Exploitation Book

Prod.com proxy

	Interlocuteur principal	Backup
MOA		
IP / Développeur	Antoine ELU	Vincent DURIEZ
Prod.com / OpenEx	Aubin VANDENBILCKE	Rim HAMMOUDA
SAP PO	Rémy CLEMENT	Miguel SIERRA CHAVEZ /Imane ABDELKAOUI

Watch tower : http://supplymonitoring.subsidia.org/en-US/app/search/bi_flux_prodcom_general_

Summary

Table of contents :

Context

Due to the migration of MQ, We are using messaging applications OpenEx & PO.
OpenEx is converting the incoming flat file to Json Format and sends it to PO.
PO will load it in BW via qRFC and proxies

Flow / Indicator description.

Prod.com proxies are used for several flows:

BW flow	Prod.com Batch	Flow name	Table name	Infoprovider
Simple flow				
BPILJ	ComponentToBatch	PRDComponentTo	ZPRDCOMPONENTTO	(Y / Z) MMCACOMP
IPILJ	DotComponentBatch	PRDDotComponent	ZPRDDOTCOMPONENT	(Y / Z) SCMDOTC (Y / Z) SCDOTCO
OPILJ	ResaonOfDelayBatch	PRDReasonOfDelay	ZPRDRESOFDELAY	(Y / Z) SC04DOC
Complex flow				
APILJ	ItemCessionToBatch	PRDItemCessionTo	ZPRDITEMCESSTO	(Y / Z) MMCESSL (Y / Z) MM_SALE
EPILJ	ShipmentOnTimeBatch	PRDShipmentOnTime	ZPRDSHIPONTIME	(Y / Z) SCMSOT (Y / Z) SCSOT
GPILJ	PriceDecreaseBatch	PRDPriceDecrease	ZPRDPRICDECREASE	(Y / Z) MMBSPRIC (Y / Z) MMBSPRIX
JPILJ	SendDynamicStockToBW	PRDDynamicStock	ZPRDDYNAMICSTOCK	(Y / Z) JPILJ (Y / Z) MMSTKPIL
MPILJ	EDDOnTimeBtach	PRDEDDOnTime	ZPRDEDDONTIME	(Y / Z) ODSEDD (Y / Z) EDDONTIM
DPILJ	WorldStock	PRDComponentsWorldStock	ZPRDSTOCK	(Y / Z) MMSTKFGR
DPILJ	WorldStock	PRDComponentsWorldStock	ZPRDSTOCK2CH	(Y / Z) MMSTKSND
DPILJ	WorldStock	PRDComponentsWorldStock	ZPRDSTOCKBKDWN	(Y / Z) MMSTKBRK
HPILJ	SendOrderCostToBW	PRDSendOrderCost	ZPRDSENDORDCOST	(Y / Z) MARGCTRL
KPILJ	ProductionLeadTime	PRDProdLeadTime	ZPRDPRODLEADTIME	(Y / Z) MLEADTIM
CPILJ	OrderRemainderAndForecast	PRDOrderRemainderAndForecast	ZPRDORDFORECAST	(Y / Z) ODSPREV (Y / Z) PREVPRD (Y / Z) OPNPRDWK
CPILJ			ZPRDORDREMINDER	(Y / Z) ODSOPN (Y / Z) MMOPNPRD (Y / Z) OPNPRDWK
APSTJ	TreeCatalogBatch	PRDTreeCatalog	ZPRDLINKFAMMOD ZPRDFAMARBO ZPRDINTERNATMOD	(Y / Z) MATERDPP (Y / Z) SUBDPDPP (Y / Z) DEPTDPP (Y / Z) FAMILDPP (Y / Z) SECTORDP

There are two types of flow:

