

# Cookbook for Adaptation to Discontinuation of Status Tables VBUK and VBUP

## Table of Contents

- Cookbook for Adaptation to Discontinuation of Status Tables VBUK and VBUP..... 1
- Change Log ..... 2
- Introduction..... 3
  - Simplification of Status Data Model..... 3
  - Adaptation..... 3
- Identifying Affected Objects and Source Code ..... 4
- Coding Adaptation..... 5
  - Case 1: SELECT for dedicated status values that can be replaced through use of data that was already read in an existing call to the respective document table ..... 5
  - Case 2: SELECT for VBUK/VBUP structure that can be replaced through use of data that was already read in an existing call to the respective document table ..... 5
  - Case 3: SELECT that can be switched from VBUK or VBUP to one statically-known document table 6
  - Case 4: SELECT that cannot be switched to one of the document tables..... 6
  - Case 5: VBUK or VBUP are read to check for the existence of a document ..... 7
  - Case 6: VBUK is read to determine VBTYP or VBOBJ for a given VBELN ..... 7
  - Case 10: Insert to VBUK or VBUP ..... 7
  - Case 11: Update to VBUK or VBUP ..... 7
  - Case 12: Delete from VBUK or VBUP..... 7
  - Case 13: DDIC Projection views (and single-table DDIC DB database views) that refer to a status table..... 8
    - Variant A: The object type is statically known ..... 8
    - Variant B: The object type is not statically known ..... 8
  - Case 14: DDIC Database views that refer to a status table ..... 8
    - Variant A: Status table and one dedicated document table are part of the view ..... 8
    - Variant B: View without document table..... 8
- Deleted DDIC views joining VBUK ..... 9
- Changed DDIC views joining VBUK or VBUP..... 9
- CDS Views of VBUP/VBUK ..... 10
  - Objective of the CDS View..... 10
  - Implementation Detail ..... 10

Impact on Customer Code.....	10
Adaptation of append structures of VBUK and VBUP .....	10
Appendix 1: Mapping of VBUK/VBUP fields to document tables.....	13
Fields from table VBUK.....	13
Fields from table VBUP.....	14
Appendix 2: Mapping of IS-OIL Append fields in VBUK and VBUP to document tables .....	15

## Change Log

Version	Date	Change
2.2	2015-10-27	New chapter "Adaptation of append structures of VBUK and VBUP"
2.3	2015-12-07	Added details regarding appends; small corrections
2.4	2016-04-21	Updated customer reserve fields in the Venn diagrams of VBUK and VBUP tables (Appendix 1)
2.5	2016-07-11	Added new section "CDS Views of VBUP/VBUK"
2.6	2017-08-11	Enhanced chapter "Adaptation of append structures of VBUK and VBUP" Update on chapter "Identifying Affected Objects and Source Code" Added Appendix 2: Mapping of IS-OIL Append fields

## Introduction

### Simplification of Status Data Model

In SAPERP, the status values of SD transactional documents are not held in the document tables themselves, but in tables VBUK (for header) and VBUP (for item).

This change is part of the S/4HANA data model simplification. The status values of SD transactional documents are now stored in the document tables themselves. As a result, the status information moved from VBUK to VBAK, from VBUP to VBAP for sales order, to LIKP and LIPS for delivery, and to VBRK for billing documents. Note that billing documents have no item status values. Appendix 1 shows the status fields and their relevance for the five tables.

The obsolete tables have not been deleted; they stay active but empty, and you can still use them for typing.

### Adaptation

You must adapt read and write accesses to tables VBUK and VBUP to these changes. Append fields added to VBUK and VBUP have to be created in other tables.

You might also want to adapt views, LDBs (logical databases), search helps, domains, and TABUs.

### *Generic Table Accesses for Value Helps and Veto Checks*

A special case of read access uses VBUK or VBUP as a check table in structure fields or table fields, especially in screens and reports that leverage the check table. Check these kinds of uses to avoid loss of functionality.

Theoretically, you can use the check table to hold a generic value help for a screen field or report select option. These value helps depend on the content of the table, and will fail if the tables have no data. However, these types of problems occur rarely because dedicated DDIC search helps are typically used for screen fields.

