



# Functional Release Notes

Version 5.11.20

---

## *Release Information*



Document: Force-5-11-20-Functional-Release-Notes DRAFT 2.docx



Release date: 2021-10-13



Document version: 1



Author: Ali Egilmez

## Content

<b>1</b>	<b>General</b> .....	<b>3</b>
<b>2</b>	<b>Planning</b> .....	<b>4</b>
<b>3</b>	<b>Process</b> .....	<b>5</b>
<b>4</b>	<b>Productivity</b> .....	<b>6</b>
<b>5</b>	<b>Appendix</b> .....	<b>7</b>
5.1	Abbreviations.....	7
5.2	SQL Examples.....	7

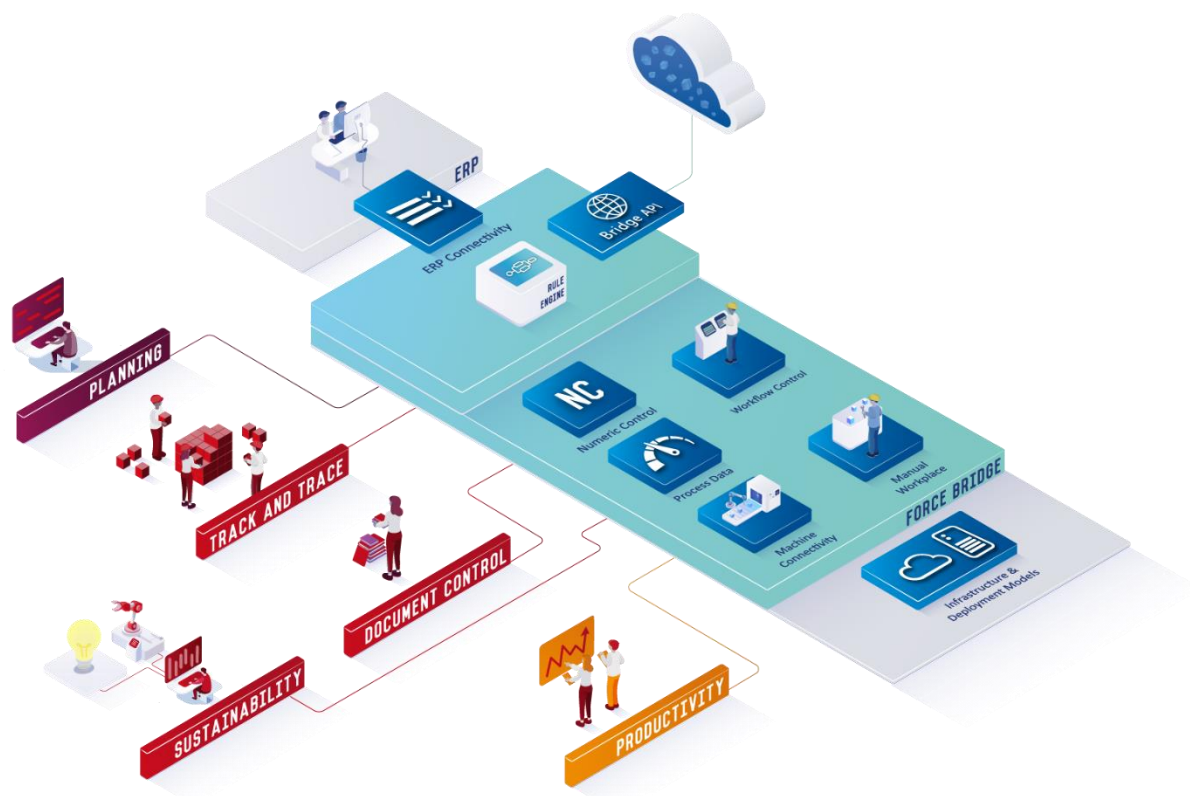
# 1 General

FORCAM provides companies with all the information they need to control and optimize their production. The modular IIoT solution FORCAM FORCE™ is as flexible as the needs of customers. With a variety of production apps, FORCAM helps companies to make their processes more transparent and improve their workflows. In this way, companies create the basis for optimization measures and sustainable success, thereby ensuring their competitiveness.

