

MECANO - Work Order

- [Objective of the application](#)
- [Main infoprovider](#)
- [Dataflow overview](#)
- [Functional and Technical rules on Workbench + Reporting](#)
 - [SNAPSHOT WEEK](#)
 - [C_EXEFLG](#)
 - [C_EXPFLG](#)
 - [Date status \(eg. C_TECODT/C_CNFDOT/C_CLSDDT/C_GS45DT\)](#)
 - [Definition of the key figures](#)

Objective of the application

The detail of work order, which is loaded from PM order attribute master data. It also use to look up during update CALL data flow.

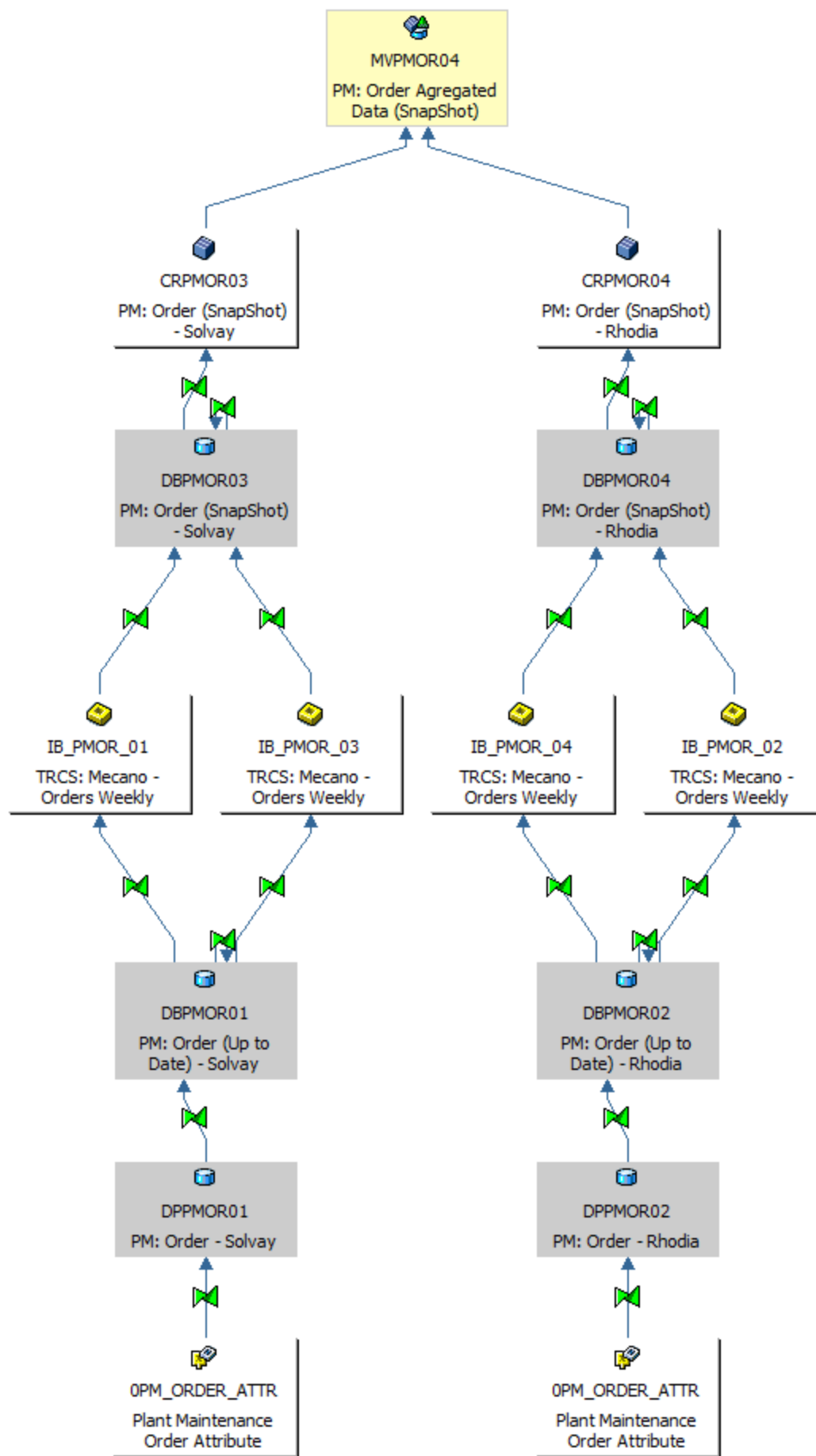
Main infoprovider

MVPMOR01 = PM: Order Detailed Data (Up to Date)

MVPMOR02 = PM: Order Detailed Data (SnapShot) - DSO level

MVPMOR04 = PM: Order Agregated Data (SnapShot)

Dataflow overview





Infosource from DBPMOR01/2 to DBPMOR03/4

In progress = IB_PMOR_01/2 C_CLOSDAY = Blank

Snapshot 2.11 = IB_PMOR_03/4 C_EXEFLG = X, C_EXEFLGD = last week

Loading on 11.01.2021 at 01:00 will select C_EXEFLG=X and C_EXEFLGD=20210104-20210110; Snapshot week 01.2021

Functional and Technical rules on Workbench + Reporting

SNAPSHOT WEEK

FM ZDETERMINE_NEXT_WEEK (depends on the selection)

1. Get from max DBPMOR03 by SELECT MAX = last week loading
2. Get current snapshot date from max snapshot week of point 1 and transform to date with FM FUNCTION 'WEEK_GET_FIRST_DAY' + 7
3. Then get snapshot week by FUNCTION 'DATE_GET_WEEK'

Note: In case, can't find max snapshot from DBPMOR03/04, it will get from global filter