More commonly in screens, check tables perform veto checks of fields VBELN and POSNR. If the following settings are established, the veto check is active:

- Check table VBUK or VBUP is listed at the structure field or table field
- Flag "check required" is set at the foreign key in this field
- Foreign key flag is set in the screen field referring to this field

If the screen field is used semantically for only one of the document types, it may be appropriate to replace check table VBUK with a more specific check table according to the semantic of the fields (for example: VBAK). Alternatively, you could code and call the veto check in the screen flow logic, or abandon it if acceptable for the scenario.

## Identifying Affected Objects and Source Code

For detailed information on how to identify which of your custom code objects might need to be adapted in order to be compatible with SAP S/4HANA, please refer to

- SAP note 2190420
- <https://blogs.sap.com/2017/02/15/sap-s4hana-system-conversion-custom-code-adaptation-process/>

This cookbook describes how to handle those findings of the SAP S/4HANA custom code check in ABAP Test Cockpit, which are labeled as being related to SAP note 2198647.

## Coding Adaptation

Typically, customer coding adapted to work on S/4HANA no longer needs to be compatible with SAPERP. It can be adapted without the constraint of keeping the consumer compatible with the SAPERP data model.

### Case 1: SELECT for dedicated status values that can be replaced through use of data that was already read in an existing call to the respective document table

Simply use the data that was already read.

Example:

```
Old:  select single * from vbak where vbeln = iv_vbeln into ls_vbak.  
      select single GBSTK from vbuk where vbeln = ls_vbak-vbeln into lv_gbstk.  
      if lv_gbstk = 'C'.  
      ...  
New:  select single * from vbak where vbeln = iv_vbeln into ls_vbak.  
      if ls_vbak-gbstk = 'C'.  
      ...
```

### Case 2: SELECT for VBUK/VBUP structure that can be replaced through use of data that was already read in an existing call to the respective document table

Simply use the data that was already read. Also, fill field VBOBJ if needed.

Example:

```
Old:  select single * from vbuk into ls_vbuk where vbeln = ls_vbak-vbeln.  
New:  move-corresponding (structure or even internal table) from LS_VBAK to LS_VBUK.
```

VBOBJ:

If the VBUK field VBOBJ is evaluated, or the usage extent of the filled vbuk-like structure or internal table is unknown, you must fill field VBOBJ manually because it was previously part of vbuk, but has not been transferred to the document header tables.

In the example above, add line

```
“ ls_vbuk-vbobj = IF_SD_DOC_OBJECT=> SALES_DOCUMENT. “
```

If you fill a table, you need a loop to fill the fields in all records.

You should use the constants SALES\_DOCUMENT, DELIVERY, and INVOICE from interface IF\_SD\_DOC\_OBJECT to do this.

Move-corresponding for internal tables:

If you want to use this pattern to replace one SELECT statement from VBUK for different objects, as was done in Report SAPDBKMOV for sales orders and deliveries, you can add 'KEEPING TARGET LINES' to allow subsequent moves into the same internal table:

```
MOVE-CORRESPONDING xvbak[] TO xvbuk[] .  
MOVE-CORRESPONDING xlikp[] TO xvbuk[] KEEPING TARGET LINES .
```

### Case 3: SELECT that can be switched from VBUK or VBUP to one statically-known document table

(If Case 1 and Case 2 cannot be applied.)

If the VBELN in the where clause has been filled from a specific document table, or if status fields that exist in only one of the document tables are read or used, replace VBUK/P with the respective table name.

(For the assignment of status values to tables, see the diagrams in Appendix 1.)

If no column list is specified, but a "\*" is used to select all columns, then:

- A change from "select ... into table" to "Select ... into corresponding fields of table" is necessary (similarly: from "appending table ..." to "appending corresponding fields of table ...").
- The remark regarding field VBOBJ from Case 2 should be considered here.

