**INT00020R** MULTI-LEVEL-TARGETED-SPECIFICATIONS

[product&Process Developement] + [ Specifications] + [Multi-Level-Targeted-Specifications]

Part-no: GM\_ST706003

Afbeelding met tekst, schermopname, software, Webpagina

Automatisch gegenereerde beschrijving

[run report]

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

FILTER-SQL-STATEMENTS

**interspec/INT00040 (=html-bestand)**

**part-no**

JOIN INNER

SPECIFICATION\_HEADER.SPECIFICATION\_HEADER.STATUS IN specification\_header TAG SPECH TO UNIQUE

STATUS.STATUS.STATUS IN status TAG STATUS AS J0

END

TABLE FILE SPECIFICATION\_HEADER

BY  SPECIFICATION\_HEADER.SPECIFICATION\_HEADER.PART\_NO

BY  SPECIFICATION\_HEADER.SPECIFICATION\_HEADER.REVISION

WHERE SPECIFICATION\_HEADER.SPECIFICATION\_HEADER.PART\_NO LIKE '&amp;PART\_PATTERN|%';

WHERE STATUS.STATUS.STATUS\_TYPE EQ 'CURRENT';

END

**TEMPLATE**

JOIN

KW\_VALUE IN SPECIFICATION\_KW TAG SK TO UNIQUE

FRAME\_ID IN SPECIFICATION\_HEADER TAG H AS J0

END

JOIN

FILE SPECIFICATION\_KW AT SK.KW\_ID TO MULTIPLE

FILE ITKW AT KW\_ID TAG K AS J1

WHERE K.KW\_ID EQ SK.KW\_ID;

WHERE K.DESCRIPTION EQ 'Frame for Athena perspective';

END

JOIN

SK.PART\_NO IN SPECIFICATION\_KW TO UNIQUE

PART\_NO IN PART TAG P AS J2

END

JOIN

SK.PART\_NO IN SPECIFICATION\_KW TO UNIQUE

PART\_NO IN SPECIFICATION\_HEADER TAG H2 AS J3

END

JOIN

H2.STATUS IN SPECIFICATION\_KW TO UNIQUE

STATUS IN STATUS TAG S AS J4

END

TABLE FILE SPECIFICATION\_KW

SUM P.DESCRIPTION

BY H2.PART\_NO AS TEMPLATE

WHERE H.PART\_NO EQ 'NONE'

  OR &amp;PART\_NO

;

FOCEXEC=UNI00020R20\_specHeader(

PART\_NO = PART\_NO

REVISION = REVISION

),

ALT='Revision History',

FOCEXEC=INT00050R\_specHistory(

PART\_NO = PART\_NO

REVISION = REVISION

JOIN

STATUS IN SPECIFICATION\_HEADER TAG header TO

STATUS IN STATUS TAG status AS J0

END

JOIN

header.LAST\_MODIFIED\_BY IN SPECIFICATION\_HEADER TO

USER\_ID IN ITUS TAG user AS J1

END

JOIN LEFT\_OUTER

FILE SPECIFICATION\_HEADER AT header.PART\_NO TO MULTIPLE

FILE STATUS\_HISTORY AT PART\_NO TAG statHist AS J2

WHERE statHist.PART\_NO EQ header.PART\_NO;

WHERE statHist.REVISION EQ header.REVISION;

WHERE statHist.REASON\_ID IS NOT MISSING;

END

JOIN LEFT\_OUTER

statHist.REASON\_ID IN SPECIFICATION\_HEADER TO

ID IN REASON TAG reason AS J3

END

-\*JOIN LEFT\_OUTER

-\* statHist.STATUS IN SPECIFICATION\_HEADER TO UNIQUE

-\* STATUS IN STATUS TAG histStatus AS J4

-\*END

DEFINE FILE SPECIFICATION\_HEADER

ISSUED\_DATE/A20 = HCNVRT(header.ISSUED\_DATE, '(HYYMDS)', 20, 'A20');

END

TABLE FILE SPECIFICATION\_HEADER

PRINT

-\* header.LAST\_MODIFIED\_ON AS 'Last Modified,On'

header.ISSUED\_DATE AS 'Issued,On'

COMPUTE FULLNAME/A21 = user.FORENAME || ' ' | user.LAST\_NAME;

AS 'Last Modified,By'

-\* header.STATUS\_CHANGE\_DATE AS 'Status,Change Date'

header.OBSOLESCENCE\_DATE AS 'Obsoleted,On'

reason.TEXT AS 'Reason for Issue'

-\* histStatus.SORT\_DESC AS 'Status at Reason'

ISSUED\_DATE NOPRINT

COMPUTE FMXNP/I3 = &FMXNP; NOPRINT

BY header.PART\_NO NOPRINT

BY HIGHEST header.REVISION AS 'Rev.'

