



**Data Model Changes Regarding
SD Document Flow (Table VBFA)**

Document Version	Status	Date
1.0	final	October, 16, 2015
1.1	final	April 06, 2018
1.2	final	April 06, 2018

Table of Content

1	Change Log	2
2	Data Model Changes	3
2.1	General Approach	3
2.1.1	Motivation	3
2.1.2	Model Changes	3
2.1.3	Field CMETH	4
2.1.4	Re-introduction of field VBFA-STUFE	4
2.1.5	Release dependency of VBFA-STUFE and indirect relationships	5
3	Adaptation of Customer Code	7
3.1	Utilities from SAP Supporting the Adaptation	7
3.1.1	How to Handle Enhancements	7
3.2	Coding Adaptation - Examples	8
3.2.1	Select Statement with STUFE = '00' (Direct Successors)	8
3.2.2	Select Statement with ORDER BY PRIMARY KEY	8
3.2.3	Select Statement with Direct and Non-Direct-Successors	9
3.2.4	Build XVBFA from Predecessor	10
3.2.5	Select Statement with Non-Direct Predecessors	11
3.2.6	Select Statement with a Range of SD Document Categories	12
3.2.7	Call of function module RV_ORDER_FLOW_INFORMATION	12
3.2.8	READ TABLE XVBFA WITH KEY <STRUCTURE> BINARY SEARCH	14
3.2.9	READ TABLE xvbfa FROM Is_xvbfa TRANSPORTING NO FIELDS.	14

1 Change Log

Version	Date	Change
1.0	2015-10-16	Initial version
1.1	2018-04-06	<p>Chapters 2.1.4, 2.1.5: VBFA-STUFE is supported again as of S/4HANA 1709 and equivalent Support packages in 1610 and 1511.</p> <p>Chapter 3.2: Impact of VBFA-STUFE on Code adaptations in S/4HANA</p> <p>Chapter 3.2.7: Calls of function module RV_ORDER_FLOW_INFORMATION</p> <p>Chapters 3.2.8, 3.2.9: Additional adaptation pattern: READ TABLE with key <key structure></p>
1.2	2018-04-06	<p>Chapter 3.2.2: SELECT with ORDER BY PRIMARY KEY</p>

2 Data Model Changes

2.1 General Approach

2.1.1 Motivation

The Sales & Distribution document flow manages relationships between Sales & Distribution (and other) documents to describe how these documents (or document items) are interconnected in business process instances.

From a persistency point of view, the document flow is represented by **database table VBFA** storing the relationships. They are stored together with additional (partially redundant) data to facilitate faster access to the relationships, and to enable faster evaluation of the data of connected documents.

For customers with a large volume of business documents, database table VBFA can grow to become quite huge. This results in a big memory footprint for SAP HANA-based systems. As an example, if VBFA contains 250 million rows cause, memory consumption would exceed 16 GB.

- Because the semantic primary key has several (character) fields, it consumes quite a lot of space as a result of the way how SAP HANA handles primary key indices.
- VBFA not only stores direct connections between documents. There are a number of cases where paths in the document flow graph are stored as separate rows, even though this path information could also be calculated on the fly if all direct relationships are available.

Therefore, one goal of the envisioned model changes was to reduce the memory footprint of database table VBFA. Additionally, there are some further limitations related to the document flow. One such limitation, the SD Document Category (part of the original primary key of VBFA), was a one-character code field in which nearly all code values had been used. Additional document categories had to be stored in a parallel extended category field "VBTYP_EXT" (as of EhP 7), with a number of drawbacks. Therefore, the SD Document Category was an early candidate for field length extension.

2.1.2 Model Changes

The following model changes have been implemented:

- VBFA Primary Key change
- SD Document Category VBTYP: Field Length Extension
- Removal of Extended SD Document Category VBTYPEXT_V, VBTYPEXT_N (these fields had been introduced as of EhP 7)
- Removal of Column STUFE in S/4HANA 1511/1610
- Removal of column CMETH in S/4HANA 1511

VBFA is still available as a database table. Besides table columns STUFE, CMETH, VBTYPEXT_V and VBTYPEXT_N, no columns have been removed. Hence, most of the accesses to VBFA still work. No replacement by a CDS view has taken place. The most disruptive change was the removal of column

STUFE, since it implies less entries in VBFA. Therefore, some selections do not return the same result as before.