### Case 4: SELECT that cannot be switched to one of the document tables

If it's obvious that VBUK or VBUP are read for different document types (from documents stored in vbak, likp and vbrk or their item tables) or if the document type is unknown, use one of the following FMs, which can read all header tables or all item tables:

- SD\_VBUK\_READ\_FROM\_DOC
- SD\_VBUK\_READ\_FROM\_DOC\_MULTI (mass-enabled for multiple documents)
- SD\_VBUP\_READ\_FROM\_DOC (can return one dedicated item of a dedicated document or all items of a document)
- SD\_VBUP\_READ\_FROM\_DOC\_MULTI (mass-enabled for multiple documents)
  - returns all items for a list of VBELNs if only IT\_VBUK\_KEY is filled
  - returns all items requested in a list of VBELN-POSNR pairs if only IT\_VBUP\_KEY is filled

The mass-enabled FMs also support VBELNs from different object types (tables, that is) in one call.

If the VB Typ or the VBOBJ of the document is available, it can be passed to the FMs by using the optional import parameter I\_VBTYP or I\_VBOBJ. This will direct the access to the one specific table.

Since S/4HANA OP 1511 SP1, these FMs also consider append fields that are part of the specific status included in the document table.

## Case 5: VBUK or VBUP are read to check for the existence of a document

```
SELECT SINGLE * FROM VBUK WHERE VBELN = XVBRP-VBELN.  
IF SY-SUBRC NE 0.  
[...structure vbuk not evaluated later...]
```

Decide whether you can replace this with a call to one document table, or with subsequent calls to several tables. You can also use FM SD\_VBUK\_READ\_FROM\_DOC or SD\_VBUP\_READ\_FROM\_DOC.

## Case 6: VBUK is read to determine VB Typ or VBOBJ for a given VBELN

Note that field VBOBJ has not been transferred to the document tables.

If VB Typ is known, VBOBJ can be derived without database access using the static methods IS\_ANY\_SALES, IS\_ANY\_DELIVERY, and IS\_ANY\_INVOICE of class CL\_SD\_DOC\_CATEGORY\_UTIL.

If only VBELN is known, decide whether you can replace the VBUK select can be replace with a SELECT to one document table or with several calls to different document tables. A hit in one table logically determines the VBOBJ. VB Typ is part of the document header tables.

You can also use FM SD\_VBUK\_READ\_FROM\_DOC, because it fills the field VBOBJ and, of course, also returns VB Typ.

## Case 10: Insert to VBUK or VBUP

For standard SD documents, all inserts to VBUK or VBUP should be obsolete. These inserts have to be removed. Instead, the inserts to the specific document table that now holds the status values have to be extended.

However, some industries may still use and fill VBUK or VBUP for their own document types, especially if they are components of a single code line.

## Case 11: Update to VBUK or VBUP

You must replace updates to Insert VBUK or VBUP have with updates for the specific document tables. If the correct document table is unknown, the two (for item) or three (for header) document tables have to be tried, one after another, until a hit is found.

However, some industries may still use VBUK or VBUP for their own document types, especially if they are components of a single code line.

## Case 12: Delete from VBUK or VBUP

Even before the data model is simplified, you should only delete VBUK/VBUP entries together with their respective document entries. Therefore, code for VBUK or VBUP deletion can be removed if it is obvious that the related document entries have been deleted.

Only deleting VBUK or VBUP entries is typically a status repair program (for example: one that is described in a note). Such programs have to be commented out, because inconsistencies between document tables and status tables are no longer possible, and adaptation doesn't make sense here.

However, some industries may still use VBUK or VBUP for their own document types, especially if they are components of a single code line, and might (for whatever reason) want to delete VBUK or VBUP entries that belong to their own document types.

### **Case 13: DDIC Projection views (and single-table DDIC DB database views) that refer to a status table**

#### *Variant A: The object type is statically known*

Adapting all consumers that read this view is the simplest adaptation. You can change them to read the document table of this object type directly, instead of reading the view. Alternatively, you can adapt the view to refer to the document table instead. Note that field VBOBJ cannot be filled within a DDIC view. Consumer adaptation is required wherever this field is evaluated.

If a view has a large number of consumers, and VBOBJ is not being evaluated, you might want to adapt only the view itself, to save yourself the effort of consumer adaptation.