1. Simple flows

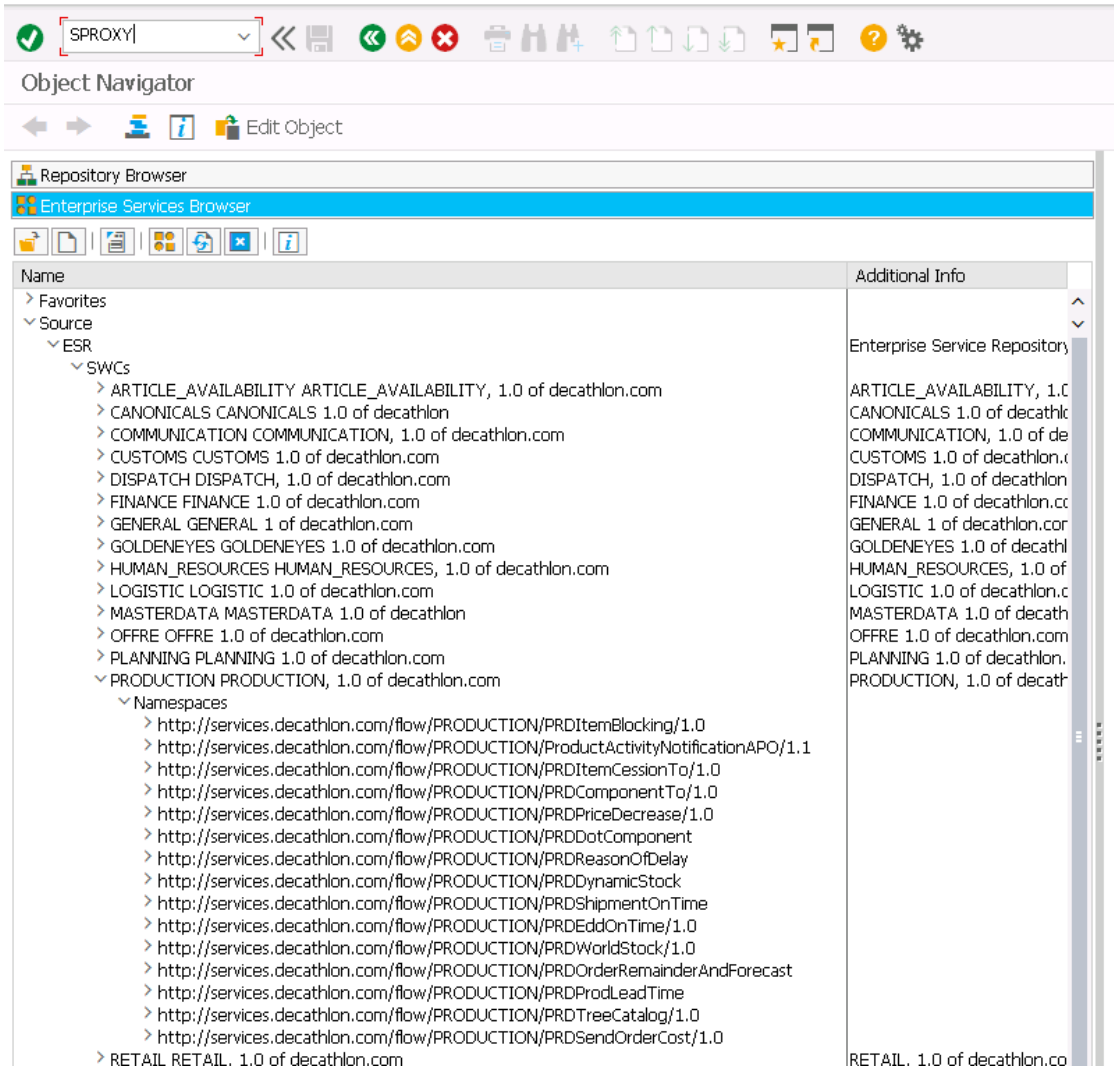
2. Complex flows

This difference is due to an issue constraint.
The number of data passing via OpenEx and SAP PO has to be reduced.
Files are splitted on Prod.com and are reassembled on BW. Before reconstructing these files, BW will check the content consistency.
Simple flows are flows which do not have a huge amount of data, they are sent with only one message for a DPP whereas Complexe flows are heavy and are divided in several messages for one DPP.

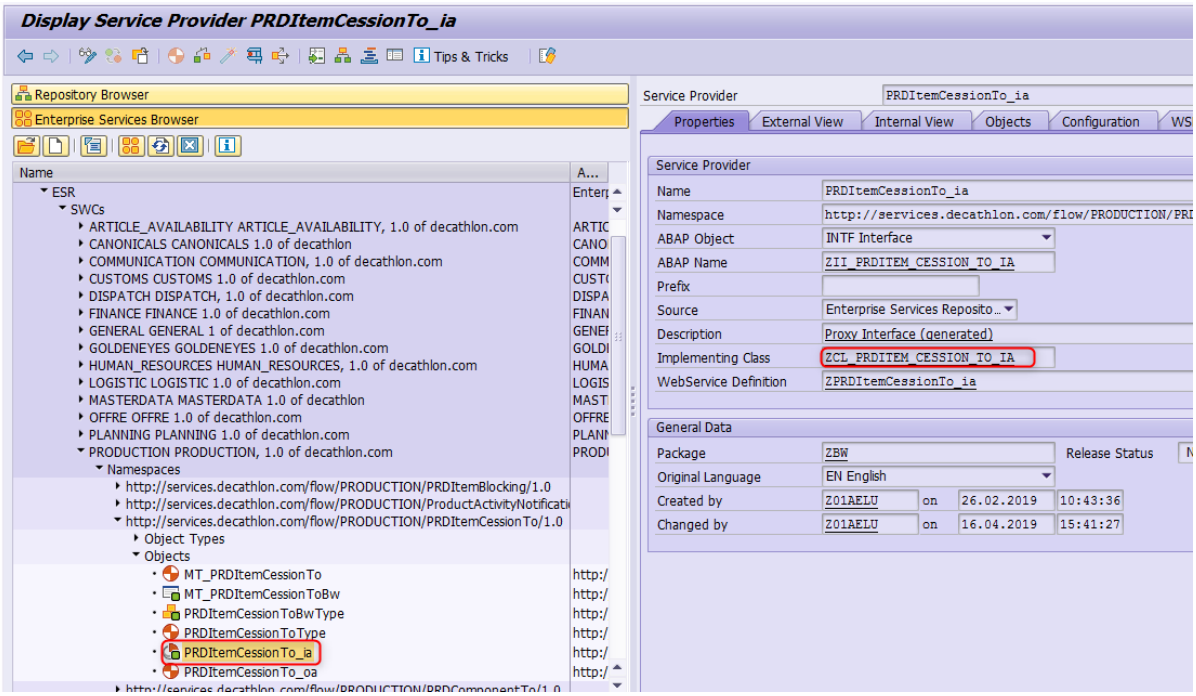
	Simple Flow	Complex Flow
DPP	1	1
Number of messages	1	N messages (N = number of flow lines / number of messages lines)