BY header.DESCRIPTION AS 'Spec.,Description'

BY status.SORT\_DESC AS 'Status'

WHERE header.PART\_NO EQ '&PART\_NO';

WHERE header.REVISION LE &REVISION;

ON TABLE SUBHEAD

"Specification History"

--INCLUDE INT00040R3\_BoM\_Properties

--TABLE FILE INT00040H\_TPL\_FULL

--INCLUDE INT00040R1\_Top\_Properties

**INT00040R\_templatedSingleSpec**

**-INCLUDE INT00040P\_BoM**

**-INCLUDE INT00040P\_Properties**

**--**

**INCLUDE INT00040R2\_TopBoMProperties**

**INCLUDE INT00040R3\_BoM\_Properties**

**INT00040P\_BOM**

**WITH**

**-INCLUDE INT00011P2\_BoM\_explosion\_SQL**

**-INCLUDE INT00013P\_BoM\_template**

**-INCLUDE INT00013P\_BoM\_filter**

**SELECT f.\***

**, part.description AS part\_description**

**, weight.weight, weight.uom AS weight\_uom**

**FROM filtered f**

**LEFT JOIN part ON (part.part\_no = f.part\_no)**

**LEFT JOIN avspecification\_weight weight ON (weight.part\_no = f.part\_no)**

**ORDER BY**

**f.hierarchy, f.functiongrp, f.item\_number**

**;**

**From:** Kitti Tóth <[Kitti.Toth@apollotyres.com](mailto:Kitti.Toth@apollotyres.com)>   
**Sent:** Friday, August 18, 2023 9:28  
**To:** Patrick Goossens <[patrick.goossens@apollotyres.com](mailto:patrick.goossens@apollotyres.com)>; Ton Spreeuwenberg <[ton.spreeuwenberg@apollotyres.com](mailto:ton.spreeuwenberg@apollotyres.com)>; Nico Gevers <[nico.gevers@apollotyres.com](mailto:nico.gevers@apollotyres.com)>  
**Cc:** Péter Lőrincz <[Peter.Lorincz@apollotyres.com](mailto:Peter.Lorincz@apollotyres.com)>; Mahavir Annadate <[Mahavir.Annadate@Apollotyres.com](mailto:Mahavir.Annadate@Apollotyres.com)>  
**Subject:** FW: Call ID: I2308-3393 has been registered. Subject:

Dear Sirs!

We need your help to solve one problem, which is Athena/interspec related.

We have 2 problems, which are related to each other.  
  
**1. New OE size was approved and it is required the flipper application. In Athena multi level targeted specification can not show that the GT BOM is including the flipper. Only you can see it in the targeted specifications. (attached the 2 pictures of the same GT BOM, which contains flipper)**  
2. The GT weight is not correct, non of the targeted or multi level targeted specifications can show the real value.   
The problem is only 1 meter flipper is mentioned(and calculated) in the targeted specifications, and because of that the GT weight not will be correct. (the flipper lenght at least the same as ply lenght+10mm)  
  
We need to solve this issue as soon as possible, because without this we can not produce the new size in the regular production.

Thank you!

Regards,

Kitti

**Best Regards,**

**Kitti Kiss-Tóth**

Tyre Engineer

Technology Department | PCR

****

Email : [kitti.toth@apollotyres.com](mailto:kitti.toth@apollotyres.com)

Tel: +36 30 743 4259  
Apollo Tyres (Hungary) Kft., H-3212 Gyöngyöshalász, Apollo Road 106.

Correspondence address: H-3201 Gyöngyös, P.O. box 133.

Afbeelding met tekst, Lettertype, Graphics, logo

Automatisch gegenereerde beschrijving

WEL: XGG\_BF66A17J1

NIET: XGG\_BF66A17

Afbeelding met tekst, schermopname, software, Computerpictogram

Automatisch gegenereerde beschrijving

SPEC-HEADER:

XGG\_BF66A17J1 1 5 Greentyre BMW F66A17 15-02-2023 00:00:00 15-02-2023 13:19:06

XGG\_BF66A17J1 2 125 Greentyre BMW F66A17 15-02-2023 00:00:00 15-02-2023 13:59:43

SELECT \* FROM BOM\_HEADER WHERE PART\_NO='XGG\_BF66A17J1';

XGG\_BF66A17J1 1 GYO 1 1 N 1 15-02-2023 00:00:00 1

XGG\_BF66A17J1 2 GYO 1 1 N 1 15-02-2023 00:00:00 1

**🡺conclusie: alternative=1 = PREFFERRED**

SELECT \* FROM BOM\_ITEM WHERE PART\_NO=’XGG\_BF66A17J1’ AND REVISION=2;

XGS\_WBTV120TN 1 GYO 1 1 1 125 28-10-2021 09:53:44 A\_Sidewall v1 1.312 m 903394

XGL\_LAQ1360F360N 1 GYO 1 1 1 125 17-09-2021 09:52:16 A\_Innerliner v1 1.312 m 903345