FORCAM is always endeavored to optimize the solution FORCAM FORCE™ further, to enrich it with functions, offering customers consistent growth and greatest possible use. For this purpose, several releases are published throughout the year.

This document lists functions that have been added or changed during the **5.11.20** release. It serves as an overview of the most important features to use FORCAM FORCE™ in the best possible way.

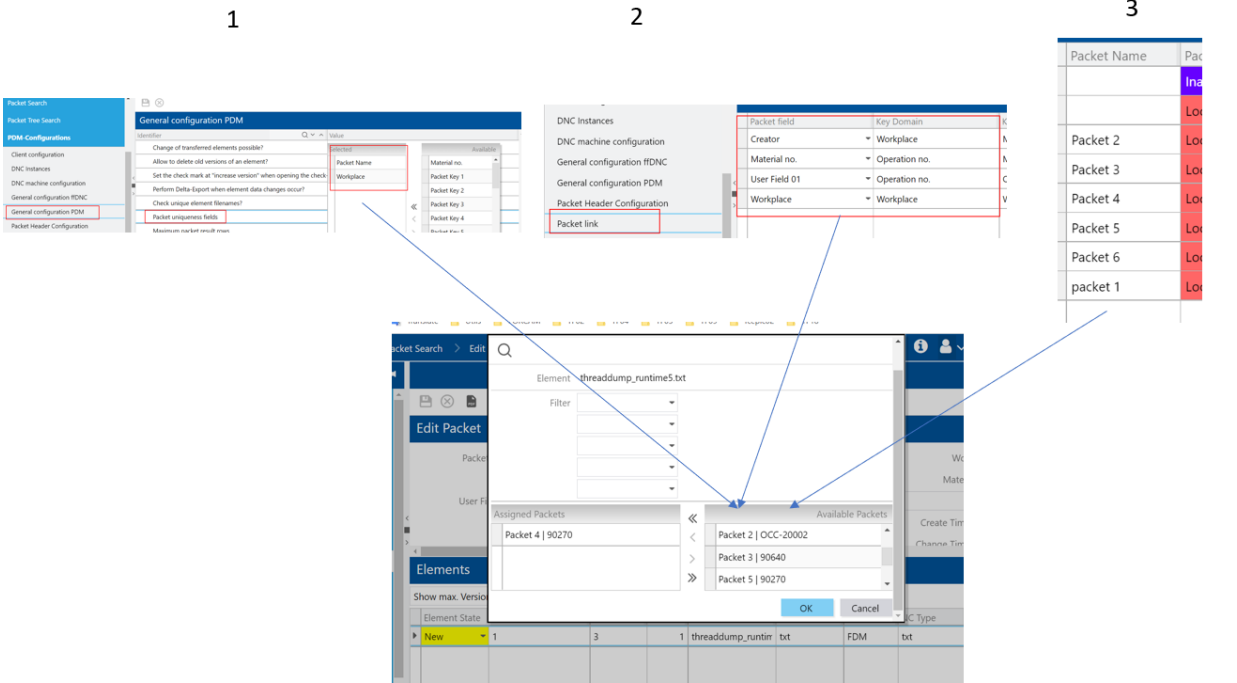
Detailed descriptions of the individual functions can be found in the respective product documentation.



## 2 Planning

Title	Description	Affected module	Affected area	Status										
<p><b>Time per unit with decimal places</b></p>	<p>The operation table can visualize “time per unit” for each operation. It shows how much time is needed for producing X units, e.g. 2,460 sec per 1 unit.            X was previously specified as an integer. It has now been changed to a decimal number with a maximum of two decimal places, e.g. 2,460 sec per 0,56 units.            The literals have been adapted to explain this functionality better:</p> <table border="1" data-bbox="443 708 875 952"> <thead> <tr> <th>Time per Unit</th> <th>Block no.</th> </tr> </thead> <tbody> <tr> <td>Q ≡</td> <td>Q ≡</td> </tr> <tr> <td>5 Min per 2.5</td> <td></td> </tr> <tr> <td>2,460 Sec per 0.12</td> <td></td> </tr> <tr> <td colspan="2">Time per one unit/stroke: 20,000 Sec</td> </tr> </tbody> </table>	Time per Unit	Block no.	Q ≡	Q ≡	5 Min per 2.5		2,460 Sec per 0.12		Time per one unit/stroke: 20,000 Sec		Workbench	Digital Planning Board	Changed
Time per Unit	Block no.													
Q ≡	Q ≡													
5 Min per 2.5														
2,460 Sec per 0.12														
Time per one unit/stroke: 20,000 Sec														