New VBFA vs. Old VBFA

Transparent Table VBFA Active						Transparent Table VBFA Active							
Sales Document Flow						Sales Document Flow							
Attributes Delivery and Maintenance Fields Input Help/Check Currency/Quantity Fields						Attributes Delivery and Maintenance Fields Entry Help/Check Currency/Quantity Fields							
Field	Key In.	Data element	Data Type	Length	Dec.	Short Description	Field	Key In.	Data element	Data Type	Length	Dec.	Short Description
MANDT	<input checked="" type="checkbox"/>	MANDT	CHAR	3		Client	MANDT	<input checked="" type="checkbox"/>	MANDT	CHAR	3		Client
SDUFI	<input checked="" type="checkbox"/>	SD_DOC_REL_UFI	RAW	14		SD Unique Document Relationship Identification	SDUFI	<input checked="" type="checkbox"/>	SD_DOC_REL_UFI	RAW	14		SD Unique Document Relationship Identification
VBELV	<input checked="" type="checkbox"/>	VBELN_VOR	CHAR	10		Preceding sales and distribution document	VBELV	<input checked="" type="checkbox"/>	VBELN_VOR	CHAR	10		Preceding sales and distribution document
POSIV	<input checked="" type="checkbox"/>	POSBE_VOR	CHAR	6		Preceding item of an SD document	POSIV	<input checked="" type="checkbox"/>	POSBE_VOR	CHAR	6		Preceding item of an SD document
VBELH	<input checked="" type="checkbox"/>	VBELN_NACH	CHAR	10		Subsequent sales and distribution document	VBELH	<input checked="" type="checkbox"/>	VBELN_NACH	CHAR	10		Subsequent sales and distribution document
POSNN	<input checked="" type="checkbox"/>	POSBE_NACH	CHAR	6		Subsequent item of an SD document	POSNN	<input checked="" type="checkbox"/>	POSBE_NACH	CHAR	6		Subsequent item of an SD document
VBSTYP_H	<input checked="" type="checkbox"/>	VBSTYP	CHAR	1		Document category of subsequent document	VBSTYP_H	<input checked="" type="checkbox"/>	VBSTYP	CHAR	1		Document category of subsequent document
RFBSD	<input checked="" type="checkbox"/>	RFBSD	QDAN	15		Referenced quantity in base unit of measure	RFBSD	<input checked="" type="checkbox"/>	RFBSD	QDAN	15		Referenced quantity in base unit of measure
MEINS	<input checked="" type="checkbox"/>	MEINS	UNIT	3		Base Unit of Measure	MEINS	<input checked="" type="checkbox"/>	MEINS	UNIT	3		Base Unit of Measure
RFBST	<input checked="" type="checkbox"/>	RFBST	CHAR	15		Reference value	RFBST	<input checked="" type="checkbox"/>	RFBST	CHAR	15		Reference value
WAEKS	<input checked="" type="checkbox"/>	WAEKS_V	COFY	5		Statistics currency	WAEKS	<input checked="" type="checkbox"/>	WAEKS_V	COFY	5		Statistics currency
VBSTYP_V	<input checked="" type="checkbox"/>	VBSTYP	CHAR	4		SD Document Category (Long)	VBSTYP_V	<input checked="" type="checkbox"/>	VBSTYP	CHAR	4		SD Document Category (Long)
FKMTH	<input checked="" type="checkbox"/>	FKMTH	CHAR	1		Quantity is calculated positively, negatively or not at all	FKMTH	<input checked="" type="checkbox"/>	FKMTH	CHAR	1		Quantity is calculated positively, negatively or not at all
TAQTI	<input checked="" type="checkbox"/>	TAQTI	CHAR	1		ID: MM-WM transfer order confirmed	TAQTI	<input checked="" type="checkbox"/>	TAQTI	CHAR	1		ID: MM-WM transfer order confirmed
ERDAT	<input checked="" type="checkbox"/>	ERDAT	DATS	8		Date on which the record was created	ERDAT	<input checked="" type="checkbox"/>	ERDAT	DATS	8		Date on which the record was created
ERZET	<input checked="" type="checkbox"/>	ERZET	TIMS	6		Entry time	ERZET	<input checked="" type="checkbox"/>	ERZET	TIMS	6		Entry time
MATNR	<input checked="" type="checkbox"/>	MATNR	CHAR	10		Material Number	MATNR	<input checked="" type="checkbox"/>	MATNR	CHAR	10		Material Number
BMART	<input checked="" type="checkbox"/>	BMART	CHAR	3		Requirement type (Inventory Management)	BMART	<input checked="" type="checkbox"/>	BMART	CHAR	3		Requirement type (Inventory Management)
PLANT	<input checked="" type="checkbox"/>	PLANT	CHAR	1		Plant	PLANT	<input checked="" type="checkbox"/>	PLANT	CHAR	1		Plant