XGR\_WR01450085 3 GYO 1 1 1 125 26-10-2022 10:53:22 A\_Ply v1 1 m 688

XGR\_WR01450085 3 GYO 1 1 1 125 26-10-2022 10:53:22 A\_Ply v1 1 m 689

XGR\_PK08490L80 1 GYO 1 1 1 125 12-10-2022 14:35:42 A\_Ply v1 1.323 m 903320

XGB\_AH138700124543 1 GYO 1 1 1 125 10-05-2022 14:41:25 A\_Bead v1 2 pcs 903323

XGR\_GSA1327195 1 GYO 1 1 1 125 15-08-2022 16:31:27 A\_Belt v1 1.838 m 903316

XGR\_NSA1327185 3 GYO 1 1 1 125 26-11-2021 13:34:15 A\_Belt v1 1.845 m 903317

GR\_NB01000012 7 GYO 1 1 1 128 29-06-2021 09:53:08 A\_Capply 31.44 m 903372

XGT\_PSTBF66A17D2 4 GYO 1 1 1 125 12-10-2022 14:54:35 A\_tread v1 1.857 m 903311

Select \* from specification\_header where part\_no in (’ XGR\_WR01450085’,’ XGR\_WR01450085’ );

XGR\_WR01450085 1 5 RL02-45-85 Rayon Flipper 02-12-2021 00:00:00 02-12-2021 09:10:03 26-10-2022 10:43:59 26-10-2022 10:43:59

XGR\_WR01450085 2 5 RL02-45-85 Rayon Flipper 26-10-2022 00:00:00 26-10-2022 10:43:59 26-10-2022 10:53:22 26-10-2022 10:53:22

XGR\_WR01450085 3 125 RL01-45-85 Rayon Flipper 26-10-2022 00:00:00 26-10-2022 10:53:22 26-10-2022 10:53:22

Select \* from part\_plant where part\_no in (’ XGR\_WR01450085’ ,’ XGR\_PK08490L80’ )

XGR\_PK08490L80 GYO

XGR\_WR01450085 GYO

* Beide zelfde plant = ok

**Your request did not return any output to display.**

**Possible causes:  
- No data rows matched the specified selection criteria.  
- Output was directed to a destination such as a file or printer.  
- An error occurred during the parsing or running of the request.**

**(FOC227) THE FOCEXEC PROCEDURE CANNOT BE FOUND: H&BUNDLE**

**(FOC227) THE FOCEXEC PROCEDURE CANNOT BE FOUND: INT00040H\_TPLBOMPARTS**

**0 NUMBER OF RECORDS IN TABLE= 0 LINES= 0**

**0 NUMBER OF RECORDS IN TABLE= 1 LINES= 1**

**0 NUMBER OF RECORDS IN TABLE= 1 LINES= 1**

**0 NUMBER OF RECORDS IN TABLE= 1 LINES= 1**

**SQL SQLORA PREPARE SQLBOM FOR**

**WITH**

**cart AS (**

**SELECT 'Array' AS origin, 'EF\_540/65R34TVZM45' AS part\_no  
, to\_date('2023/10/09 10:36:32', 'YYYY/MM/DD HH24:MI:SS') AS reference\_date  
, NULL AS plant FROM dual**

