

# SAP Table and Field search techniques

## Dennis Barrett, SAP America

**The problem:** You want to create an ABAP query or ABAP report using fields that you can find on transaction screens, but you don't know where the data are stored. This paper describes some techniques you can use to find the data. Some fields may still resist these techniques; there are certainly more ways yet to chase them down. Thanks to the many SAP consultants and others who contributed to these strategies.

1. The first step is always to find out the table and field name of the screen field. With the cursor in the field you want, press F1 or click on the "?". The initial help screen will give you descriptions and information about the field. From there press F9 or click on *[Technical info* . Here you can get names of the table, field and the data element.
2. Double-click on the field labeled *Field data /Table name*.

Screen title	Object label	Name description	Result
Dictionary: Table/Structure: Display fields	Name	Structure	Keep looking
Dictionary: Table/Structure: Display fields	Name	Transparent table	Success
Dictionary: Display View	Database view	(none)	Success

Many screen fields are presented using structures which are Data Dictionary objects that have fields but do not carry data. They are like empty tables. If the field is in a structure, you'll have to keep looking.

Transparent tables store data, and views are linked groups of transparent tables. You'll find the field's name somewhere in the list. If the field is in a view, you can look at the definition of the view to find out the underlying table that contains the field.

3. Back out to the "Technical Information" dialog box and double-click on the field labeled "Field data/Data element". This will show you the value table if there is one. If you're just looking for master data like customer name - KUNNR - then the value table will give you what you want. If you need the data from a transaction, then the master data won't do and you get to keep looking.
4. When R/3 moves data from one table to another (or into a structure), it often (not always) moves it into a target field having the same Data element as the source field. From the Data element display, follow the path *{ Utilities { Where-used list { Table fields { Enter*. This will list the tables, views and structures that use this data element. If your field is a commonly used one, this list may be so large that it's hard to find your data storage location. For example KUNNR shows up in 743 objects in Release 4.0B. An infrequently-used field may show up in fewer objects, and you can find your data source in the list. Note also that the list includes (sometimes lots of) structures. When you drill into a fieldname in the list, you'll see the table display which will identify it as a structure or a transparent table. If it's a structure, you'll have to keep looking. If the Data element doesn't work, you can try the same thing with the Domain.

## SAP Table and Field search techniques

### Dennis Barrett, SAP America

5. Since the data element *Where-used* list generally contains so many (non-data carrying) structures, the list is not terribly useful for this purpose. You can create the data dictionary view "Z\_DATA\_ELEMENT" described below which will show only the data tables that use your data element. Use /SE11 or {Tools {ABAP/4 Workbench {Dictionary <Z\_DATA\_ELEMENT {Utility {Table contents and enter name of the data element you are searching for in its selection field. This view will give you a list of only data tables that use that data element.

Z_DATA_ELEMENT		Base tables			
Table					
DD03L					
DD02L					
DD02T					
DD04T					

  

Z_DATA_ELEMENT		Joins			
Table	Field	=	Table	Field	
DD03L	TABNAME	=	DD02L	TABNAME	
DD03L	TABNAME	=	DD02T	TABNAME	
DD03L	ROLLNAME	=	DD04T	ROLLNAME	

  

Z_DATA_ELEMENT		Fields	
Base table	Base field	View field	
DD02L	TABNAME	TABNAME	
DD03L	FIELDNAME	FIELDNAME	
DD02L	TABCLASS	TABCLASS	
DD03L	POSITION	FLD_POSNR	
DD03L	KEYFLAG	KEYFLAG	
DD03L	MANDATORY	MANDATORY	
DD03L	ROLLNAME	DATA_ELEM	
DD03L	CHECKTABLE	CHECKTABLE	
DD03L	NOTNULL	NOTNULL	
DD02T	DDTEXT	TBL_DESCR	
DD04T	DDTEXT	D_E_DESCR	

  

Z_DATA_ELEMENT		Selection cond.			
NOT	Table	Field name	Op	Comparison value	AND/OR
	DD04T	DDLLANGUAGE	EQ	'E'	AND
	DD02T	DDLLANGUAGE	EQ	'E'	AND
	DD03L	AS4LOCAL	EQ	'A'	AND
	DD02L	AS4LOCAL	EQ	'A'	AND
	DD02L	TABCLASS	EQ	'TRANSP'	OR
	DD02L	TABCLASS	EQ	'CLUSTER'	OR
	DD02L	TABCLASS	EQ	'POOL'	

6. Go to SE16 or follow the path {Tools {ABAP/4 Workbench {Overview {Data Browser. Press F4, then SAP Applications. Drill down through the menu tree to find the tables associated with your application. Your field may well be in one of those tables.