Proxy information

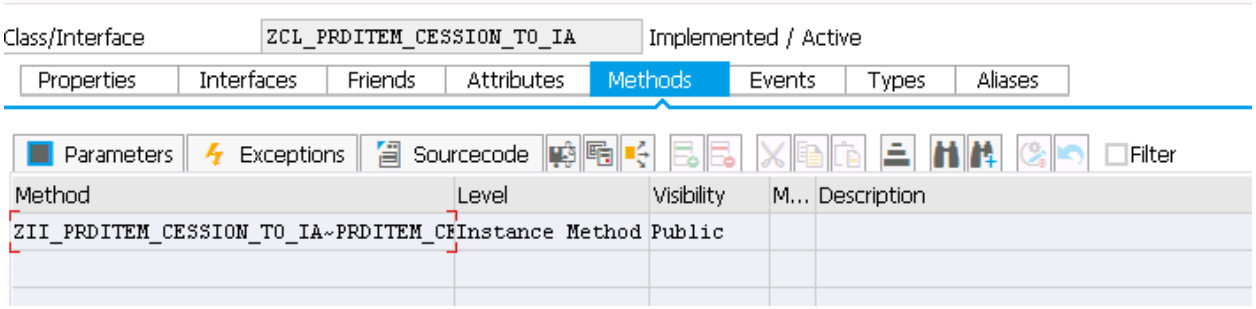
Proxies are available via the transaction code “SPROXY”.
To find the needed proxies, go to Source → ESR → SWCS → PRODUCTION PRODUCTION, 1.0 of decahtlon.com.



To reach the code of the proxies, open the object finishing with “_ia” in Objects part (see screenshot below).
The property tab will show up. All the implemented code to make the proxy run is written in the implementing class (Example below : ZCL_PRDITEM_CESSION_TO_IA).

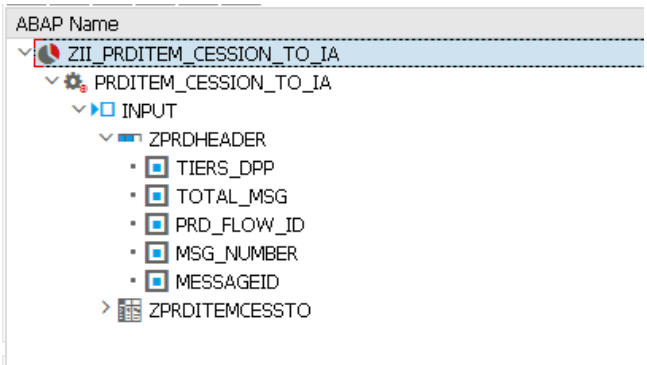


Double clicking on the Implementing class will give the following screen :



The ABAP code is in this method.

In the implementing class there are two different tables used to fill the BW tables



ZPRDHEADER contains all the information about the sent flow.

ZPRDITEMCESSTO contains all the proxy data which will be loaded in the infoproviders.

First step of the proxy ABAP code is to fill the Header table ZPRD_FLOW_HEADER.