### 3 Process

Title	Description	Affected module	Affected area	Status
<p>Distinguish the packet in search filter when assigning elements to packet</p>	<p>Scenario: During packet edit when assigning an element to a packet. To distinguish packets from each other, the system previously used the values of the fields that the user configured as “packet linked”. This was changed to the following: The filter now displays values based on whether packet uniqueness is defined. If so, then packet uniqueness combination is displayed in the search filter. If packet link is defined, then packet link combination is displayed in the search filter. Otherwise, packet name is displayed in the search filter.</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <span>1</span> <span>2</span> <span>3</span> </div> 	Workbench	Document Control	Changed

## 4 Productivity

Title	Description	Affected module	Affected area	Status
<p><b>Reduced visualization loading times</b></p>	<p>The Workplace <b>Current status</b> table <b>FR_DS_WORKPLACE_CS</b> has been extended with new columns <b>CURRENT_SHIFT_ID</b> and <b>NONWORKING_SHIFT_TYPE</b>. These columns will be updated with the Business logic with shift information. This shift information can be used by the SQL reporting queries to avoid the complex joins to find the shift information, resulting in shorter loading times for calling up and using visualizations. The Business Logic (lib/ds/monitor/workplace-current-status.drl) has been updated to fill the shift information into the FR_DS_WORKPLACE_CS table along with existing current status data. Once the changes are done, the SQL must be adapted. For the SQL delivered by FORCAM, we recommend the SQL which you can find in the Appendix.</p>	Office	Reporting	Changed

## 5 Appendix

### 5.1 Abbreviations

Abbreviation/Term	Meaning
PDM	Production Data Management
sec	Seconds
SQL	Standardized query language for databases

### 5.2 SQL Examples

#### Recommended SQL for Target Shift Quantity:

```

SELECT WORKPLACE_ID,
ROUND(SUM(TARGET_QTY_IN_SHIFT), 0) as TARGET_QTY_IN_SHIFT
FROM
(
    SELECT WORKPLACE_ID,
        SECOND_PER_UNIT,
        SHIFT_ID,
        CASE WHEN ((SECOND_PER_UNIT IS NOT NULL) AND (SECOND_PER_UNIT > 0))
            THEN
                CAST(ROUND((CAST(SUM(DURATION) AS NUMERIC) * 1000/SECOND_PER_UNIT), 0) AS INT )
            ELSE 0
        END AS TARGET_QTY_IN_SHIFT,
        --CAST(ROUND((CAST(SUM(DURATION) AS NUMERIC) * 1000/SECOND_PER_UNIT), 0) AS INT ) AS
TARGET_QTY_IN_SHIFT,
        SUM(DURATION) as OEE_TIME_PER_SHIFT
    FROM
    (
        --All OEE relevant time line elements in the current shift
        SELECT WPLTL.WORKPLACE_ID as WORKPLACE_ID,
            LEADING_OPERATION_ID AS OPERATION_ID,
            OP.TIME_PER_UNIT as SECOND_PER_UNIT,
            (CASE WHEN
                ((SELECT COUNT(*) FROM V_MD_TIMEBASE_ENTRY WHERE TIMEBASE_ID IN
                    (SELECT ID FROM FR_MD_TIMEBASE WHERE MNEMONIC='OEE') AND PHASE_ID =
WPLTL.WORKPLACE_PHASE_ID AND OPERATING_STATUS_ID =
                    (CASE WPLTL.MALFUNCTION_REASON_VECTOR_ID WHEN 0 THEN WPLTL.WORKPLACE_STATUS_ID
ELSE
                        (SELECT REASON_ID_1 FROM FR_MD_MALFUNCTION_REASON_VEC WHERE
ID=WPLTL.MALFUNCTION_REASON_VECTOR_ID) END ) = 0)
                THEN 0 ELSE (DATEDIFF (second, WPLTL.START_TS, CASE WHEN WPLTL.END_TS = '1970-01-01
01:00:00.0000000' THEN CURRENT_TIMESTAMP ELSE WPLTL.END_TS END)) END) AS DURATION,
            WPLTL.START_TS,
            CASE WHEN WPLTL.END_TS = '1970-01-01 01:00:00.0000000' THEN CURRENT_TIMESTAMP
            ELSE WPLTL.END_TS END AS END_TS,
            SHIFT_ID
        FROM FR_DS_WORKPLACE_BASE_TL WPLTL
        JOIN FR_DS_WORKPLACE_CS WCS on WPLTL.WORKPLACE_ID = WCS.WORKPLACE_ID and WPLTL.SHIFT_ID =
WCS.CURRENT_SHIFT_ID and WCS.NONWORKING_SHIFT_TYPE = 0
        JOIN FR_MD_OPERATION OP ON OP.ID = WPLTL.LEADING_OPERATION_ID
        WHERE @WORKPLACE(WPLTL.WORKPLACE_ID)
    )DATA
    GROUP BY WORKPLACE_ID,