LOKON	<input checked="" type="checkbox"/>	LOKON	CHAR	3		Warehouse Number / Warehouse Complex	LOKON	<input checked="" type="checkbox"/>	LOKON	CHAR	3		Warehouse Number / Warehouse Complex
ADAT	<input checked="" type="checkbox"/>	ADAT	DATS	8		Date of Last Change	ADAT	<input checked="" type="checkbox"/>	ADAT	DATS	8		Date of Last Change
FKTYP	<input checked="" type="checkbox"/>	FKTYP	CHAR	1		Billing category	FKTYP	<input checked="" type="checkbox"/>	FKTYP	CHAR	1		Billing category
BROEW	<input checked="" type="checkbox"/>	BROEW_15	QDAN	15		Gross weight	BROEW	<input checked="" type="checkbox"/>	BROEW_15	QDAN	15		Gross weight
GEWEI	<input checked="" type="checkbox"/>	GEWEI	UNIT	3		Weight Unit	GEWEI	<input checked="" type="checkbox"/>	GEWEI	UNIT	3		Weight Unit
VOLUM	<input checked="" type="checkbox"/>	VOLUM_15	QDAN	15		Volume	VOLUM	<input checked="" type="checkbox"/>	VOLUM_15	QDAN	15		Volume
VOLER	<input checked="" type="checkbox"/>	VOLER	UNIT	3		Volume unit	VOLER	<input checked="" type="checkbox"/>	VOLER	UNIT	3		Volume unit
FFLNR	<input checked="" type="checkbox"/>	FFLNR	CHAR	10		Billing plan number / invoicing plan number	FFLNR	<input checked="" type="checkbox"/>	FFLNR	CHAR	10		Billing plan number / invoicing plan number
FFLTS	<input checked="" type="checkbox"/>	FFLTS	CHAR	6		Item for billing plan/invoice plan/payment cards	FFLTS	<input checked="" type="checkbox"/>	FFLTS	CHAR	6		Item for billing plan/invoice plan/payment cards
RFBSD_FLD	<input checked="" type="checkbox"/>	RFBSD_FLTV	FLTP	14		Referenced quantity in sales unit (float)	RFBSD_FLD	<input checked="" type="checkbox"/>	RFBSD_FLTV	FLTP	14		Referenced quantity in sales unit (float)
RFBSD_FLI	<input checked="" type="checkbox"/>	RFBSD_FLIT	FLTP	14		Referenced quantity in base unit of measure (float)	RFBSD_FLI	<input checked="" type="checkbox"/>	RFBSD_FLIT	FLTP	14		Referenced quantity in base unit of measure (float)
VBOME	<input checked="" type="checkbox"/>	VBOME	UNIT	3		Sales unit	VBOME	<input checked="" type="checkbox"/>	VBOME	UNIT	3		Sales unit
ABDES	<input checked="" type="checkbox"/>	ABDES_CH	FLTP	16		Guaranteed (factor between 0 and 1)	ABDES	<input checked="" type="checkbox"/>	ABDES_CH	FLTP	16		Guaranteed (factor between 0 and 1)
SOBKI	<input checked="" type="checkbox"/>	SOBKI	CHAR	1		Special Stock Indicator	SOBKI	<input checked="" type="checkbox"/>	SOBKI	CHAR	1		Special Stock Indicator
SOBWH	<input checked="" type="checkbox"/>	LVS_SOBWH	CHAR	16		Special Stock Number	SOBWH	<input checked="" type="checkbox"/>	LVS_SOBWH	CHAR	16		Special Stock Number
KBREF	<input checked="" type="checkbox"/>	LVS_KBREF	CHAR	1		Indicator Inventory Management active	KBREF	<input checked="" type="checkbox"/>	LVS_KBREF	CHAR	1		Indicator Inventory Management active
BTGEW	<input checked="" type="checkbox"/>	BTGEW	QDAN	13		Net weight	BTGEW	<input checked="" type="checkbox"/>	BTGEW	QDAN	13		Net weight
LOGSYS	<input checked="" type="checkbox"/>	LOGSYS	CHAR	10		Logical system	LOGSYS	<input checked="" type="checkbox"/>	LOGSYS	CHAR	10		Logical system
MBSTA	<input checked="" type="checkbox"/>	MBSTA	CHAR	1		Goods movement status	MBSTA	<input checked="" type="checkbox"/>	MBSTA	CHAR	1		Goods movement status
CMETH	<input checked="" type="checkbox"/>	CMETH	CHAR	1		Quantity Conversion Method	CMETH	<input checked="" type="checkbox"/>	CMETH	CHAR	1		Quantity Conversion Method
STUFE	<input checked="" type="checkbox"/>	STUFE	CHAR	4		Material Document Year	STUFE	<input checked="" type="checkbox"/>	STUFE	CHAR	4		Material Document Year
VBSTYPKT_V	<input checked="" type="checkbox"/>	VSD_VBSTYP_KXT_V	CHAR	4		Extension of SD document category of preceding document	VBSTYPKT_V	<input checked="" type="checkbox"/>	VSD_VBSTYP_KXT_V	CHAR	4		Extension of SD document category of preceding document
VBSTYPKT_H	<input checked="" type="checkbox"/>	VSD_VBSTYP_KXT_H	CHAR	4		Extension of subsequent SD Document Category	VBSTYPKT_H	<input checked="" type="checkbox"/>	VSD_VBSTYP_KXT_H	CHAR	4		Extension of subsequent SD Document Category