#### *Variant B: The object type is not statically known*

Because the data could be stored in one of several tables, this scenario cannot be solved with a view based on a single table. Furthermore, a DDIC database view across all possible tables will not work because it only supports inner joins. (Note: You could create a CDS view by using union or outer join.)

The consumers reading this view have to be adapted. One option is to use the FMs listed in case 4 instead of the view. Alternatively, each consumer SELECT in the obsolete view can be replaced with multiple subsequent SELECTs on the possible tables (or on views referring to these tables).

### **Case 14: DDIC Database views that refer to a status table**

#### *Variant A: Status table and one dedicated document table are part of the view*

You can simplify this kind of view. Status fields and existing select conditions in the view have to be adapted to refer to the document table. The status table can then be removed from the view.

If the view uses only these two tables (for example, a join of VBAK and VBUK, joined via VBELN), it may be useful to delete the view and adapt all consumers to access the document table directly. This applies if there is no explicit desire to limit the document table's fields (for example: to reduce the amount of transferred data) or to implement central selection conditions (for example: for reuse).

Again, you should take the number of consumers into account.

#### *Variant B: View without document table*

From an adaptation perspective, this corresponds to Case 13 with additional joins, and must be handled accordingly.

## Deleted DDIC views joining VBUK

The following DDIC views have been deleted in S/4HANA:

View	Joined Tables
/SPE/LIKPUK	LIKP, VBUK
V_CMPB_ASGN	CMPB_ASGN, CMPB_MAP, VBUK
SD_BACKORDER	VAKPA, VBAK, VBUK
VAKPA_REF	VAKPA, VBAK, VBUK

You should replace read accesses to these views with SELECT statements on a single table, or if needed, with appropriate joins.

## Changed DDIC views joining VBUK or VBUP

The following DDIC views have been adapted in S/4HANA:

View
VBAKUK
VBRKUK
LIPSUP
VBSSUK
V_CM01_OPT
V_CM02_OPT
V_CM03_OPT
V_CM04_OPT
V_CM05_OPT
V_CM06_OPT
V_CM07_OPT
V_CM08_OPT
V_CM09_OPT
V_CM10_OPT
V_OLR3_VBAX
V_OLR3_VBAX_INQ
V_OLR3_VBAX_QUO

In these views, view field VBOBJ (referring to VBUK-VBOBJ) was removed in most cases. If there is a consumer that reads and evaluates view field VBOBJ, see the remark regarding field VBOBJ in section “Case 2: SELECT for VBUK/VBUP structure that can be replaced through use of data that was already read in an existing call to the respective document table”.

For some views, status fields that were not relevant for the respective object were also removed.

## CDS Views of VBUP/VBUK

### Objective of the CDS View

A SELECT statement on the database table VBUP and VBUK is no longer available since it does not contain any data. Instead, you can access the status information by performing the SELECT on the CDS views named V\_VBUP\_CDS and V\_VBUK\_CDS respectively. The CDS views are valid for all variants of SELECT statements.

	CDS View of VBUK	CDS View of VBUP
<b>Package</b>	VZ_XPRA	VZ_XPRA
<b>DDL Name</b>	V_VBUK_S4	V_VBUP_S4
<b>SQL Name</b>	V_VBUP_CDS	V_VBUK_CDS

### Implementation Detail

The CDS view of table VBUK contains fields identical to the table VBUK. A union statement implements the CDS view to merge the corresponding fields from tables VBAK, LIKP and VBRK. If a field only exists in one of these three tables, its value is extracted from its original table. The field will be filled with empty values for the two other tables in the union statement. For example, field ABSTK only exists in table VBAK. In the union statement, it will be selected as *“as ABSTK”* in the two other tables. For the field VBOBJ, pre-set value ‘A’ for records from VBAK, ‘F’ for VBRK, ‘L’ for LIKP.

Similarly, the CDS view of table VBUP contains fields identical to the table VBUP, except for the append fields. A union statement implements the CDS view to merge the corresponding fields from tables VBAP and LIPS.

### Impact on Customer Code

The CDS view of tables VBUK and VBUP are provided to customers as a way to perform SELECT statement on database tables VBUK and VBUP.

