the value of the parameters and what happen in each process

