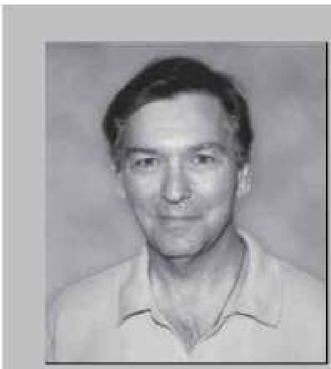
18 Techniques for Locating the Underlying Data of a Screen Field

18种根据屏幕字段 查找潜在数据的技巧

Dennis Barrett / 翻译：强晟 qiangsheng@gmail.com



Dennis Barrett is an applications consultant with SAP America who focuses on Service Management/ Customer Service and the Service Provider solution.
He has been consulting in computers for over 15 years, always blending applications and programming. He is a certified ABAP programmer, and is also the author of "SAP R/3 ABAP/4 Command Reference."

丹尼斯・巴雷特是SAP美国的应用顾问，致力于服务管理/客户服务和服务提供解决方案。
他已经在计算机行业从事顾问工作15年，是认证ABAP程序员，也是《SAP R/3 APAP/4 命令参考》一书的作者。

(complete bio appears on page 18)
（关于他的完整简历请见25页）

Picture this. An Operations Manager wants a special report of her service orders. She gives you a sketch of what she wants it to look like (columns, rows, headings, grouping, subtotals, totals, and the like), and points to fields on the Service Management transaction screens that have the data she wants you to show in the report.

想象一下这样的情景。一个业务经理希望得到一个关于她的服务订单的专用报表，她给了你一个报表的草图（包括列、行、标题、分组、小计、合计等等），而“服务管理”事务屏幕中的字段就包含了她希望你在报表中显示的数据。

What do you do? Most likely, you would check to find any existing reports delivered with the system (or already written for this client) that provide the information the Operations Manager wants, or that can be copied and revised to fit her needs. If you don't find any, you might then look into the appropriate reporting system — in this case, the Plant Maintenance Information System (PMIS) — to see if you can adapt it. In this case, however, you can't find the report, and you can't cobble one together from existing ones. You must write a report or an ABAP Query to fulfill the requirements.

你该怎么做？通常的，你会检查系统中运行的，能够提供这个业务经理所需信息的现存报表（或者是已经为这个client写过的），或者可以复制和修改后能符合需要的报表。如果没有找到，你也许会在适当的报表系统——例如，工厂维护信息系统（PMIS）——中看看是否能找到合适的。即使这样，你仍然无法找到报表，或者无法从现存报表中找到可修改的，那只能写一个报表或者一个ABAP查询来实现这个需求。

Now, suppose the Operations Manager asks if you can somehow add just one more little function to her Create Measurement Documents transaction: paste the associated sales order number into the MDoc text field. This scenario requires you to create an enhancement. In both the first and second scenario, you'll need access to specific data from the database tables. How do you find that data with just the screen fields as your guide? We all know the information associated with a transaction is stored in several (sometimes many) related tables. So, when you need several fields from a transaction for a report or an enhancement, you may need to find many of the transaction's tables and establish the links among them. Those links are often not obvious.

现在，设想一下业务经理问你是否以某种方式在《创建计量单位文档事务》中增加一个小小的功能：向MDoc文本字段中粘贴相关销售订单号。这个要求需要你创建一个增强功能。在前面两个情形中，你都需要访问数据库表中的特定数据。你怎么以屏幕字段做指导来找到这些数据？我们都知道与一个事务关联的信息都存储在几个（偶尔还会是很多个）相关的表里。这样，当你需要在报表或增强功能中使用一个事务中的几个字段时，你需要找出许多事务的表并在它们之间建立关联。这些关联通常都不会很明显。

I've come across these very situations on numerous occasions while working for several clients with the SD, MM, IM, WM, and Service Management (now called Customer Service) modules. Over time I have gathered suggestions from colleagues and developed some techniques myself to find the data I needed. I will share these techniques — 18 of them in all — with you now.

我曾经在众多场合遇到这些情况，有时需要同时在SD、MM、IM、WM和服务管理（现在叫客户服务）等模块对多个client进行工作。随着时间推移，我收集了同事和自己在开发中寻找需要数据的建议。我现在来跟你分享这些技巧——共18种。

The Starting Gate and Finish Line

起跑线和终点线

I will be presenting a wide variety of techniques for reaching the "finish line" of this exercise — namely, locating the table and field that stores the underlying data of a screen field. You're unlikely to need all of them; you'll probably be successful within the first four techniques. I include the remaining techniques because some fields can be very hard to chase down, and any one of these other tools just might do the trick.