Performing the SELECT statement on the CDS view should yield the same result as when performing on tables VBUK or VBUP, except for the append fields. You need to change the table names in the SELECT statement to V\_VBUK\_CDS and V\_VBUP\_CDS respectively. Examples are given below.

<b>SELECT on database table</b>	<b>SELECT on CDS views</b>
<i>SELECT * FROM VBUP</i>	<i>SELECT * FROM V_VBUP_CDS</i>
<i>SELECT ABSTK FROM VBUK</i>	<i>SELECT ABSTK FROM V_VBUK_CDS</i>

Note that when implementing a SELECT statement via V\_VBUK\_CDS, you must specify the object type VBOBJ; only then will a given document number lead to a unique result. Otherwise, the caller must be aware that several hits can occur for one document number, and must handle the situation accordingly.

Note that the CDS view is one possible approach provided by SAP to give our customers the experience of seamless data migration. However, it is not the recommended method to select data, due to the performance and robustness. SAP recommends that you use the other coding adaptations as mentioned above in the cookbook.

### Adaptation of append structures of VBUK and VBUP

If you have added one or several append structures to database tables VBUK or VBUP (document status header and item table) in SAP ERP to store additional data for documents in tables

VBAK/VBAP, LIKP/LIPS, or VBRK, and these fields will be used in S/4HANA, you must add the appended fields to the respective document tables.

You must add an append field to document header status table VBUK to one or several of the document header tables VBAK, LIKP, VBRK, depending on the fact for which of the document types the respective field is relevant.

You must add an append field to document item status table VBUP to one or several of the document item tables VBAP, LIPS depending on the fact for which of the document types the respective field is relevant.

Each append field must have the same name and the same datatype as in the source table, and has to be appended to the table specific status include (VBAK\_STATUS, LIKP\_STATUS, VBRK\_STATUS, VBAP\_STATUS or LIPS\_STATUS). Only then will the field be considered by automatic data conversion from VBUK to VBAK/LIKP/VBRK and from VBUP to VBAP/LIPS, and also by the read function modules listed in case 4 above.

Do not choose another name or datatype. Do not append the field directly to the document table. It is possible to have several append structures for one status include.

*When do you have to add the append fields to the status include structures in order to have the data transferred automatically during the SAP S/4HANA system conversion?*

In the development system, you have to add the fields during phase MAIN\_SHDRUN/ACT\_UPG, when you are prompted for SPDD. In all other systems, you may use modification adjustment transports or customer transport integration during the system conversion.

The append fields must be contained in one of the custom transport requests from the development system.

Until target release 1610 the conversion pre-check class CLS4H\_CHECKS\_SD\_DATA\_MODEL delivered via SAP Note 2224436 provides a check for such append fields and issues a warning if such append fields are recognized in the source release. The warning refers to the necessary action to add the append fields to the status include structures.

As of SUM SP20 (as of target release 1709) SUM checks for append fields in VBUK/VBUP and asks for adjustment of the target tables in phase MAIN\_SHDRUN/ACT\_UPG (compare screenshot below).

33%

Process is in dialog state

Current Phase: MAIN\_SHDRUN/ACT\_UPG

### Repository Modifications

You must now include your repository modifications in the new SAP import.

The program has found 12 modified objects that you need to adjust and 0 objects are automatically maintained by import of an adjustment transport that you have included into this procedure.

If you do not have included adjustment transports, please check log file SPDD\_RESET\_CANDIDATES.CJA for potential candidates to reset to SAP original. You may execute report RSUPG\_SPDD\_RESET\_CANDIDATES as an ordinary user (not DDIC) in the shadow system to automatically perform this reset to SAP original for all identified candidates.

**CAUTION:** If you do not make adjustments now in transaction SPDD, your modifications will be lost. Even if you do not want to keep your modifications, use transaction SPDD to reset these objects to the original version.

The S/4HANA conversion has found tables with customer appends which require customer action due to data model changes in S/4 HANA.

To adjust your repository, proceed as follows:

1. Logon to the shadow instance as user DDIC in client 000 on server plx098 instance 02.
2. Set the 'System Change' option with transaction SE06.
3. Create users for modification adjustment by copying user DDIC with transaction SU01.
4. Logon to the shadow instance as ordinary user and make adjustments with transaction SPDD and additionally follow the instructions in the SAP Notes listed below. For more information, see the SUM Guide.