© 2013 SAP AG or an SAP affiliate company. All rights reserved.

3

2.1.3 Field CMETH

Field CMETH was removed in S/4HANA 1511. It is needed for Industry Solution IS-OIL which is supported as part of S/4HANA as of 1610.

Therefore field CMETH was added again in 1610.

For IS-OIL customers performing the S/4HANA system conversion the temporary lack of the field in 1511 has no impact because the target release of the conversion must be 1610 or higher for IS-OIL customers.

2.1.4 Re-introduction of field VBFA-STUFE

While the initial model changes were decided the target architecture for information lifecycle management was still under discussion. Data aging was one preferred direction for the phase-out of data like documents.

Finally, archiving is supported in an unchanged way in S/4HANA for on-Premise installations.

Therefore it can happen that documents in the middle of a document flow are already archived while the other documents still reside in the database.

In this case not all **direct** relationships are available any more in the database. Hence, the indirect relationships cannot be calculated on-the fly.

Therefore the indirect relationships must still/again be stored also in S/4HANA.

As a consequence the field VBFA-STUFE was re-introduced again and all indirect relationships in VBFA have to be rebuilt in case they were deleted or not stored in S/4HANA 1511/1610.

2.1.5 Release dependency of VBFA-STUFE and indirect relationships

As indicated in the long text description of note 2470721 the indirect relationships in the SD document flow table VBFA are handled differently in the early releases of S/4HANA:

In the Simplifications for S/4HANA, the field STUFE had been removed from the database table VBFA.

In the initial releases of S/4HANA:

- only direct relationships are stored in VBFA
- indirect relationships are not stored in VBFA
- indirect relationships are deleted in the S/4HANA conversion

The field STUFE is introduced again and indirect relationships are stored again in table VBFA as of the following releases and SP levels:

- 1709 (SP00)
- 1610 SP02 = FPS2 (feature pack stack 2)
- 1511 SP04
- Via correction instructions of note 2418242 in 1610 SP00/SP01
- Via correction instructions of note 2418242 in 1511 SP00/SP01/SP02/SP03

A reconstruction of missing indirect docflow entries (“VBFA reconstruction”) is **necessary** in the following cases:

- New installation on 1610 SP00/SP01/SP02 or
- New installation on 1511 SP00/SP01/SP02/SP03/SP04
- System conversion from SAP ERP to 1610 SP00/SP01 or
- System conversion from SAP ERP to 1511 SP00/SP01/SP02/SP03

A reconstruction of VBFA is **not necessary** in the following cases:

- New installation on 1709 or higher release
- New installation on 1610 SP03 or higher SP level
- New installation on 1511 SP05 or higher SP level
- System conversion from SAP ERP to 1709 or higher release
- System conversion from SAP ERP to 1610 SP02 or higher SP level
- System conversion from SAP ERP to 1511 SP04 or higher SP level

The upgrade to 1709 from 1511 or 1610 can only happen if the reconstruction of missing indirect docflow entries has been executed before the upgrade.

This is enforced by a so-called SI check (Simplification Item check) that is executed in the upgrade preparation to 1709. Report /SDF/RC_START_CHECK shows a red traffic light for Simplification item “SI21: SD_VBFA_STUFE” if the VBFA reconstruction is missing.

Note 2418242 describes the steps that are recommended for VBFA reconstruction.

Hence as of 1709 the field VBFA-STUFE and the indirect relationships have the same functionality and the same persistence like in SAP ERP.

In 1610 and in 1511 it depends on the SP level and the system history whether indirect relationships are persisted like in SAP ERP.

For the custom code changes the needed adaptations depend on the release/SP level of the S/4HANA system. For better readability of the examples in this cookbook we will use the following notation:

- “After VBFA reconstruction” stands for “VBFA-STUFE is available and indirect relationships are completely persisted in VBFA”.
- “Before VBFA reconstruction” stands for “Indirect relationships may not be completely persisted in VBFA”.
- “After re-introduction of field VBFA-STUFE” means that the field exists in the database table, but indirect relationships may or may not be completely persisted in table VBFA.
- “Without field VBFA-STUFE” means that the field does not exist in the database table, and therefore also no indirect relationships are persisted.

If you are executing code adaptations you must match the release status of your system to the respective case for indirect relationships in VBFA, and apply this case per example in this cookbook.

If you have executed code adaptations according to this cookbook before VBFA-STUFE was re-introduced and you want to follow-up after an S/4HANA upgrade to a release with VBFA-STUFE (e.g. 1709) then refer to note 2470721 to see which delta adaptations may be necessary due to the re-introduction of VBFA-STUFE.

3 Adaptation of Customer Code

3.1 Utilities from SAP Supporting the Adaptation

In SD, all accesses to the SD document flow table VBFA in which non-direct successors or predecessors (STUFE > 0) are selected are subject to adaptation. Without VBFA reconstruction these types of accesses have to be calculated on the fly by using class CL_SD_DOCUMENT_FLOW_RT.