7. Alternatively, use 'RunTime Analysis'. Go to transaction SE30 or follow {System {Utilities {Runtime Analysis {Execute. Enter the transaction code, then [Execute the transaction. After you post the transaction, the system will return you to the RunTime Analysis screen. Choose [Analyze then [Tables to see all the transparent and pool tables that were accessed during the transaction. [Hit list will show you all the program calls and specify the ABAP program that was running.

## SAP Table and Field search strategies

8. Sometimes the *Where-used* list for the original screen structure with *\$Programs* checked will show the place where the field is associated with the database. Compare the programs in this list to those you found in the RunTime Analysis Hit List to narrow your search.
- 8a Here's another technique that two people sent after seeing the article: Satyanarayana Hadeed in Saudi Arabia and Roberto Vaccaro in Brazil. Thanks, people:
  1. Data element field - Utilities - Where used List
  2. Sub screen as shown in Figure 8 of your article is displayed.
  3. Select "Indirect application" or function "F6".
  4. Here we get another sub screen, where one can directly go to list of Tables by double clicking "Table". The list contains only database Tables (e.g. no structures or views) in which the Data element is used.
  5. A big advantage of this approach is that you can drill into the table and see its entire structure.
9. One or more of the other screen fields may be directly connected to their database tables. If you can discover the "Header" table of the transaction, it may include the field you want. For example, assume you want the "Sold to" data from the Sales order initial screen. It's listed as RV45S-KUNNR, but you find that RV45Z is a structure. Its data element is KUNAG, but Release 4.0B uses that data element in 276 tables, so it would be tough to find it there. On the same screen, the "Order" is listed as VBAK-VBELN. When you look up the VBAK table, you find that it does include KUNNR.
10. Transaction data is normally stored in several hierarchically-connected tables, and R/3 may have one or more views defined for the transaction you're dealing with. You can search through those views to find other associated tables. Any one of those might store your data. For example, assume you want the "Order quantity" data from the Sales order: Single-Line Overview. It's listed as RV45A-KWMENG, but you find that RV45A is a structure. You already know that "Order" is VBAK-VBELN, so you look for views in the Where-used list for VBAK. View VIVEDA connects VBAK, VBUK and VBAP and you find that VBAP includes the field KWMENG.
11. Tables that are related in a business process are sometimes connected in a Logical Database. The VBAK Where-used list for Logical databases, for example, can steer you to lots of other candidate tables.
12. Many transactions have header information and detail or item lines. Remember that R/3 is a German product. "Kopf" is German for "head", and "position" can be interpreted in German as "detail" or item. You'll often find header/item table pairs with "K" and "P" in their names, such as VBAK & VBAP.
13. Sometimes text fields are stored in structure RSTXT or the like. Texts are stored all over the place in R/3, but the long texts frequently are stored in tables STXH and STXL, text header and text line respectively. The texts are stored in raw (i.e. non-ASCII) form, so you must use function modules to get them. There are several function modules for processing these texts, including Read\_Text which reads text lines from the database. You'll need the Text Name, ID and Object parameters to execute the function modules. Those parameters are available from the text field by following *{Long Text {GoTo {Header.*

## SAP Table and Field search strategies

14. R/3 records changes that are made to many documents, and frequently the change information includes the name of the table and field affected. You can change the field you're looking for in such a document, and then read the change record to find the table and field name. Some examples of menu paths to change records in release 4.0B are:

In Plant Maintenance or Service Management

Technical Objects: {Extras} {Display Changes

Order: {Extras} {Order documents} {Changes

Notification: {Extras} {Notification documents} {Changes

Contracts & planning: {Environment} {Changes

Technical Objects: {Extras} {Display Changes

In Sales and Distribution

Order: {Environment} {Changes

Delivery: {Environment} {Changes

Tables TCDOB and TCDOBT list all the available object classes for change document creation. You can get clues there about whether your application supports change documents.