The following tables with customer appends are affected.

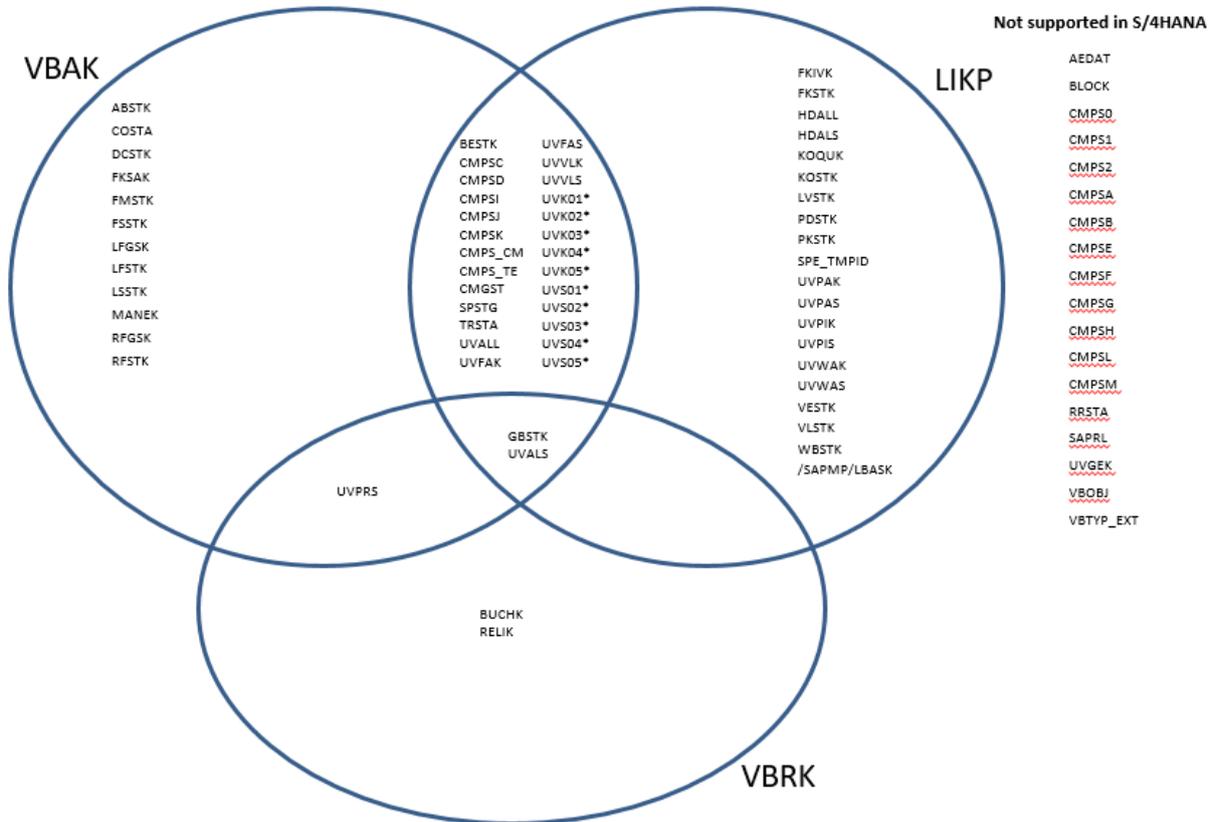
Tables and affected appends	SAP Note
<ul style="list-style-type: none"><li>Table "VBUK" ZCGCUST1</li></ul>	2198647
<ul style="list-style-type: none"><li>Table "KONV" ZKONV_CUST_FIELDS</li></ul>	2220005