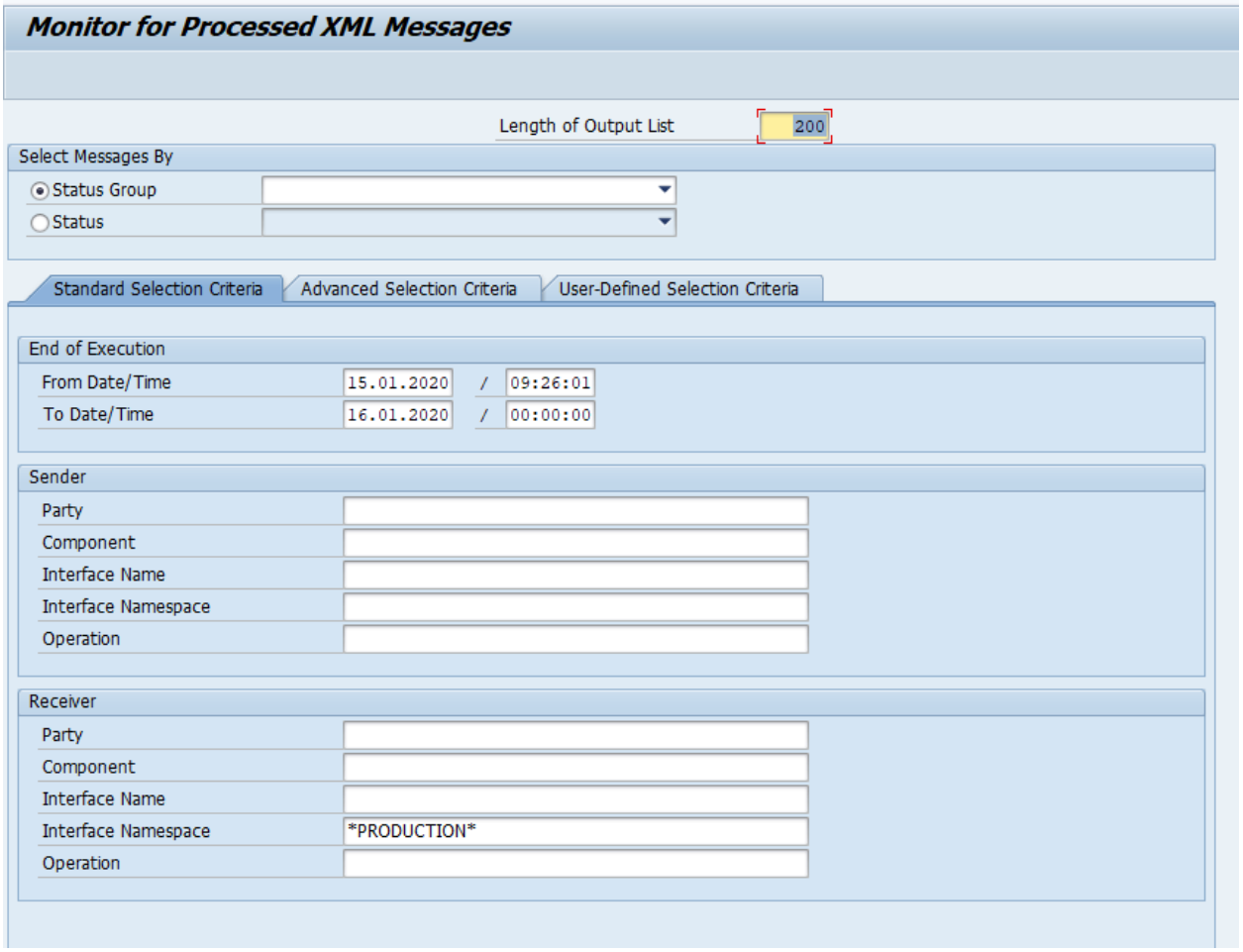
```
* Variable Declaration
    v_msg_number = input-mt_prditem_cession_to_bw-zprdheader-msg_number.
    v_msg_id = input-mt_prditem_cession_to_bw-zprdheader-prd_flow_id.
    v_tiers_dpp = input-mt_prditem_cession_to_bw-zprdheader-tiers_dpp.
    v_msg_total = input-mt_prditem_cession_to_bw-zprdheader-total_msg.
    v_msg_name = input-mt_prditem_cession_to_bw-zprdheader-messageid.

* BW DPP recovery
    SELECT /bic/yvendor
      FROM /bic/pysoutiers
      INTO v_vendor
      WHERE
      /bic/ytyptiers = '002'
      AND /bic/ytiers = v_tiers_dpp
      AND /bic/ysoutiers = v_tiers_dpp.
    ENDSELECT.

* Header table alimentation
    wa_header-zflow = 'APILJ'.
    wa_header-zprddpp  = v_tiers_dpp.
    wa_header-zbwdpp  = v_vendor.
    wa_header-zuniqueid = v_msg_id.
    wa_header-znbmessages = v_msg_total.
    wa_header-ztype = ''.

* Vérification du chargement en double d'un batch
    SELECT * FROM zprd_flow_header INTO ls_doublon WHERE zflow = 'APILJ' AND zprddpp =
      v_tiers_dpp AND zstatusflow = 0.
    ENDSELECT.
    IF sy-subrc = 0.
      IF ls_doublon-zuniqueid = v_msg_id.
        v_statusflow ='1'.
      ELSE.
        wa_header-zstatusflow = 1.
        v_statusflow = '9'.
      ENDIF.
    ELSE.
      wa_header-zstatusflow = 0.
      v_statusflow = '1'.
    ENDIF.

    APPEND wa_header TO it_header.
```

Second step is to map all the data to the fields of the BW table.