```

## Appendix

```

        SECOND_PER_UNIT,
        SHIFT_ID
    )DATA_SUM
    GROUP BY WORKPLACE_ID

```

### Recommended SQL for Current OEE Shift

```

select
    cur_shift_oee.WORKPLACE_ID as SHIFT_WORKPLACE_ID,
    cur_shift_oee.AVAILABILITY as SHIFT_AVAILABILITY,
    cur_shift_oee.PERFORMANCE as SHIFT_PERFORMANCE,
    cur_shift_oee.QUALITY as SHIFT_QUALITY,
    cur_shift_oee.OEE as SHIFT_OEE
from (
    select
        cur_shift_prod_oee.WORKPLACE_ID,
        cur_shift_prod_oee.SHIFT_ID,
        sum(cur_shift_prod_oee.TB_OEE_DURATION) as TB_OEE_DURATION,
        sum(cur_shift_prod_oee.TB_PROD_DURATION) as TB_PROD_DURATION,
        ((case when
            sum(cur_shift_prod_oee.TB_OEE_DURATION) > 0 then (sum(cur_shift_prod_oee.TB_PROD_DURATION)
* 1.0 / sum(cur_shift_prod_oee.TB_OEE_DURATION))
            else 0
        end) * 100) as AVAILABILITY,
        ((case when
            sum(cur_shift_prod_oee.TB_PROD_DURATION) > 0 then
(sum(isnull(cur_shift_qty_sum.CALC_DURATION, 0)) / sum(cur_shift_prod_oee.TB_PROD_DURATION))
            else 0
        end) * 100) as PERFORMANCE,
        ((case when
            sum(cur_shift_qty_sum.QTY) > 0 then (sum(cur_shift_qty_sum.YIELD_QTY) /
sum(cur_shift_qty_sum.QTY))
            else 0
        end) * 100) as QUALITY,
        ((case when
            (sum(cur_shift_prod_oee.TB_OEE_DURATION) > 0 and sum(cur_shift_qty_sum.QTY) > 0) then
((sum(cur_shift_qty_sum.YIELD_QTY) * sum(cur_shift_qty_sum.CALC_DURATION)) /
(sum(cur_shift_qty_sum.QTY) * sum(cur_shift_prod_oee.TB_OEE_DURATION)))
            else 0
        end) * 100) as OEE
    from (
        select
            cur_shift_tb_sum.WORKPLACE_ID,
            cur_shift_tb_sum.SHIFT_ID,
            sum(cur_shift_tb_sum.TB_OEE_DURATION) as TB_OEE_DURATION,
            sum(cur_shift_tb_sum.TB_PROD_DURATION) as TB_PROD_DURATION
        from (
            select
                cur_shift_tb.WORKPLACE_ID,
                cur_shift_tb.SHIFT_ID,
                CASE WHEN OEE_TIMEBASE_INDICATOR = 0 THEN 0 ELSE cur_shift_tb.SUM_DURATION end as
TB_OEE_DURATION,
                CASE WHEN PROD_TIMEBASE_INDICATOR = 0 THEN 0 ELSE cur_shift_tb.SUM_DURATION end as
TB_PROD_DURATION
            from (
                select
                    WBTL.WORKPLACE_ID,
                    WBTL.SHIFT_ID,
                    sum(case when DURATION = 0 then cast(datediff(SECOND, START_TS, CURRENT_TIMESTAMP)
as bigint) * 1000 else DURATION end) as SUM_DURATION,
                    WBTL.WORKPLACE_PHASE_ID,
                    WBTL.WORKPLACE_STATUS_ID,
                    WBTL.MALFUNCTION_REASON_VECTOR_ID,
                    OEE_TIMEBASE_INDICATOR,
                    PROD_TIMEBASE_INDICATOR
                from FR_DS_WORKPLACE_BASE_TL WBTL
                join FR_DS_WORKPLACE_CS WCS on WBTL.WORKPLACE_ID = WCS.WORKPLACE_ID and WBTL.SHIFT_ID =
WCS.CURRENT_SHIFT_ID
                where
                    @WORKPLACE(WBTL.WORKPLACE_ID)
            )
        )
    )