```
ELECT SINGLE /BIC/C_LOW
  INTO w_date_text
  FROM /BIC/PC_GLBFLT
  WHERE /BIC/C_STREAM = 'P2P_PM' AND
        /BIC/C_RULE = 'INIT_WEEK' AND
        /BIC/C_GLBFLT = 1 AND
        OBJVERS = 'A' AND
        /BIC/C_ACTIVE = 'Y'.
```

If can't find in global filter with this selection, it will get the loading date and converse to snapshot week by FUNCTION 'DATE_GET_WEEK'

C_EXEFLG

Determine at DBPMOR0x -> ODSO DBPMOR0x (self loop) end routine

call method oref_mecano_utils->calculate_kpi_2_11

C_EXPFLG

Determine at DBPMOR0x -> ODSO DBPMOR0x (self loop) end routine

call method oref_mecano_utils->calculate_kpi_2_11

If there is **NO TECO date**, it compares the required end date and **execution date** (get date by priority GS45, GS50, GS55, 6CMP, confirm date and reference date).

Execution date <= required end date : C_EXPFLG = X (order is executed before the required date)

Data Browser: Table /BIC/ADBPMOR0100 Select Entries 1

by priority

	/BIC/C_PMORDR	/BIC/C_EXEF...	/BIC/C_EXPF...	/BIC/C_TECO...	/BIC/C_EXEFLGD	/BIC/C_GS45DT	/BIC/C_GS50DT	/BIC/C_GS55DT	/BIC/C_6CMPDT	/BIC/C_CNFDT	/BIC/C_REFDT	/BIC/C_REQEND
LOGSYS	/BIC/C_PMORDR	/BIC/C_EXEF...	/BIC/C_EXPF...	/BIC/C_TECO...	/BIC/C_EXEFLGD	/BIC/C_GS45DT	/BIC/C_GS50DT	/BIC/C_GS55DT	/BIC/C_6CMPDT	/BIC/C_CNFDT	/BIC/C_REFDT	/BIC/C_REQEND
PF1_020	310201082380	X			10.01.2021		10.01.2021			10.01.2021	06.01.2021	07.01.2021