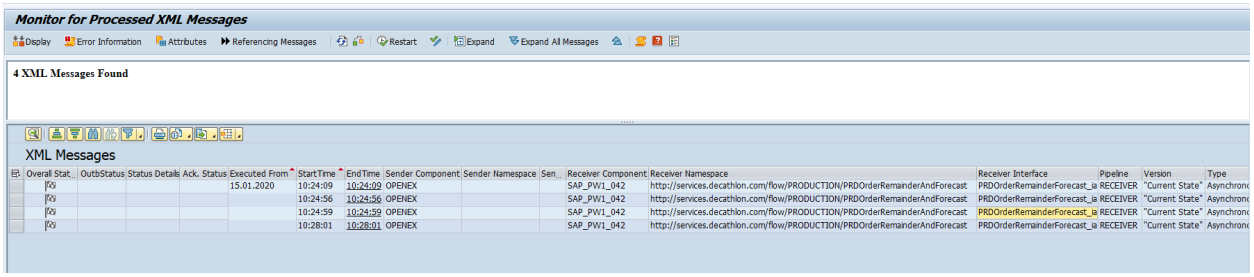
```
    v_counter = 0.
    LOOP AT input-mt_prditem_cession_to_bw-zprditemcessto INTO ls_body.

      v_counter = v_counter + 1.
      ls_apilj-ztyptiedp = ls_body-ztyptiedp.
      ls_apilj-z_sign_1 = ls_body-z_sign_1.
      ls_apilj-ztiedpp = ls_body-ztiedpp.
      ls_apilj-z_sign_2 = ls_body-z_sign_2.
      ls_apilj-zsoutiedp = ls_body-zsoutiedp.
      ls_apilj-z_sign_3 = ls_body-z_sign_3.
      ls_apilj-calday = ls_body-calday.
      ls_apilj-z_sign_4 = ls_body-z_sign_4.
      ls_apilj-doc_num = ls_body-doc_num.
      ls_apilj-z_sign_5 = ls_body-z_sign_5.
      ls_apilj-material = ls_body-material.
      ls_apilj-z_sign_6 = ls_body-z_sign_6.
      ls_apilj-zdeliv_nb = ls_body-zdeliv_nb.
      ls_apilj-z_sign_7 = ls_body-z_sign_7.
      ls_apilj-zcesstype = ls_body-zcesstype.
      ls_apilj-ztyptiebu = ls_body-ztyptiebu.
      ls_apilj-z_sign_8 = ls_body-z_sign_8.
      ls_apilj-ztiebuy = ls_body-ztiebuy.
      ls_apilj-z_sign_9 = ls_body-z_sign_9.
      ls_apilj-zsoutiebu = ls_body-zsoutiebu.
      ls_apilj-z_sign_10 = ls_body-z_sign_10.
      ls_apilj-ztyptiers = ls_body-ztyptiers.
      ls_apilj-z_sign_11 = ls_body-z_sign_11.
      ls_apilj-ztiers = ls_body-ztiers.
      ls_apilj-z_sign_12 = ls_body-z_sign_12.
      ls_apilj-zsoutiers = ls_body-zsoutiers.
      ls_apilj-z_sign_13 = ls_body-z_sign_13.
      ls_apilj-zprodtyp = ls_body-zprodtyp.
      ls_apilj-zmmsalqty = ls_body-zmmsalqty.
      ls_apilj-z_sign_14 = ls_body-z_sign_14.
      ls_apilj-zmmsales = ls_body-zmmsales.
      ls_apilj-z_sign_15 = ls_body-z_sign_15.
      ls_apilj-zmmmargi = ls_body-zmmmargi.
      ls_apilj-z_sign_16 = ls_body-z_sign_16.
      ls_apilj-loc_currcy = ls_body-loc_currcy.
      ls_apilj-z_dummy_1 = ls_body-z_dummy_1.
      ls_apilj-ztyptieve = ls_body-ztyptieve.
      ls_apilj-z_sign_17 = ls_body-z_sign_17.
      ls_apilj-ztieven = ls_body-ztieven.
      ls_apilj-z_sign_18 = ls_body-z_sign_18.
      ls_apilj-zsoutieve = ls_body-zsoutieve.
      ls_apilj-z_sign_19 = ls_body-z_sign_19.
```

Check proxy loading

You can use the transaction SXMB_MONI to check the SAP PO flow integration.
“Monitor for Processed XML Messages”