Method	Description
HAS_SUCCESSOR	Check if a successor of a specific type exists
HAS_PREDECESSOR	Check if a predecessor of a specific type exists
GET_SUCCESSORS	Get successors of a specific type and path length
GET_PREDECESSORS	Get predecessors of a specific type and path length
CONSTRUCTOR	Constructor
BUILD_XVBFA_FROM_PREDECESSOR	Get all predecessors up to a maximum path length

In SAP GUI, the document flow based on VBFA always requires the field STUFE to display the hierarchical flow correctly. Function module RV_ORDER_FLOW_INFORMATION takes care of building up the VBFA entries with STUFE correctly by using the new structure VBFAS.

For the SD document category VB Typ, an interface with constants and a utility class are delivered:

- if_sd_doc_category
- cl_sd_doc_category_util

For details on this interface and this class, see the cookbook for the VB Typ Field Length Extension, which is attached to the same SAP Note 2198647.

3.1.1 How to Handle Enhancements

Enhancements of the SD Document Flow (for example, append fields to VBFA) remain unchanged and can still be used. If the enhancement contains data that belongs to non-direct successors or predecessors (STUFE > 0), the enhancement of class CL_SD_DOCUMENT_FLOW_RT is necessary. In this case, please create a subclass of class CL_SD_DOCUMENT_FLOW_RT in the customer namespace and redefine the methods you need to adapt.

3.2 Coding Adaptation - Examples

In most cases, adapting source code requires some knowledge about the business process and the different possible use cases which can occur.

In the following examples 'Old' denotes coding as it has been implemented in SAP ERP. 'New' denotes the recommendation for the new adapted version of the coding in S/4HANA.

In some cases the coding in S/4HANA depends on the release and SP level due to the re-introduction of VBFA-STUFE. The dependencies are elaborated per example.

3.2.1 Select Statement with STUFE = '00' (Direct Successors)

The old coding selects direct successors (STUFE = '00') for the items (posnn ne '000000') of document number i_vbeln with a filter on the document category of the document itself (vbtyp_v) and of the successor document (vbtyp_n).

Intermediate version:

As long as STUFE is missing in table VBFA you have to remove the STUFE = '00' part from the select statement because it's no longer valid. Without field STUFE there are only direct successors or direct predecessors in the SD document flow table.

This changes again with the re-introduction of the field STUFE and the reconstruction of indirect relationships.

Old, and again after VBFA reconstruction:

```
SELECT * FROM vbfa INTO CORRESPONDING FIELDS OF
TABLE LT_VBFA
WHERE vbelv = i_vbeln
AND vbtyp_v = 'J'
AND vbtyp_n = 'Q'
AND posnn NE '000000'
AND stufe EQ '00'.
```

Intermediate :

```
SELECT * FROM vbfa INTO CORRESPONDING FIELDS OF
TABLE LT_VBFA
WHERE vbelv = i_vbeln
AND vbtyp_v = IF_SD_DOC_CATEGORY=>DELIVERY
AND vbtyp_n = IF_SD_DOC_CATEGORY=>WMS_TRANS_ORDER
AND posnn NE '000000'.
```

General and release-independent:

It is recommended to call method GET_SUCCESSORS of class CL_SD_DOCUMENT_FLOW_RT, with IV_PATH_LENGTH = '1'.

This method call works independently from the release level in all S/4HANA systems.

3.2.2 Select Statement with ORDER BY PRIMARY KEY

The old coding contains a select statement on VBFA with the addition ORDER BY PRIMARY KEY.

Old coding (version 0):

```
select * from vbfa into table lt_vbfa
where vbelv = <ls_vbak>-vbeln
and stufe = '00'
order by primary key.
```

Adapted coding in S/4HANA as long as VBFA-STUFE is missing (version 1):

WHERE clause on STUFE must be eliminated and ORDER BY statement must be adapted due to the changed primary key.

```
select * from vbfa into table lt_vbfa
      where vbelv = <ls_vbak>-vbeln
      order by vbelv posnv vbeln posnn vbtyp_n.
```

Adapted coding in S/4HANA as soon as VBFA-STUFE is added again (version 2):

```
select * from vbfa into table lt_vbfa
      where vbelv = <ls_vbak>-vbeln
      and   stufe = '00'
      order by vbelv posnv vbeln posnn vbtyp_n.
```

General and release-independent:

It is recommended to call method GET_SUCCESSORS of class CL_SD_DOCUMENT_FLOW_RT

- with IV_PATH_LENGTH = '1' for only direct successors

- with a higher value for IV_PATH_LENGTH to cover also indirect successors

This method call works independently from the release level in all S/4HANA systems.

You must not rely on a certain sort order of the internal tables which are returned by the method.