**),**

**partList AS (**

**SELECT**

**H.part\_no, H.revision**

**, H.issued\_date, H.planned\_effective\_date, H.obsolescence\_date**

**, H.class3\_id**

**, B.plant, B.preferred, B.alternative**

**, S.status, S.sort\_desc AS status\_code, S.status\_type**

**, L.reference\_date**

**FROM cart L**

**JOIN specification\_header H ON (H.part\_no = L.part\_no)**

**JOIN status S ON (S.status = H.status)**

**JOIN class3 C ON (C.class = H.class3\_id)**

**LEFT JOIN bom\_header B ON (**

**B.part\_no = H.part\_no**

**AND B.revision = H.revision**

**)**

**WHERE S.status\_type IN ('HISTORIC', 'CURRENT')**

**AND C.sort\_desc = 'FOC\_NONE'**

**AND B.preferred = 1**

**AND B.alternative = FOC\_NONE**

**AND (L.plant IS NULL OR B.plant = L.plant)**

**AND NOT EXISTS ( SELECT 1**

**FROM specification\_header**

**JOIN status USING (status)**

**WHERE part\_no = H.part\_no**

**AND revision > H.revision**

**AND status\_type IN ('HISTORIC', 'CURRENT')**

**)**

**AND (**

**'DEFAULT' <> 'REFDATE'**

**OR (**

**'DEFAULT' = 'REFDATE'**

**AND L.reference\_date >= coalesce(H.issued\_date, H.planned\_effective\_date)**

**AND (H.obsolescence\_date IS NULL OR L.reference\_date < H.obsolescence\_date)**

**)**

**)**

**GROUP BY**

**H.part\_no, H.revision**

**, H.issued\_date, H.planned\_effective\_date, H.obsolescence\_date**

**, H.class3\_id**

**, B.plant, B.preferred, B.alternative**

**, S.status, S.sort\_desc, S.status\_type**

**, L.reference\_date**

**)**

**, specList AS (**

**SELECT**

**h2.part\_no, h2.revision**

**, h2.issued\_date, h2.obsolescence\_date, h2.planned\_effective\_date**

**, h2.status, s2.sort\_desc AS status\_code, s2.status\_type**

**, h2.class3\_id**

**, CASE WHEN 'DEFAULT' = 'REFDATE' THEN 1 ELSE 0 END AS f\_checkRefDate**

**FROM specification\_header h2**

**JOIN status s2 ON (**

**s2.status = h2.status**

**AND status\_type IN ('HISTORIC', 'CURRENT')**

**)**

**WHERE ('DEFAULT' = 'HIGHEST' OR h2.issued\_date IS NOT NULL)**

**)**

**, componentList AS (**

**SELECT**

**h2.part\_no, h2.revision**

**, h2.issued\_date, h2.obsolescence\_date, h2.planned\_effective\_date**

**, h2.status, s2.sort\_desc AS status\_code, s2.status\_type**

**, h2.class3\_id**

**, CASE WHEN 'DEFAULT' = 'REFDATE' THEN 1 ELSE 0 END AS f\_checkRefDate**

**FROM specification\_header h2**

**JOIN status s2 ON (**

**s2.status = h2.status**

**AND status\_type IN ('HISTORIC', 'CURRENT')**

**)**

**WHERE h2.part\_no NOT LIKE 'X%'**

**AND ('DEFAULT' = 'HIGHEST' OR h2.issued\_date IS NOT NULL)**

**UNION ALL**

**SELECT**

**h2.part\_no, h2.revision**

**, h2.issued\_date, h2.obsolescence\_date, h2.planned\_effective\_date**

**, h2.status, s2.sort\_desc AS status\_code, s2.status\_type**

**, h2.class3\_id**

**, CASE WHEN 'DEFAULT' = 'REFDATE' THEN 1 ELSE 0 END AS f\_checkRefDate**

**FROM specification\_header h2**

**JOIN status s2 ON (**

**s2.status = h2.status**

**AND status\_type IN ('HISTORIC', 'CURRENT')**

**)**

**WHERE h2.part\_no LIKE 'X%'**

**AND ('DEFAULT' = 'HIGHEST' OR h2.issued\_date IS NOT NULL)**

**)**

**, root\_function AS (**

**SELECT sk.part\_no, sk.kw\_value**

**FROM specification\_kw sk**

**JOIN itkw ON (**

**itkw.kw\_id = sk.kw\_id**

**AND itkw.description = 'Function'**

**)**

**)**

**, bom\_function AS (**

**SELECT f.characteristic\_id, f.description**

**FROM characteristic f**

**JOIN characteristic\_association ca ON (**

**ca.characteristic = f.characteristic\_id**

**AND ca.intl = f.intl**

**)**

**JOIN association a ON (**

**a.association = ca.association**

**AND a.intl = ca.intl**

**)**

**WHERE a.description = 'Function'**

**)**

**, tree (**

**root\_part, root\_rev**