- Adjustments made, continue with procedure

## Appendix 1: Mapping of VBUK/VBUP fields to document tables

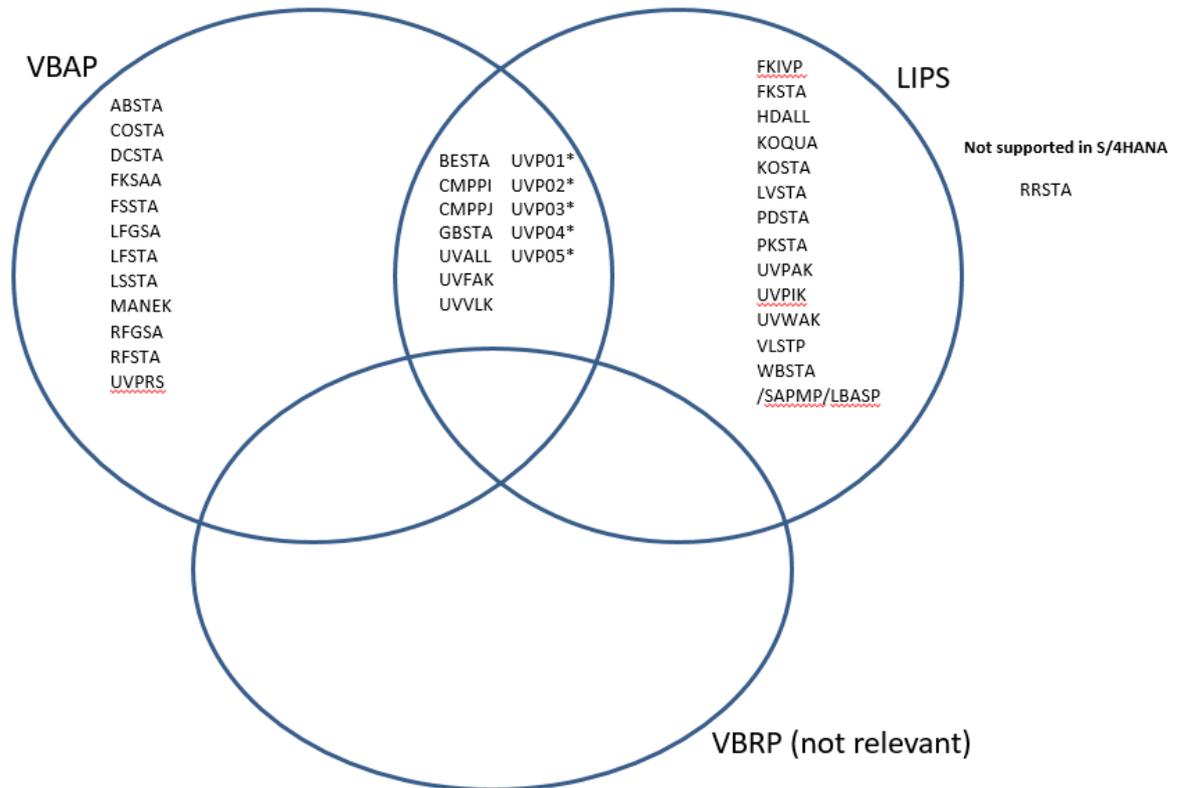
These diagrams show which status fields from tables VBUK and VBUP have been transferred to which document tables:

### Fields from table VBUK



\* These customer reserve fields are missing in SAP S/4 HANA 1511 SP00 and SP01. The fields are available again as of 1511 FPS2.

## Fields from table VBUP



\* These customer reserve fields are missing in SAP S/4 HANA 1511 SP00 and SP01. The fields are available again as of 1511 FPS2.

## Appendix 2: Mapping of IS-OIL Append fields in VBUK and VBUP to document tables

VBUK:

Fieldname	VBAK	LIKP	VBRK
OIG_SSTSF	n/a	OIG_SSTSF	n/a
OIC_NXTINV	n/a	OIC_NXTINV	n/a

VBUP:

Fieldname	VBAP	LIPS
OIFINAL	OIFINAL	OIFINAL
OIINVCST1	n/a	OIINVCST1
OIINVCST2	n/a	OIINVCST2
OIINVCST3	n/a	OIINVCST3
OIINVCST4	n/a	OIINVCST4
OIINVCST5	n/a	OIINVCST5
OIINVCST6	n/a	OIINVCST6
OIINVCST7	n/a	OIINVCST7
OIINVCST8	n/a	OIINVCST8
OIINVCST9	n/a	OIINVCST9