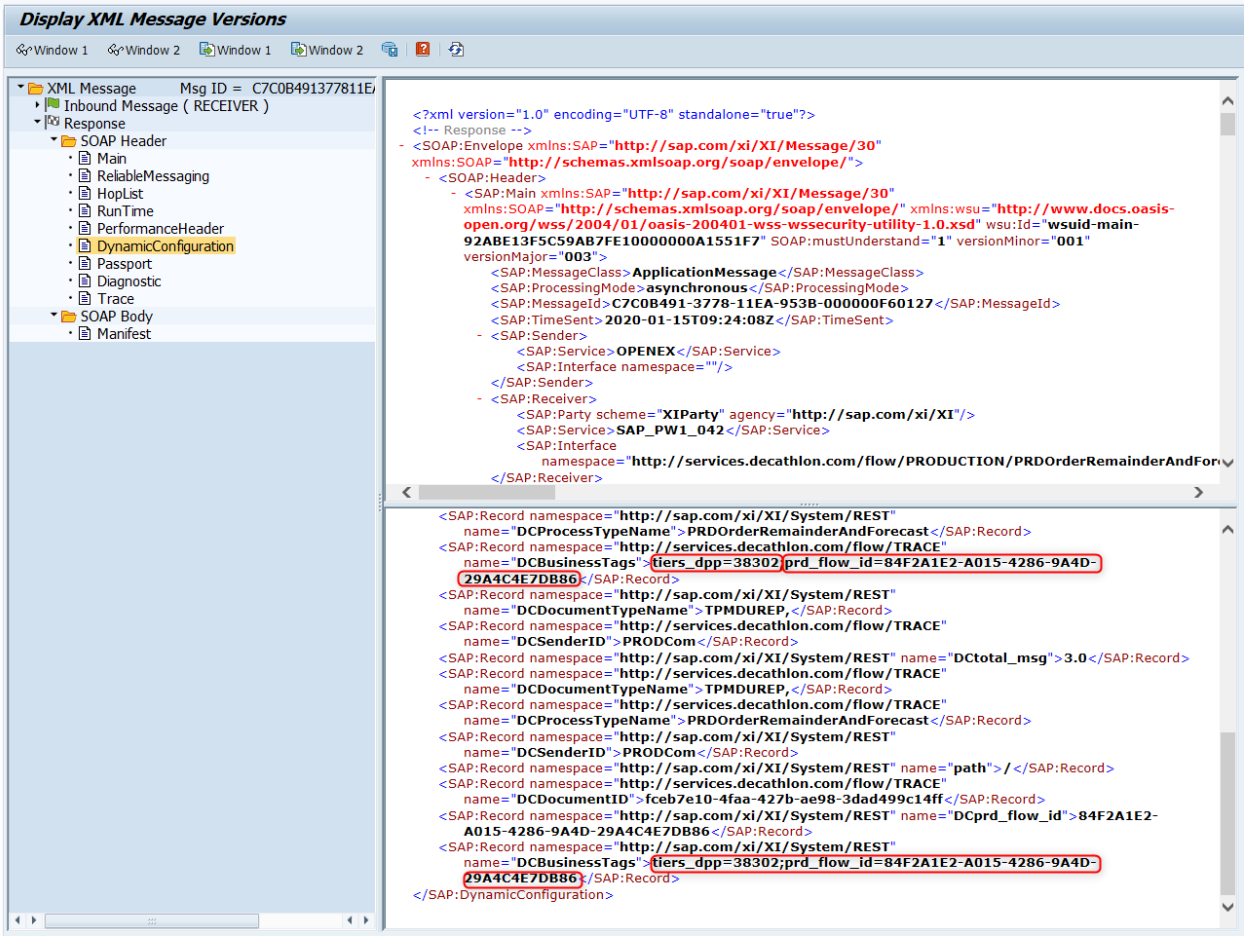


All the Prod.com flows are tagged as “PRODUCTION” so you can filter the Interface Namespace with “*PRODUCTION*” to see all these flows.



Then you have all the last proxies loadings.

(You can see all the technical information about a flow by double cliking on it and then check the ‘DynamicConfiguration”)



Loadings explanation

Glossary

Difference between Message and Flow :

One flow is the data given by a prod.com batch.
One DPP has one flow but can have multiple messages ID.

Due to performance constraints, the flows are divided in several messages of the same size (xxx lines).
This means a flow is able to contain a few messages

Flow ID (Unique ID)	Message ID

05459848-8EE5-4E47-990F-74050BE69149	0042953a-bba7-49f1-bdbd-bbafdc264236
	30bece77-4afd-4aa8-b2fc-89d7c4972f21
	6d24602e-1829-4d60-9ee4-e716b19bc4ce

Proxy

Records are sent via PROD.COM into BW Highway thanks to the proxies.

There are 2 steps in the Proxy process:

1. ZPRD_FLOW_HEADER

In the proxies, each DPP is verified at the beginning of the process.
The goal is to check if the DPP which is being sent has already been received. If it is the first time, the status will be 0. If not, status will be 1 as duplicated.
This avoids data duplication during loadings in Infoproviders but is by the way integrated in the table in case the restart is normal.

With this check, a BW table (ZPRD_FLOW_HEADER) will be updated with all the flows information

Field	Description	Use	Example
ZFLOW	Prod.com Data Flow	Name of Prod.com flow	APILJ
ZTYPE	Type	Type of flow	
ZPRDDPP	Prod.com DPP	Type Tiers du DPP	02157
ZBWDPP	Vendor	Code vendor	0000090540
ZUNIQUEID	Unique ID	Key used to verify if the sent file is unique	A357ED75-2EE4-431C-9CC5-C92CEF1A336D
ZNBMESSAGES	Number of messages for a flow	Number of messages sent in the proxy after Prod.com cut (Number of messages for a flow + Flow Type + DPP)	150

ZNBTABLEMESSAGES	Number of messages integrated in the staging table	Number of messages integrated in the BW table	150
ZNBINTMESSAGES	Number of messages integrated in target infoprovider	Number of messages loaded in the Infoprovider	150
ZSTATUSFLOW	Status of Unique Id by Flow	Check if sent data is duplicated or not	0