**, root\_status, root\_status\_type**

**, root\_preferred, root\_alternative**

**, root\_issued\_date**

**, root\_density**

**, parent\_part, parent\_rev**

**, parent\_status, parent\_status\_type**

**, parent\_seq**

**, parent\_function, parent\_uom**

**, part\_no, revision, plant, item\_number**

**, issued\_date, obsolete\_date**

**, status, status\_code, status\_type**

**, class3\_id**

**, function**

**, preferred, alternative, base\_quantity**

**, phr, quantity, density, uom**

**, num\_1, num\_2, num\_3, num\_4, num\_5**

**, char\_1, char\_2, char\_3, char\_4, char\_5**

**, date\_1, date\_2**

**, boolean\_1, boolean\_2, boolean\_3, boolean\_4**

**, ch\_1, ch\_2**

**, parent\_quantity**

**, normalized\_quantity**

**, component\_volume, volume, normalized\_volume**

**, parent\_volume**

**, component\_kg, assembly\_kg**

**, parent\_kg**

**, lvl , step**

**, path, pathNode**

**, parent\_branch, branch**

**, indentStr**

**, breadcrumbs**

**, reference\_date**

**, part\_reference\_date**

**, e\_issuedAfterExplDate**

**, e\_noPhantomRevision**

**) AS (**

**SELECT**

**p.part\_no AS root\_part, p.revision AS root\_rev**

**, p.status AS root\_status, p.status\_type AS root\_status\_type**

**, bh.preferred AS root\_preferred, bh.alternative AS root\_alternative**

**, p.issued\_date AS root\_issued\_date**

**, bi.num\_1 AS root\_density**

**, p.part\_no AS parent\_part, p.revision AS parent\_rev**

**, p.status AS parent\_status, p.status\_type AS parent\_status\_type**

**, bi.item\_number AS parent\_seq**

**, NULL AS parent\_function, NULL AS parent\_uom**

**, bi.component\_part AS part\_no, hh.revision, bi.plant, bi.item\_number**

**, coalesce(hh.issued\_date, hh.planned\_effective\_date) AS issued\_date**

**, coalesce(hh.obsolescence\_date, CURRENT\_DATE +1) AS obsolete\_date**

**, hh.status, hh.status\_code, hh.status\_type**

**, hh.class3\_id**

**, f.description AS function**

**, bh.preferred, bh.alternative, bh.base\_quantity**

**, bi.num\_5 AS phr, bi.quantity, bi.num\_1 AS density, bi.uom**

**, bi.num\_1, bi.num\_2, bi.num\_3, bi.num\_4, bi.num\_5**

**, bi.char\_1, bi.char\_2, bi.char\_3, bi.char\_4, bi.char\_5**

**, bi.date\_1, bi.date\_2**

**, bi.boolean\_1, bi.boolean\_2, bi.boolean\_3, bi.boolean\_4**

**, bi.ch\_1, bi.ch\_2**

**, bh.base\_quantity AS parent\_quantity**

**, bi.quantity \* 1.0 AS normalized\_quantity**

**, bi.quantity / bi.num\_1 AS component\_volume**

**, bi.quantity / bi.num\_1 AS volume**

**, bi.quantity / bi.num\_1 AS normalized\_volume**

**, 1.0 AS parent\_volume**

**, CASE**

**WHEN bi.uom = 'kg' THEN bi.quantity \* 1.0**

**ELSE NULL**

**END**

**, CASE**

**WHEN bi.uom = 'kg' THEN bi.quantity \* 1.0**

**ELSE NULL**

**END**

**, SUM(CASE**

**WHEN bi.uom = 'kg' THEN bi.quantity \* 1.0**

**ELSE NULL**

**END) OVER ()**

**, 1 AS lvl**

**, ROW\_NUMBER() OVER (ORDER BY bi.item\_number) -1**

**, CAST('0000' AS varchar2(120)) AS path**

**, to\_char(bi.item\_number, 'FM0999') AS pathNode**

**, CAST('' AS VARCHAR2(100)) AS parent\_branch**

**, CAST(to\_char(bi.item\_number, 'FM0999') AS varchar2(100)) AS branch**

**, CAST ('' AS VARCHAR2(200)) AS indentStr**

**, CAST(r.kw\_value || '.' || f.description AS VARCHAR2(254)) AS breadcrumbs**

**, CAST(p.reference\_date AS date)**

**, CAST(CASE**

**WHEN 'DEFAULT' = 'REFDATE' THEN p.reference\_date**

**WHEN 'DEFAULT' = 'DEFAULT' AND bi.component\_part LIKE 'X%'**

**THEN p.issued\_date**

**ELSE coalesce(p.obsolescence\_date, CURRENT\_DATE)**

**END AS date)**

**, CASE WHEN p.revision = 1 AND hh.revision = 1 AND p.reference\_date < coalesce(hh.issued\_date, hh.planned\_effective\_date)**