我将会介绍达成本练习"终点线"的各种各样的技巧——也就是说，定位那些在屏幕字段中存储潜在数据的表和字段。你未必需要所有这些方法，一般来说，前四种技巧已经足够了。我之所以写其他的技巧是因为有些字段非常难以追踪，而这些工具或许可以解决问题。

The Starting Point

起点

The starting point for this set of techniques is always the Technical Information screen, because it is here that you learn the screen field's name, and can discern whether or not the underlying data object is a structure, view, or table.

这套技巧的起点通常是技术信息屏幕，因为你可以从这里知道屏幕字段的名字，并且辨别出这个潜在数据对象是不是一个结构、视图或者表。

If the underlying data object is a table, your search is over. The field name shown on the Technical Information screen, plus this table name, arms you with the details you need. If the data object is a view, just one more click and you've got that table name. If the data object is a structure, you still have to unearth the name of the table.

如果潜在数据对象是一个表，你的查找就结束了。字段名称显示在技术信息屏幕中，还包括了表名称，都是你需要的细节。如果数据对象是一个视图，只需要再点击一下就能得到表名称。如果数据对象是个结构，你还不得不需要继续发掘表的名称。

Technique #1: Check the Technical Information Screen

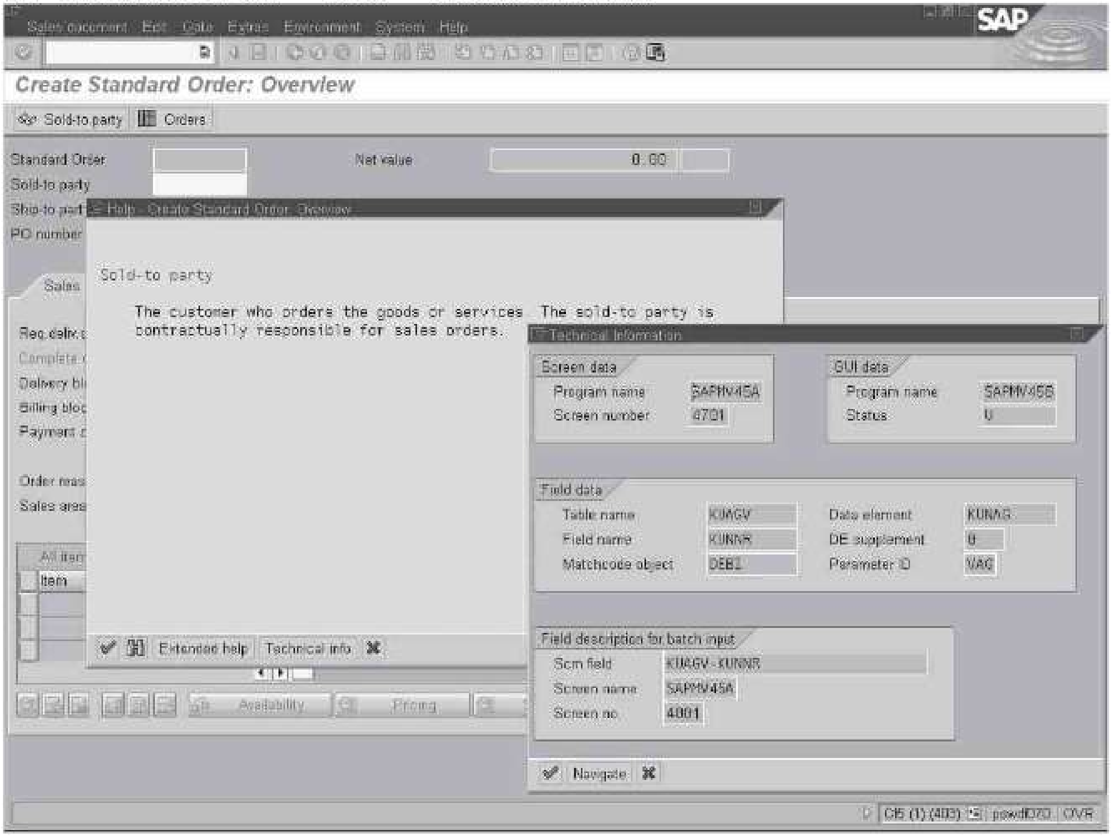
技巧一：检查技术信息屏幕

I always start my search for the underlying data object by getting the names of the field and Data Dictionary object from the transaction screen. Call up the Technical Information screen by positioning your cursor in the target screen field and then pressing **F1** or clicking on the ? button. This will bring up the initial Help screen. From within the Help screen, press **F9** or click on the **Technical info** button.