2. Body table

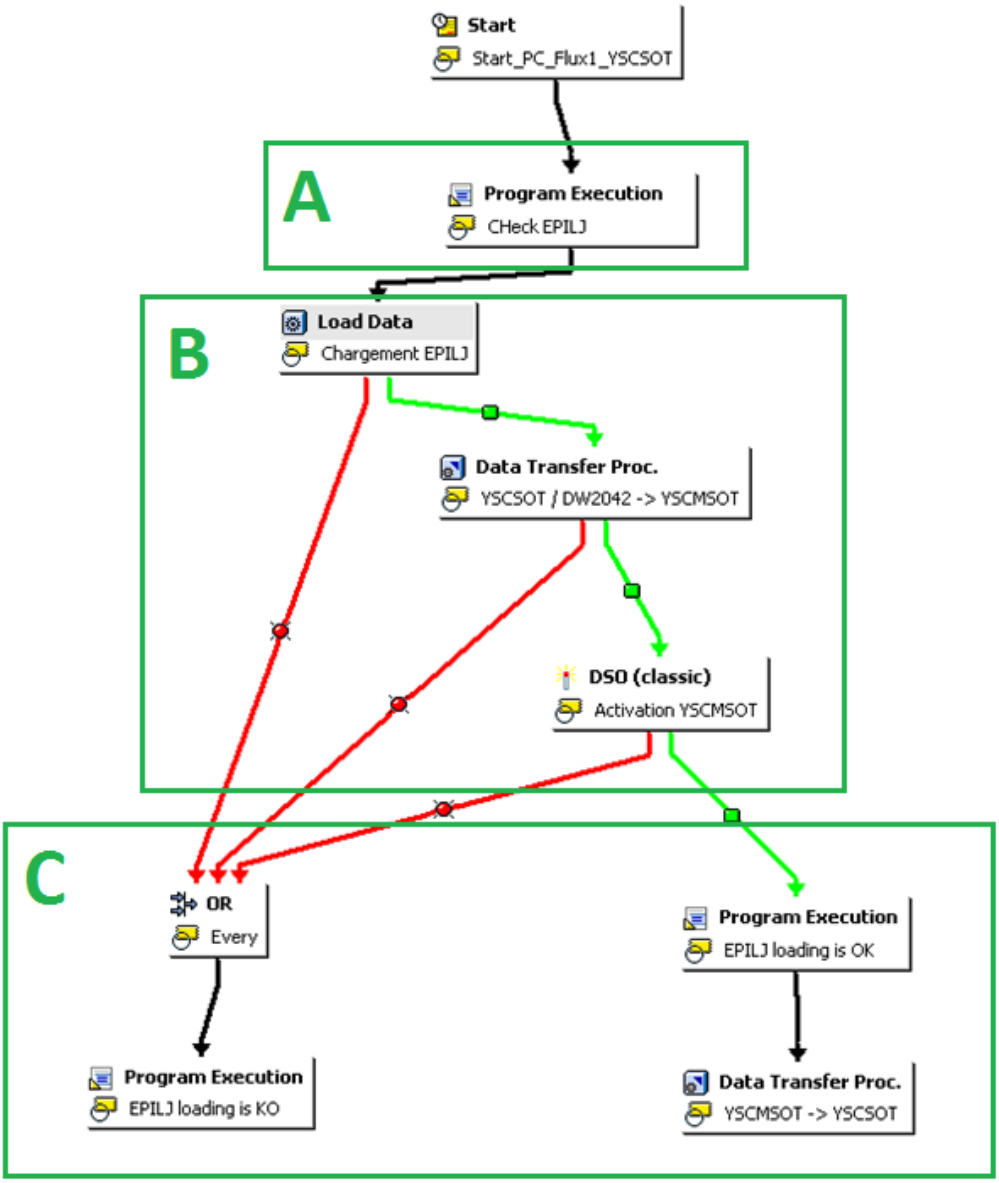
Mapping is done in the proxy and corresponding table of the flows will be filled.
Some fields are added for the flow administration:

Field	Use	Comment
MESSAGEID	Message ID	ID of the sent Message (different from Unique ID which is particular to the flox
COUNTER	Number of lines integrated in the table	This counter allows to check the unicity of each line
MESSAGENB	Message Number	Message number
UNIQUEID	Unique ID	Flow ID
STATUS	Status of flow	

The status field will be used to check which data can be loaded.
The status can have 4 different values:
1 : New Data is available in the BW table
2 : Data to be loaded is correct : enough messages for BW integration
3 : Data is loaded
4 : Error during loadings. Check needed by support team
9 : Data has been duplicated by a restart of the prod.com batch, Check needed by support team.

Process chain

During the process chain loading, there are 3 different parts:



A. Check if data is ready to be loaded

First step is the report Z_PROD_CALCULTION.

This report is used to check if we received enough messages for each flow before loading them in the infoproviders.

In table ZPRDFLOWPARAM, the name of flows are stored with the minimum percentage needed to load data from the ZPRD* table to the infoprovider.

2 possible cases

1. Number of messages is lower than the percentage.

Status of record will be set to 1.
This means data is still available to be loaded next time, if the percentage is reached.

2. Number of messages is greater or equal than the percentage needed.

Status will be 2 and the number of messages will be stored in table ZPRD_FLOW_HEADER for field ZNBTABLEMESSAGES.

For example, APILJ flow could need 80% of the total number of messages to be loaded.
If we have 75% of messages integrated in the tables, the flow will not be loaded. Status of records will stay to 1.
If 80% is reached, the flow will be loaded. Status of records will be put to 2.

IMPORTANT:
For simple flows (OPILJ, BPILJ, IPILJ), the report Z_PROD_CALCULATION is not used.
The status will be automatically set to 2 or 9 in the proxy.

B. Load data in the infoprovider

As soon as the report is finished, data will be loaded.
Infopackages have a restriction on the status and will load only lines with status = 2.
(Messages validated during the report).

C. Change status of BW table records

During the loading, a check will be done at each step.

Infopackage fails: Status of each lines will be set to 4
DTP fails: Status of each lines will be set to 4
Activation fails: Status of each lines will be set to 4

If the DSO loading succeeds, the status of each line will be set to 3

EXB - Flow 1, 2 & 3 - In case of relaunching

When?

Weekly

Process chain names:

BW flow	Prod.com Batch	Flow name	Table name	Infoprovider	Highway Process chain	Legacy Process chain
Simple flow						
BPILJ	ComponentToBatch	PRDComponentTo	ZPRDCOMPONENTTO	(Y / Z) MMCACOMP	YTD_LOAD_COMP_CESS_SALES	ZHWM_ZMMCACOMP
IPILJ	DotComponentBatch	PRDDotComponent	ZPRDDOTCOMPONENT	(Y / Z) SCMDOTC (Y / Z) SCDOTCO	Y_DOT_COMPOSANT	ZHWM_ZSCDOTCO
OPILJ	ResaonOfDelayBatch	PRDReasonOfDelay	ZPRDRESOFDELAY	(Y / Z) SC04DOC	YTD_SC04DOC1	ZHWM_SC04DOC
Complex flow						
APILJ	ItemCessionToBatch	PRDItemCessionTo	ZPRDITEMCESSTO	(Y / Z) MMCESSL (Y / Z) MM_SALE	YTD_ZMM_SALE	ZHWM_ZMM_SALE
EPILJ	ShipmentOnTimeBatch	PRDShipmentOnTime	ZPRDSHIPONTIME	(Y / Z) SCMSOT (Y / Z) SCSOT	YSCSOT	ZHWM_ZSCSOT
GPILJ	PriceDecreaseBatch	PRDPriceDecrease	ZPRDPRICDECREASE	(Y / Z) MMBSPRIC (Y / Z) MMBSPRIX	YGPILJ_PRICE_DEC	ZHWM_YMMBSPRI
JPILJ	SendDynamicStockToBW	PRDDynamicStock	ZPRDDYNAMICSTOCK	(Y / Z) JPILJ (Y / Z) MMSTKPIL	YLOAD_CUBE_YMMSTKPIL	ZHWM_ZMMSTKPIL
MPILJ	EDDOnTimeBtach	PRDEDDOnTime	ZPRDEDDONTIME	(Y / Z) ODSEDD (Y / Z) EDDONTIM	YSTART_EDD_ON_TIME	ZHWM_EDDONTIME
DPILJ	WorldStock	PRDComponentsWorldStock	ZPRDSTOCK	(Y / Z) MMSTKFGR	YDPILJ_PRODSTOCK	ZHWM_ZMMSTKFGR ZHWM_ZMMSTKSND ZHWM_ZMMSTKBRK
DPILJ	WorldStock	PRDComponentsWorldStock	ZPRDSTOCK2CH	(Y / Z) MMSTKSND		
DPILJ	WorldStock	PRDComponentsWorldStock	ZPRDSTOCKBKDWN	(Y / Z) MMSTKBRK		
HPILJ	SendOrderCostToBW	PRDSendOrderCost	ZPRDSENDORDCOST	(Y / Z) MARGCTRL	Y_MARG_DPP	ZHWM_MARGCTRL
KPILJ	ProductionLeadTime	PRDProdLeadTime	ZPRDPRODLEADTIME	(Y / Z) MLEADTIM	YLOAD_KPILJ	ZHWM_ZMLEADTIM
CPILJ	OrderRemainderAndForecast	PRDOrderRemainderAndForecast	ZPRDORDFORECAST	(Y / Z) ODSPREV (Y / Z) PREVPRD (Y / Z) OPNPRDWK	Y_PRE_LOAD_CPILJ	ZHWM_CPILJ_PREVPRD_OPNPRD
CPILJ			ZPRDORDREMINDER	(Y / Z) ODSOPN (Y / Z) MMOPNPRD (Y / Z) OPNPRDWK		
APSTJ	TreeCatalogBatch	PRDTreeCatalog	ZPRDLINKFAMMOD ZPRDFAMARBO ZPRDINTERNATMOD	(Y / Z) MATERDPP (Y / Z) SUBDPDPP (Y / Z) DEPTDPP (Y / Z) FAMILDPP (Y / Z) SECTORDP		

In case of Error during loading

In case of duplicated lines :

Status will be 9.

If this error occurs, please contact Prod.com support team to check if it is normal or not.

Case 1 :

Batch has been relaunched by mistake. Delete the status 9 lines from the table.

Case 2 :

A relaunch has been done to correct older data.

Delete the first Unique ID loaded in BW infoproviders with a selective deletion.

If several batchs have been sent and are in Status “9”, check which unique ID is the last one loaded and delete all the others Unique ID in Status “9”.

Create or use an infopackage with a data loading restriction on status 9

Restart the Process chain.

In case of Status “4” :

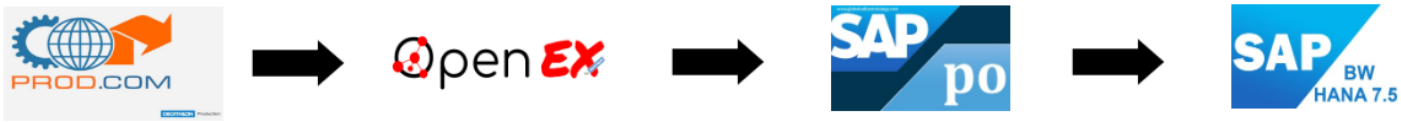
Error on infopackage step:
Do not restart the step.
Check is required on the BW table of the concerned proxy
Complete restart of the process chain is needed.

Error on DTP from Datasource to DSO:
Do not restart the step
Check the error and correct in the BW table if needed
Complete restart of the process chain is needed.

Error on activation:
Do not restart the step
Check the error and correct in the BW table if needed
Complete restart of the process chain is needed.

Error on DTP from DSO to Infoprovider:
Check the error.
Restart the step.

Responsibility Matrix:



Qui / Erreur	Erreur Prod.com	Erreur communication prod.com / OpenEx	Erreur OpenEx	Erreur communication OpenEx / SAP PO	Erreur SAP PO	Erreur communication SAP PO / SAP BW	Erreur SAP BW
Prod.com	S / R	S / R	S / R	S / R	-	-	I
OpenEx	-	C	C	C	-	-	-
SAP PO	-	-	-	C	C	C	-
BW	I	I	I	C	S / R	S / R	S / R

S	Supervision	Logiciel de supervision?
R	Relance	
C	Consulté	
I	Informé	