**THEN '1'**

**ELSE '0'**

**END AS e\_issuedAfterExplDate**

**, CASE WHEN hh.revision IS NULL THEN 1 ELSE 0 END AS e\_noPhantomRevision**

**FROM partList p**

**LEFT JOIN root\_function r ON (r.part\_no = p.part\_no)**

**LEFT JOIN bom\_header bh ON (**

**bh.part\_no = p.part\_no**

**AND bh.revision = p.revision**

**AND bh.preferred = p.preferred**

**)**

**LEFT JOIN bom\_item bi ON (**

**bi.part\_no = bh.part\_no**

**AND bi.revision = bh.revision**

**AND bi.plant = bh.plant**

**AND bi.alternative = bh.alternative**

**)**

**LEFT JOIN specList hh ON (**

**bi.component\_part = hh.part\_no**

**AND (f\_checkRefDate = 0**

**OR (**

**hh.issued\_date <= p.reference\_date**

**AND (hh.obsolescence\_date IS NULL OR p.reference\_date < hh.obsolescence\_date)**

**)**

**)**

**AND NOT EXISTS (**

**SELECT 1**

**FROM specification\_header h2a**

**JOIN status s2a ON (s2a.status = h2a.status)**

**WHERE h2a.part\_no = hh.part\_no**

**AND h2a.revision > hh.revision**

**AND s2a.status\_type IN ('HISTORIC', 'CURRENT')**

**AND ('DEFAULT' <> 'REFDATE'**

**OR (**

**h2a.issued\_date <= p.reference\_date**

**AND (h2a.obsolescence\_date IS NULL OR p.reference\_date < h2a.obsolescence\_date)**

**)**

**)**

**)**

**)**

**LEFT JOIN bom\_function f ON (f.characteristic\_id = bi.ch\_1)**

**WHERE 1 = 1**

**AND bh.preferred = 1**

**AND bh.alternative = FOC\_NONE**

**UNION ALL**

**SELECT**

**t.root\_part, t.root\_rev**

**, t.root\_status, t.root\_status\_type**

**, t.root\_preferred, t.root\_alternative**

**, t.root\_issued\_date**

**, t.root\_density**

**, t.part\_no AS parent\_part, t.revision AS parent\_rev**

**, t.status AS parent\_status, t.status\_type AS parent\_status\_type**

**, t.item\_number AS parent\_seq**

**, t.function AS parent\_function, t.uom AS parent\_uom**

**, bi.component\_part AS part\_no, hh.revision, t.plant, bi.item\_number**

**, hh.issued\_date, hh.obsolescence\_date AS obsolete\_date**

**, hh.status, hh.status\_code, hh.status\_type**

**, hh.class3\_id**

**, f.description AS function**

**, bh.preferred, bh.alternative, bh.base\_quantity**

**, bi.num\_5 AS phr, bi.quantity, bi.num\_1 AS density, bi.uom**

**, bi.num\_1, bi.num\_2, bi.num\_3, bi.num\_4, bi.num\_5**

**, bi.char\_1, bi.char\_2, bi.char\_3, bi.char\_4, bi.char\_5**

**, bi.date\_1, bi.date\_2**

**, bi.boolean\_1, bi.boolean\_2, bi.boolean\_3, bi.boolean\_4**

**, bi.ch\_1, bi.ch\_2**

**, bh.base\_quantity AS parent\_quantity**

**, DECODE(bh.base\_quantity, 0, 0, bi.quantity \* t.normalized\_quantity / bh.base\_quantity) AS normalized\_quantity**

**, DECODE(bi.num\_1, 0, 0, bi.quantity / bi.num\_1) AS component\_volume**

**, DECODE(bi.num\_1, 0, 0, bi.quantity / bi.num\_1) \* coalesce(**

**t.volume**

**, DECODE(bh.base\_quantity, 0, 0, bi.quantity / bh.base\_quantity)**

**, 1.0**

**) AS volume**

**, DECODE(bi.num\_1, 0, 0, bi.quantity / bi.num\_1) \* coalesce(**

**t.volume**

**, DECODE(bh.base\_quantity, 0, 0, bi.quantity \* t.normalized\_quantity / bh.base\_quantity)**

**, 1.0**

**) AS normalized\_volume**

**, t.volume AS parent\_volume**

**, CASE**

**WHEN bi.uom = 'kg'**

**THEN DECODE(bh.base\_quantity, 0, 0, bi.quantity \* t.normalized\_quantity / bh.base\_quantity)**

**ELSE NULL**

**END**

**, CASE**

**WHEN t.uom <> 'kg' AND bi.uom = 'kg'**

**THEN DECODE(bh.base\_quantity, 0, 0, bi.quantity \* t.normalized\_quantity / bh.base\_quantity)**

**ELSE NULL**

**END**

**, SUM(CASE**

**WHEN t.uom <> 'kg' AND bi.uom = 'kg'**

**THEN DECODE(bh.base\_quantity, 0, 0, bi.quantity \* t.normalized\_quantity / bh.base\_quantity)**

**ELSE 0**

**END) OVER (PARTITION BY t.branch)**

**, t.lvl +1 AS lvl**

**, ROW\_NUMBER() OVER (PARTITION BY t.branch ORDER BY bi.item\_number) -1**

**, substr(t.path || '.' || t.pathNode, 1, 120) AS path**

**, to\_char(bi.item\_number, 'FM0999') AS pathNode**

**, t.branch AS parent\_branch**

**, substr(t.branch || '.' || to\_char(bi.item\_number, 'FM0999'), 1, 100) AS branch**

**, substr(t.indentStr || '&nbsp;&nbsp;&nbsp;', 1, 200) AS indentStr**

**, cast(t.breadcrumbs || '.' || case**

**when f.description is null then '(null)'**

**when f.description = '' then '(null)'**

**else f.description**

**end AS varchar2(254))**