If there is a TECO date,

Reference date <= required end date : C_EXPFLG = X (order is executed before the required date)

Date status (eg. C_TECODT/C_CNFDT/C_CLSDDT/C_GS45DT)

From datasource OPM_ORDER_ATTR-ZTECODATE

User exit ZXRSAU02_OPM_ORDER_ATTR

It gets from table **JCDS** (Change Documents for System/User Statuses (Table JEST))

```
loop at itb_jcds_system_status_dates[]
  assigning <fs_system_status_dates>
  where OBJNR = <fs_data>-OBJNR.
  case <fs_system_status_dates>-STAT.
    when 'I0001' or 'I0002' or 'I0010'.
      clear: <fs_data>-ZCNFDATE,
              <fs_data>-ZTECODATE,
              <fs_data>-ZCLSDDATE.
    when 'I0009'.
      if <fs_data>-ZCNFDATE is initial.
        <fs_data>-ZCNFDATE = <fs_system_status_dates>-update.
      endif.
      clear: <fs_data>-ZTECODATE,
              <fs_data>-ZCLSDDATE.
    when 'I0045'.
      if <fs_data>-ZTECODATE is initial.
        <fs_data>-ZTECODATE = <fs_system_status_dates>-update.
      endif.
      clear: <fs_data>-ZCLSDDATE.
    when 'I0046'.
      if <fs_data>-ZCLSDDATE is initial.
        <fs_data>-ZCLSDDATE = <fs_system_status_dates>-update.
      endif.
  endcase.
endloop.
```

```
loop at itb_jcds_user_status_dates[]
  assigning <fs_user_status_dates>
  where OBJNR = <fs_data>-OBJNR
  and STSMA = <fs_data>-ZZSTATPRF.
  case <fs_user_status_dates>-TXT04.
    when 'GS00' or 'GS10' or 'GS20' or 'GS25' or 'GS30' or 'GS31' or 'GS35' or 'GS40'.
      clear: <fs_data>-ZGS45DATE,
              <fs_data>-ZGS50DATE,
              <fs_data>-ZGS55DATE.
    when 'GS45'.
      if <fs_data>-ZGS45DATE is initial.
        <fs_data>-ZGS45DATE = <fs_user_status_dates>-update.
      endif.
      clear: <fs_data>-ZGS50DATE,
              <fs_data>-ZGS55DATE.
    when 'GS50'.
      if <fs_data>-ZGS50DATE is initial.
        <fs_data>-ZGS50DATE = <fs_user_status_dates>-update.
      endif.
      clear: <fs_data>-ZGS55DATE.
    when 'GS55'.
      if <fs_data>-ZGS55DATE is initial.
        <fs_data>-ZGS55DATE = <fs_user_status_dates>-update.
      endif.
    when '6CMP'.
      if <fs_data>-Z6CMPDATE is initial.
        <fs_data>-Z6CMPDATE = <fs_user_status_dates>-update.
      endif.
  endcase.
endloop.
```

Definition of the key figures

Key Figure	Key Figure Description		Exe Flag	Expected Exe flag	Req End date	Maintenance Plan	INPR Flag snapshot
		KF	C_EXEFLG	C_EXPFLG	C_REQEND	C_MPLAN	C_CLOSFLG
BW_RKF_MVPMOR02_0001	Number of orders in progress	K_COUNTER					X

BW_RKF_MVPMOR02_0002	Number of orders executed	K_COUNTER	X			#	
BW_RKF_MVPMOR02_0003	Number of orders executed before required date	K_COUNTER	X	X	<> #	#	
BW_RKF_MVPMOR02_0004	Number of orders executed (without a required end date)	K_COUNTER	X		#	#	
BW_RKF_MVPMOR02_0005	Number of orders executed (with a required end date)	K_COUNTER	X		<> #	#	