我通常从事务屏幕中开始搜寻潜在数据对象——如字段和数据字典对象的名字。如果想调出技术信息屏幕，首先将光标定位在目标屏幕字段上，然后按**F1**或者点击?按钮图标，这将打开初始帮助屏幕，在这个帮助屏幕中，按**F9**或者点击技术信息按钮。

Figure 1 The Release 4.0B Technical Information Screen for the "Sold-to party" Field

图1 在发行版4.0B中关于"售达方"字段的技术信息屏幕



Sold-to party

The customer who orders the goods or services. The sold-to party is contractually responsible for sales orders.

Screen data

Program name SAPMV45A

Screen number 4701

GUI data

Program name SAPMV45B

Status U

Field data

Table name KUAGV

Field name KUNNR

Matchcode object DEB1

Data element KUNAG

DE supplement 0

Parameter ID VAG

Field description for batch input

Scrn field KUAGV-KUNNR

Screen name SAPMV45A

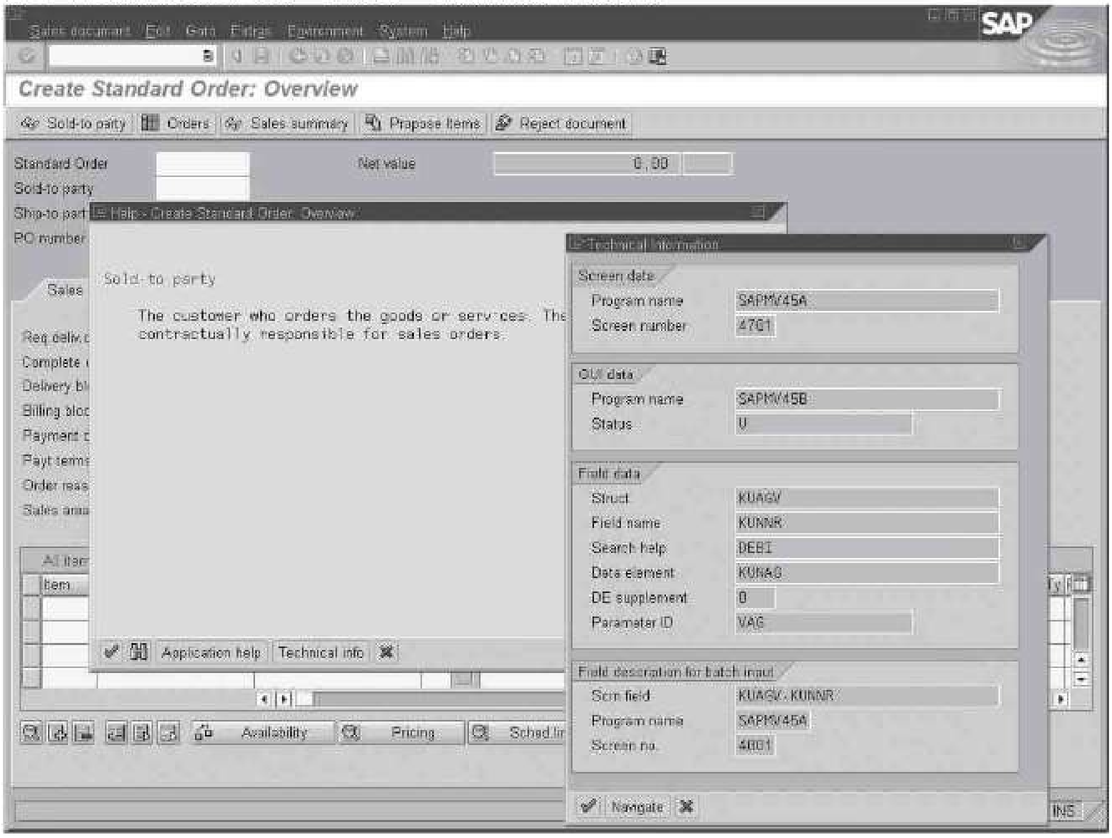
Screen no 4001

Take a look now at **Figure 1**. It shows the Technical Information screen for the **Sold-to party** field in the Create Standard (Sales) Order screen for Release 4.0B. You see in the "Field data" block that the **Table name** is KUAGV and the **Field name** is KUNNR[1]. In Releases 4.0B and below, this screen calls the associated object a "table", whether it's a table, view, or structure. Notice in **Figure 2** that the 4.5B release (and above) informs you that KUAGV is actually a structure.