**, CAST(t.reference\_date AS date)**

**, CAST(CASE**

**WHEN bi.component\_part NOT LIKE 'X%' AND 'DEFAULT' = 'REFDATE' THEN t.reference\_date**

**WHEN bi.component\_part LIKE 'X%' AND 'DEFAULT' = 'REFDATE' THEN t.reference\_date**

**ELSE t.part\_reference\_date**

**END AS date)**

**, CASE WHEN t.root\_rev = 1**

**AND hh.revision = 1**

**AND t.reference\_date < coalesce(hh.issued\_date, hh.planned\_effective\_date)**

**THEN '1'**

**ELSE '0'**

**END AS e\_issuedAfterExplDate**

**, CASE WHEN hh.revision IS NULL THEN 1 ELSE 0 END AS e\_noPhantomRevision**

**FROM tree t**

**JOIN bom\_header bh ON (**

**bh.part\_no = t.part\_no**

**AND bh.revision = t.revision**

**AND bh.plant = t.plant**

**AND bh.preferred = 1**

**AND bh.alternative = FOC\_NONE**

**)**

**JOIN bom\_item bi ON (**

**bh.part\_no = bi.part\_no**

**AND bh.revision = bi.revision**

**AND bh.plant = bi.plant**

**AND bh.alternative = bi.alternative**

**)**

**LEFT JOIN componentList hh ON (**

**hh.part\_no = bi.component\_part**

**AND (f\_checkRefDate = 0**

**OR (**

**hh.issued\_date <= t.reference\_date**

**AND (hh.obsolescence\_date IS NULL OR t.reference\_date < hh.obsolescence\_date)**

**)**

**)**

**AND NOT EXISTS (**

**SELECT 1**

**FROM specification\_header h2a**

**JOIN status s2a ON (**

**s2a.status = h2a.status**

**AND s2a.status\_type IN ('HISTORIC', 'CURRENT')**

**)**

**WHERE h2a.part\_no = hh.part\_no**

**AND h2a.part\_no NOT LIKE 'X%'**

**AND h2a.revision > hh.revision**

**AND ('DEFAULT' = 'HIGHEST' OR h2a.issued\_date IS NOT NULL)**

**UNION ALL**

**SELECT 1**

**FROM specification\_header h2a**

**JOIN status s2a ON (**

**s2a.status = h2a.status**

**AND s2a.status\_type IN ('HISTORIC', 'CURRENT')**

**)**

**WHERE h2a.part\_no = hh.part\_no**

**AND h2a.part\_no LIKE 'X%'**

**AND h2a.revision > hh.revision**

**AND ('DEFAULT' = 'HIGHEST' OR h2a.issued\_date IS NOT NULL)**

**)**

**)**

**LEFT JOIN bom\_function f ON (f.characteristic\_id = bi.ch\_1)**

**WHERE 1 = 1**

**AND hh.status\_type = 'FOC\_NONE'**

**AND t.lvl <= FOC\_NONE**

**AND t.revision IS NOT NULL**

**AND t.function IS NOT NULL AND t.function NOT LIKE '%FOC\_NONE'**

**) CYCLE part\_no SET e\_cyclic TO '1' DEFAULT '0'**

**, templateBoM (**

**tpl\_root\_part, tpl\_root\_revision, tpl\_root\_description**

**, tpl\_part\_no, tpl\_revision**

**, tpl\_description**

**, tpl\_frame\_id, tpl\_frame\_rev**

**, tpl\_item\_number**

**, hierarchy**

**, function\_description**

**, function\_code**

**, functiongrp**

**, mask**

**, lvl**

**) AS (**

**SELECT**

**sh.part\_no, sh.revision, sh.description**

**, sh.part\_no, sh.revision**

**, sh.description**

**, sh.frame\_id, sh.frame\_rev**

**, 1**

**, '1'**

**, (SELECT kw\_value FROM root\_function WHERE part\_no = sh.part\_no)**

**, (SELECT kw\_value FROM root\_function WHERE part\_no = sh.part\_no)**

**, (SELECT kw\_value FROM root\_function WHERE part\_no = sh.part\_no)**

**, '%' || (SELECT kw\_value FROM root\_function WHERE part\_no = sh.part\_no)**

**, 0**

**FROM specification\_header sh**

**JOIN status s ON (s.status = sh.status)**

**WHERE sh.part\_no = 'NONE'**

**AND s.status\_type = 'CURRENT'**

**UNION ALL**

**SELECT**

**t.tpl\_root\_part, t.tpl\_root\_revision, t.tpl\_root\_description**

**, bi.component\_part**

**, bish.revision**

**, bish.description**

**, bish.frame\_id, bish.frame\_rev**

**, bi.item\_number**

**, '1.' || coalesce(bi.char\_1, '0.' || to\_char(bi.item\_number, 'FM0999'))**

**, bi.char\_2**

**, bic.description**

**, bi.char\_3**

**, bi.char\_4**

**, t.lvl +1**

**FROM templateBoM t**

**JOIN bom\_item bi ON (bi.part\_no = t.tpl\_part\_no and bi.revision = t.tpl\_revision)**

**LEFT JOIN specification\_header bish ON (bish.part\_no = bi.component\_part)**

**LEFT JOIN status bis ON (bis.status = bish.status)**

**LEFT JOIN characteristic bic ON (bic.characteristic\_id = bi.ch\_1)**

**WHERE bis.status\_type = 'CURRENT'**

**)**

**, filtered (**

**root\_part, root\_rev, root\_status**

**, part\_no, revision, status\_type**

**, breadcrumbs, item\_number**

**, branch, lvl, indentPart**

**, tpl\_root\_part, tpl\_root\_description**

**, tpl\_part\_no, tpl\_revision, tpl\_description, tpl\_item\_number**