3.2.3 Select Statement with Direct and Non-Direct-Successors

You need to call method GET_SUCCESSIONS of class CL_SD_DOCUMENT_FLOW_RT because you are interested in the non-direct successors. Unless VBFA is reconstructed these entries are no longer stored in the SD document flow table.

The method call works independently from the release level in all S/4HANA systems.

Old:

```
CHECK NOT lt_sorders IS INITIAL.
SELECT * FROM vbfa
      INTO TABLE pt_vbfa
      WHERE vbelv IN lt_sorders.

LOOP AT pt_vbfa INTO ls_vbfa.
  IF ( ls_vbfa-vbtyp_n CA vbtyp_lief AND
      ls_vbfa-stufe EQ 0 ) OR
      ( ls_vbfa-vbtyp_n CA vbtyp_fakt AND
      ls_vbfa-stufe LE 1 ).
    CONTINUE.
  ELSE.
    DELETE TABLE pt_vbfa FROM ls_vbfa.
  ENDIF.
ENDLOOP.
```

New:

```
check not lt_sorders is initial.
data(lo_docflow_rt) = new cl_sd_document_flow_rt( ).

* deliveries
call method lo_docflow_rt->get_successors
  exporting
    it_document      = lt_sorders
    it_successor_type = cl_sd_doc_category_util=>rg_delivery_outgoing( )
    iv_path_length   = 1
  importing
    et_item          = lt_successors.
loop at lt_successors into ls_successors.
  ls_vbfa-vbelv     = ls_successors-predecessor_id.
  ls_vbfa-vbtyp_n   = ls_successors-successor_type.
  ls_vbfa-vbeln     = ls_successors-successor_id.
  ls_vbfa-posnn     = ls_successors-successor_item_id.
  insert ls_vbfa into table pt_vbfa.
endloop.
```

```

* invoices
call method lo_docflow_rt->get_successors
  exporting
    it_document      = lt_sorders
    it_successor_type = cl_sd_doc_category_util=>rg_any_invoice( )
    iv_path_length   = 2
  importing
    et_item          = lt_successors.
loop at lt_successors into ls_successors.
  ls_vbfa-vbelv     = ls_successors-predecessor_id.
  ls_vbfa-vbtyp_n   = ls_successors-successor_type.
  ls_vbfa-vbeln     = ls_successors-successor_id.
  ls_vbfa-posnn     = ls_successors-successor_item_id.
  insert ls_vbfa into table pt_vbfa.
endloop.

```

3.2.4 Build XVBFA from Predecessor

This special method BUILD_XVBFA_FROM_PREDECESSOR in class CL_SD_DOCUMENT_FLOW_RT is introduced to build up the XVBFA from predecessors up to the maximal path length.

Note that, like in chapter 3.2.8, the implicit specification of XVBRK_KEY in the READ TABLE command has to be replaced by an explicit specification of the key fields.

The new coding version works for all S/4HANA releases. It does not depend on the existence of VBFA-STUFE or on VBFA reconstruction. Indirect relationships are handled accordingly within the encapsulation of the method BUILD_XVBFA_FROM_PREDECESSOR.

Old:

```

FORM GN_XVBFA_AUFBAUEN USING DA_VGBEL.

DATA: DA_TABIX LIKE SY-TABIX,
      DA_XVBFA LIKE XVBFA OCCURS 0.

CHECK: DA_VGBEL NE BELNR_VBFA.
BELNR_VBFA = DA_VGBEL.

XVBRK_KEY-MANDT = SY-MANDT.
XVBRK_KEY-VBELN = DA_VGBEL.

READ TABLE XVBFA WITH KEY XVBRK_KEY BINARY SEARCH.
IF SY-SUBRC EQ 8 AND SY-TABIX EQ 1.      "Select into empty XVBFA
  SELECT * FROM VBFA INTO TABLE XVBFA WHERE VBELV EQ DA_VGBEL
    ORDER BY PRIMARY KEY.
ELSEIF SY-SUBRC EQ 8.                    "Append to XVBFA
  SELECT * FROM VBFA APPENDING TABLE XVBFA WHERE VBELV EQ DA_VGBEL
    ORDER BY PRIMARY KEY.
ELSEIF SY-SUBRC EQ 4.                    "Insert into XVBFA
  DA_TABIX = SY-TABIX.
  SELECT * FROM VBFA INTO TABLE DA_XVBFA WHERE VBELV EQ DA_VGBEL
    ORDER BY PRIMARY KEY.
  INSERT LINES OF DA_XVBFA INTO XVBFA INDEX DA_TABIX.
ENDIF.

* Aufbauen verdichteten Fluß XVBAPF
CALL FUNCTION 'GN_XVBAPF_CREATE'
  EXPORTING
    VBELN      = DA_VGBEL
  TABLES
    FXVBAPF   = XVBAPF
    FXVBFA    = XVBFA.

ENDFORM.