```



## Appendix

```

        group by WBTL.WORKPLACE_ID, WBTL.SHIFT_ID, WBTL.WORKPLACE_PHASE_ID,
WBTL.WORKPLACE_STATUS_ID, WBTL.MALFUNCTION_REASON_VECTOR_ID, OEE_TIMEBASE_INDICATOR,
PROD_TIMEBASE_INDICATOR
    ) cur_shift_tb
  ) cur_shift_tb_sum
  group by cur_shift_tb_sum.WORKPLACE_ID, cur_shift_tb_sum.SHIFT_ID
) cur_shift_prod_oe
left join (
  select
    cur_shift_qty.WORKPLACE_ID,
    cur_shift_qty.SHIFT_ID,
    isnull(sum(cur_shift_qty.AMOUNT * op.TIME_PER_UNIT), 0) as CALC_DURATION,
    isnull(sum(cur_shift_qty.AMOUNT), 0) as QTY,
    isnull(sum(cur_shift_qty.YIELD_QTY), 0) as YIELD_QTY
  from (
    select
      qtyle.WORKPLACE_ID,
      qtyle.SHIFT_ID,
      qtyle.OPERATION_ID,
      sum(qtyle.AMOUNT) as AMOUNT,
      sum(case when qtytp.CODE = 1 then qtyle.AMOUNT else 0 end) as YIELD_QTY
    from FR_DS_OPERATION_QUANTITY_LE qtyle
    join FR_MD_OPERATION_QUANTITY_TYPE qtytp on qtytp.ID = qtyle.QUANTITY_TYPE_ID
    join FR_MD_OPERATION op on op.ID = qtyle.OPERATION_ID
    join FR_DS_WORKPLACE_CS WCS on qtyle.WORKPLACE_ID = WCS.WORKPLACE_ID and qtyle.SHIFT_ID =
WCS.CURRENT_SHIFT_ID
    where
      @WORKPLACE(qtyle.WORKPLACE_ID)
    group by qtyle.WORKPLACE_ID, SHIFT_ID, OPERATION_ID
  ) cur_shift_qty
  join FR_MD_OPERATION op on op.ID = cur_shift_qty.OPERATION_ID
  group by cur_shift_qty.WORKPLACE_ID, cur_shift_qty.SHIFT_ID
) cur_shift_qty_sum on
  cur_shift_qty_sum.WORKPLACE_ID = cur_shift_prod_oe.WORKPLACE_ID
  and cur_shift_qty_sum.SHIFT_ID = cur_shift_prod_oe.SHIFT_ID
  group by cur_shift_prod_oe.WORKPLACE_ID, cur_shift_prod_oe.SHIFT_ID
) cur_shift_oe

```