15. If you can't find the change document in your transaction, have a look in the change document tables CDHDR and CDPOS (Now there's a mix of English and German naming conventions: CD Header and CD Position!) as follows. After you have changed and saved the field you're searching for, open CDHDR in the ABAP Workbench dictionary, the follow {Utilities} {Table contents} to find the change record. In the selection screen enter today's date, time range and your user name. It should pull the one matching record. Ctrl-C copy the value of CHANGENR, and use it to pull the appropriate record from CDPOS. This record contains the names of the table and field affected as well as the new and old values so you can assure that you're looking at the right record.

16. If your field is in a structure, that structure may contain one or more .INCLUDEs which themselves are actually tables. For example in a Sales Order Item Overview, the Condition Type is listed as KOMV-KSCHL but KOMV is a structure. KOMV does contain the .INCLUDE KONV which is itself a table that contains the field KSCHL.

17. In /SE11 or {Tools} {ABAP/4 Workbench} {Dictionary} <Tablename} {Display} {Extras} {Graphics} you'll see a graphical representation of the selected table and its links to other tables. You can use this to see the entire family of tables related to a business object, and may be able find where your field is located.

18. If all else fails, it's time for some serious archeology. In one session go into transaction /ST05 or {Tools} {ABAP/4 Workbench} {Test} {SQL Trace}. In another session, go to your transaction but don't run it yet. Switch over and click on the [Trace On] button and [Ok]. Switch back and run your transaction. Switch over and click [Trace off], then [List trace] and [Ok]. You'll have a list of all the SQL calls to the tables with their parameters. It may be easier to download it into a word processor document for searching {System} {List} {Save} {Local file} {unconverted} <Filename = "c:\mypath\filename.doc". Search for your field name and notice the tables it is associated with. It may involve detective work to determine which table actually stores the data. This approach requires an understanding of SQL commands and principles.

19. Another heavy-duty ABAP approach is to debug the program. Go to SE38 or follow {Tools} {ABAP/4 Workbench} {ABAP/4 Editor}, enter the program name and click [Debugging]. Press [Enter] to get to the debugging screen, put a watchpoint on the variable, and then click on [Continue]. Any change in the variable triggers the watchpoint, presenting you with the code immediately below the line that changed the variable. You may be able see references to the data table where the information is stored.

## SAP Table and Field search strategies

20. Finally, extract some data from the set of tables associated with your transaction, and copy those records into a text file. Use the "Search" function in your word processor to find data flows by content where the source and destination fields may have different names and data elements. R/3 often uses internal numbers to link records that you'd never think of, such as Object ID and Object Type. Their values however can be quite distinctive, and you may be able to find the path of data flow by searching for the values (ignoring the field names) when you have pulled data from several transactions.

For example, to find Measurement documents associated with a Service order, we needed to trace the following connections. There was nothing obvious about which fields linked the data.

Tables	Joins	Conditions
AUFK or AFIH AFKO AFFH CRVE_A EQUI IMPTT IMRG	AUFNR AUFNR AUFPL AUFPL OBJID OBJID EQU NR EQU NR OBJNR MPOBJ POINT POINT	OBJTY= 'FH'

We created a Service Order and several Measurement documents in R/3. Then we pulled out the records starting with the Service order numbers, and chased through the various joins to pull all the associated records. While displaying each table in */SE11* we used *{Utilities {Table contents* to extract the data. We exported the records as text documents using */%pc* or *{System {List {Save {Local file* and merged them all into a single Word document. We then used Word's *Find* function to follow the data through the tables.

---

Note: Navigation instructions in this paper use the York-Mills Notation described in "York-Mills Notation: SAP Interactive Command Recording System" by Dennis Barrett and Tipton Cole, 1996-1998. That paper is available at <http://www.mindspring.com/~dennis.barrett/ymn/ymn.html>.