```

New:

```
form gn_xvbf_aufbauen using da_vgbel.

data: da_tabix like sy-tabix,
      da_xvbf like xvbf occurs 0.

check: da_vgbel ne belnr_vbfa.
       belnr_vbfa = da_vgbel.

data(lo_docflow_rt) = new cl_sd_document_flow_rt( ).
call method lo_docflow_rt->build_xvbf_from_predecessor
  exporting
    is_document           = value #( id = da_vgbel )
    iv_max_path_length = 2
  importing
    et_xvbf              = data(lt_xvbf).

xvbrk_key-mandt = sy-mandt.
xvbrk_key-vbeln = da_vgbel.
read table xvbf with key mandt = xvbrk_key-mandt vbelv = xvbrk_key-vbeln binary search.
if sy-subrc eq 8 and sy-tabix eq 1.           "Select into empty XVBFA
  xvbf[] = lt_xvbf.
elseif sy-subrc eq 8.                       "Append to XVBFA
  append lines of lt_xvbf to xvbf.
elseif sy-subrc eq 4.                       "Insert into XVBFA
  da_tabix = sy-tabix.
  insert lines of lt_xvbf into xvbf index da_tabix.
endif.

* Aufbauen verdichteten Fluß XVBAPF
call function 'GN_XVBAPF_CREATE'
  exporting
    vbeln   = da_vgbel
  tables
    fxvbapf = xvbpf
    fxvbf   = xvbf.

endform.
```

3.2.5 Select Statement with Non-Direct Predecessors

The old coding does not work without reconstructed VBFA.

The new coding works independently from the release level in all S/4HANA systems.

Old:

```
CONSTANTS: con_stufe TYPE i VALUE '01'.           " Level of the document flow record

* Fetch details of the reference sales document
SELECT SINGLE * FROM vbfa INTO ls_vbfa           "#EC CI_NOFIRST
  WHERE vbeln   = c_ls_vbrp-aubel
  AND vbttyp_n = con_credit_memo_req           "Credit memo request
  AND vbttyp_v = con_order                   "Order
  AND stufe    = con_stufe.                   "Level of the document flow record
```

New:

```
* Fetch details of the reference sales document
data(lo_docflow_rt) = new cl_sd_document_flow_rt( ).
call method lo_docflow_rt->get_predecessors
  exporting
    it_document           = value #( ( type = if_sd_doc_category=>credit_memo_req id = c_ls_vbrp-
aubel ) )
    it_predecessor_type = value #( ( sign = 'I' option = 'EQ' low = if_sd_doc_category=>order ) )
    iv_path_length      = 2
  importing
    et_document          = data(lt_document).
```

If you want all successors or predecessors up to a defined path length, you can set the parameter `IV_SELECT_STRICT_PATH_LENGTH` to `abap_false` in methods `GET_SUCCESORS` / `GET_PREDECESSORS` of class `CL_SD_DOCUMENT_FLOW_RT`.

3.2.6 Select Statement with a Range of SD Document Categories

The new coding can be used in any S/4HANA release, independent from VBFA-STUFE or VBFA reconstruction. In the old coding the usage of literals for the VB Typ values shall be replaced anyway.

Without VBFA reconstruction the result set of the old SELECT is not correct. The old SELECT must be adapted to the new method call.

Old:

```
SELECT vbeln posnn vbtyp_n FROM vbfa " all deliveries and invoices for order
      INTO (l_del_inv_ord-sub_bln, l_del_inv_ord-sub_psnr, l_vbtyp_n)
WHERE
      vbelv = l_current_order-ord_bln      AND
      posnv = l_current_order-ord_psnr     AND

      ( vbtyp_n = 'J' OR vbtyp_n = 'T' OR
        vbtyp_n = 'M' OR vbtyp_n = 'U' OR vbtyp_n = '5' ) AND " deliveries/invoices
      ( vbtyp_v = 'C' OR vbtyp_v = 'H' ).          " type of preceding - order
```

New:

```
* get all related delivery items to order item -> and CDecs then

clear ls_so_items.
SELECT SINGLE vbtyp FROM vbak INTO ls_so_items-type WHERE vbeln = l_current_order-
ord_bln. " get vbtyp
ls_so_items-id      = l_current_order-ord_bln.
ls_so_items-item_id = l_current_order-ord_psnr.
insert ls_so_items into table lt_so_items.

INSERT LINES OF cl_sd_doc_category_util=>rg_any_invoice( ) INTO TABLE rt_range.
INSERT LINES OF cl_sd_doc_category_util=>rg_any_delivery( ) INTO TABLE rt_range.

DATA(lo_docflow_rt) = NEW cl_sd_document_flow_rt( ).

CALL METHOD lo_docflow_rt->get_successors
EXPORTING
      it_item      = lt_so_items
      it_successor_type = rt_range
      iv_path_length = 2
      IV_SELECT_STRICT_ON_PATH_LEN = ABAP_FALSE
IMPORTING
      et_item      = lt_successors.
```

3.2.7 Call of function module RV_ORDER_FLOW_INFORMATION

The type of the tables parameter `VBFA_TAB` in SAP ERP is "LIKE VBFA".