**, hierarchy, functiongrp, function\_code, function\_description, mask**

**, phr, quantity, density, uom**

**, num\_1, num\_2, num\_3, num\_4, num\_5**

**, char\_1, char\_2, char\_3, char\_4, char\_5**

**, date\_1, date\_2**

**, boolean\_1, boolean\_2, boolean\_3, boolean\_4**

**, ch\_1, ch\_2**

**, normalized\_quantity**

**, component\_kg, assembly\_kg**

**) AS (**

**SELECT**

**p.part\_no, p.revision, p.status\_type**

**, p.part\_no, p.revision, p.status\_type**

**, r.kw\_value, 0**

**, NULL, 0, p.part\_no**

**, tpl.tpl\_root\_part, tpl.tpl\_root\_description**

**, tpl.tpl\_part\_no, tpl.tpl\_revision, tpl.tpl\_description, tpl.tpl\_item\_number**

**, tpl.hierarchy, tpl.functiongrp, tpl.function\_code, tpl.function\_description, tpl.mask**

**, NULL, 1.0, 1.0, 'pcs'**

**, NULL, NULL, NULL, NULL, NULL**

**, NULL, NULL, NULL, NULL, NULL**

**, NULL, NULL**

**, NULL, NULL, NULL, NULL**

**, NULL, NULL**

**, 1.0**

**, NULL, NULL**

**FROM partList p**

**JOIN root\_function r ON (r.part\_no = p.part\_no)**

**JOIN templateBoM tpl ON (tpl.lvl = 0)**

**UNION ALL**

**SELECT**

**tree.root\_part, tree.root\_rev, tree.root\_status\_type**

**, tree.part\_no, tree.revision, tree.status\_type**

**, tree.breadcrumbs, tree.item\_number**

**, tree.branch, tree.lvl, tree.indentStr || tree.part\_no**

**, tpl.tpl\_root\_part, tpl.tpl\_root\_description**

**, tpl.tpl\_part\_no, tpl.tpl\_revision, tpl.tpl\_description, tpl.tpl\_item\_number**

**, tpl.hierarchy, tpl.functiongrp, tpl.function\_code, tpl.function\_description, tpl.mask**

**, tree.phr, tree.quantity, tree.density, tree.uom**

**, tree.num\_1, tree.num\_2, tree.num\_3, tree.num\_4, tree.num\_5**

**, tree.char\_1, tree.char\_2, tree.char\_3, tree.char\_4, tree.char\_5**

**, tree.date\_1, tree.date\_2**

**, tree.boolean\_1, tree.boolean\_2, tree.boolean\_3, tree.boolean\_4**

**, tree.ch\_1, tree.ch\_2**

**, tree.normalized\_quantity**

**, tree.component\_kg, tree.assembly\_kg**

**FROM tree**

**JOIN templateBoM tpl ON (tree.breadcrumbs LIKE tpl.mask)**

**WHERE EXISTS ( SELECT 1**

**FROM bom\_header node**

**WHERE node.part\_no = tree.part\_no**

**AND node.revision = tree.revision**

**AND node.alternative = tree.alternative**

**AND node.plant = tree.plant**

**)**

**)**

**SELECT f.\***

**, part.description AS part\_description**

**, weight.weight, weight.uom AS weight\_uom**

**FROM filtered f**

**LEFT JOIN part ON (part.part\_no = f.part\_no)**

**LEFT JOIN avspecification\_weight weight ON (weight.part\_no = f.part\_no)**

**ORDER BY**

**f.hierarchy, f.functiongrp, f.item\_number**

**;**

**END**

**TABLE FILE SQLBOM**

**SUM**

**PART\_DESCRIPTION**

**INDENTPART**

**QUANTITY**

**UOM**

**WEIGHT**

**WEIGHT\_UOM**

**COMPUTE COMP\_WEIGHT/D20.3Sc = IF UOM EQ 'kg' THEN QUANTITY ELSE QUANTITY \* WEIGHT;**

**MAX.COMPONENT\_KG**

**BY TPL\_ROOT\_PART**

**BY TPL\_ROOT\_DESCRIPTION**

**BY ROOT\_PART**

**BY ROOT\_REV/I3**

**BY HIERARCHY**

**BY FUNCTIONGRP**

**BY FUNCTION\_CODE/A120**

**BY FUNCTION\_DESCRIPTION**

**BY MASK**

**BY ITEM\_NUMBER**

**BY PART\_NO**

**BY REVISION**

**BY LVL/I3**

**WHERE BREADCRUMBS IS NOT MISSING;**

**ON TABLE HOLD AS INT00040H\_BOM FORMAT XFOCUS**

**END**

**0 NUMBER OF RECORDS IN TABLE= 0 LINES= 0**

**TABLE FILE SQLBOM**

**SUM**

**COMPUTE UNION/A6 = IF LAST TPL\_REVISION EQ 0 THEN '' ELSE 'UNION';**

**COMPUTE TUPLE/A640 = 'SELECT ''' || HIERARCHY**

**|| ''',''' || FUNCTIONGRP**

**|| ''',''' || FUNCTION\_DESCRIPTION**

**|| ''',''' || TPL\_PART\_NO**

**|| ''',' || FPRINT(TPL\_REVISION, 'P9', 'A9')**

**|| ',''' || PART\_NO**

**|| ''',' || FPRINT(REVISION, 'P9', 'A9')**

**|| ',' || FPRINT(ITEM\_NUMBER, 'P5', 'A5')**

**|| ',''' || BRANCH**

**|| ''' FROM dual';**

**BY HIERARCHY NOPRINT**

**BY FUNCTIONGRP NOPRINT**

**BY TPL\_PART\_NO NOPRINT**

**BY TPL\_REVISION NOPRINT**

**BY PART\_NO NOPRINT**

**BY REVISION NOPRINT**

**ON TABLE HOLD AS INT00040H\_TPLBOMPARTS FORMAT ALPHA**

**ON TABLE SET HOLDLIST PRINTONLY**

**ON TABLE SET ASNAMES ON**

**END**

**0 NUMBER OF RECORDS IN TABLE= 0 LINES= 0**

**0 ERROR AT OR NEAR LINE 31 IN PROCEDURE INT00040P\_Properties**

**(FOC305) SPECIFIED LABEL NOT FOUND: :NOBOM**