In S/4HANA the type of the tables parameter `VBFA_TAB` is "LIKE VBFAS".

Without field `VBFA-STUFE` the structure `VBFAS` consists of an Include of `VBFA` plus the separate field `STUFE`.

That means in this situation the types `VBFA` and `VBFAS` do not match and therefore the data declaration for the variable `VBFA_TAB` has to be adapted.

After re-introduction of field VBFA-STUFE (see chapter 2.1.5) the structure VBFAS contains an include of VBFA which itself contains STUFE. So VBFAS does not contain field STUFE as a separate field outside the include of VBFA anymore.

After re-introduction of field VBFA-STUFE both versions, the old coding and the new coding, work fine because the types VBFA and VBFAS match.

Old:

```

data: da_vbfa like vbfa occurs 0 with header line.

...

CALL FUNCTION 'RV_ORDER_FLOW_INFORMATION'
  EXPORTING
    belegtyp      = us_xvbak-vbtyp
    comwa         = da_vbco6
    nachfolger    = 'X'
    n_stufen      = '50'
    vorgaenger    = 'X'
    v_stufen      = '50'
    no_acc_doc    = us_det_dyndat
  TABLES
    vbfa_tab      = da_vbfa
  EXCEPTIONS
    no_vbfa       = 1
    no_vbuk_found = 2
    error_message = 3
    others        = 4.

...

```

New:

```

data: da_vbfa like vbfas occurs 0 with header line.

...

CALL FUNCTION 'RV_ORDER_FLOW_INFORMATION'
  EXPORTING
    belegtyp      = us_xvbak-vbtyp
    comwa         = da_vbco6
    nachfolger    = 'X'
    n_stufen      = '50'
    vorgaenger    = 'X'
    v_stufen      = '50'
    no_acc_doc    = us_det_dyndat
  TABLES
    vbfa_tab      = da_vbfa
  EXCEPTIONS
    no_vbfa       = 1
    no_vbuk_found = 2
    error_message = 3
    others        = 4.

...

```

3.2.8 READ TABLE XVBFA WITH KEY <STRUCTURE> BINARY SEARCH

Variants:

```

READ TABLE XVBFA WITH KEY VBFA_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY VBUK_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY VBAK_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY VBAP_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY LIKP_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY LIPS_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY VBRK_KEY BINARY SEARCH
READ TABLE XVBFA WITH KEY VBRP_KEY BINARY SEARCH

```

This adaptation pattern is necessary due to the primary key changes of table VBFA. It does not depend on the existence of VBFA-STUFE or on VBFA reconstruction.

Old:

```

...
data: begin of xvbfakey,
      mandt like sy-mandt,
      vbelv like vbfa-vbelv,
      posnv like vbfa-posnv,
end of xvbfakey.

...

XVBFAKEY-MANDT = SY-MANDT.
XVBFAKEY-VBELV = XLIPOS-VBELN.
XVBFAKEY-POSNV = XLIPOS-POSNR.

READ TABLE XVBFA WITH KEY XVBFAKEY BINARY SEARCH.
...

```

New:

```

...

xvbfakey-mandt = sy-mandt.
xvbfakey-vbelv = xlipos-vbeln.
xvbfakey-posnv = xlipos-posnr.

READ TABLE xvbfa WITH KEY mandt = xvbfakey-mandt vbelv = xvbfakey-
vbelv posnv = xvbfakey-posnv BINARY SEARCH.
...

```

3.2.9 READ TABLE xvbfa FROM ls_xvbfa TRANSPORTING NO FIELDS.

This adaptation pattern is necessary due to the primary key changes of table VBFA. It does not depend on the existence of VBFA-STUFE or on VBFA reconstruction.

Old:

```

...

ls_xvbfa-vbelv = <ls_dastmp>-vbeln.

```

```

ls_xvbfa-vbeln    = <ls_dastmp>-tknum.
ls_xvbfa-vbtyp_v = 'J'.
ls_xvbfa-vbtyp_n = '8'.
READ TABLE xvbfa FROM ls_xvbfa TRANSPORTING NO FIELDS.

```

...

New:

...

```

ls_xvbfa-vbelv    = <ls_dastmp>-vbeln.
ls_xvbfa-vbeln    = <ls_dastmp>-tknum.
ls_xvbfa-vbtyp_v  = if_sd_doc_category=>delivery.
ls_xvbfa-vbtyp_n  = if_sd_doc_category=>shipment.
READ TABLE xvbfa WITH KEY vbelv = ls_xvbfa-vbelv
                    vbeln = ls_xvbfa-vbeln
                    vbtyp_v = ls_xvbfa-vbtyp_v
                    vbtyp_n = ls_xvbfa-vbtyp_n
                    TRANSPORTING NO FIELDS.

```

...