现在看一下图一，它显示了在发行版4.0B中，创建标准（销售）订单屏幕的售达方字段的技术信息

屏幕。你可以看到，在“字段数据”块中，表名称是KUAGV、字段名称是KUNNR[注1]。在发行版4.0B和以前的版本中，这个屏幕把相关对象都称为“表”，而不管它是个表、视图还是结构。注意图二中，在4.5B（及以后版本）中，提示你KUAGV实际上是个结构（译注：在我们用的ECC5版中没有区分，SAP又退化了，直到双击查看对象时才显示出来它是表、结构还是视图）。

Figure 2 The Release 4.5B Technical Information Screen for the “Sold-to party” Field
图二 在发行版4.5B中关于“售达方”字段的技术信息屏幕



Sold-to party

The customer who orders the goods or services. The contractually responsible for sales orders.

Screen data
Program name SAPMV45A
Screen number 4701

GUI data
Program name SAPMV45B
Status U

Field data
Struct. KUAGV
Field name KUNNR
Search help DEBI
Data element KUNAG
DE supplement 0
Parameter ID VAG

Field description for batch input
Scrn field KUAGV-KUNNR
Program name SAPMV45A
Screen no. 4001

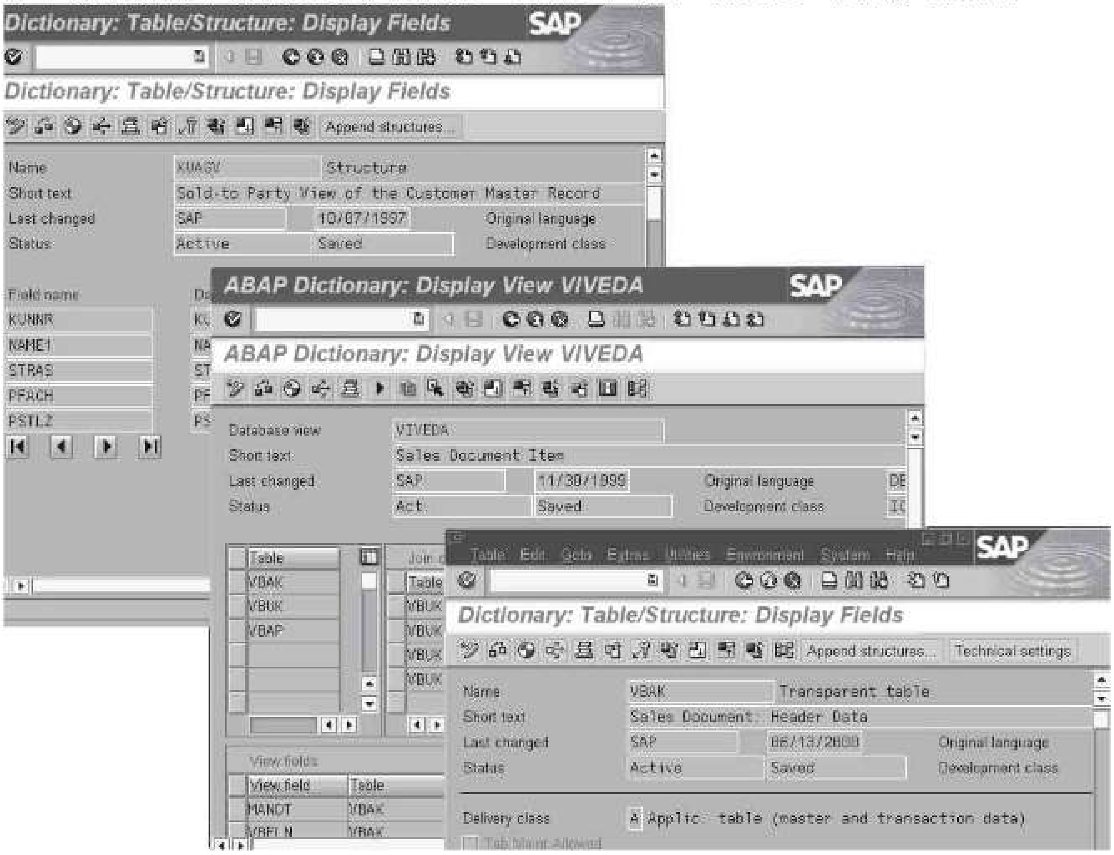
[1] The examples herein mention names of tables, structures, and fields that I found in my searches. You may find a different name in the same search — particularly a different structure’s name. For example, the **Sold-to party** field in the Sales Order initial screen may be KUAGV-KUNNR or RV45S-KUNNR (or maybe something else). I’ll show one name in this article; don’t worry if you find another. It’s a reflection of the continual enhancement of R/3, not (I hope) an error in the article.

注1 这里的例子提及了在我的研究中找到的表、结构和字段的名字。你可能在相同的研究中发现一个不同的名字——尤其是不同的结构名称。例如，在销售订单初始屏幕中售达方字段可能是KUAGV-KUNNR或者RV45S-KUNNR（还可能是其他名字）。我在文章中只写其中一个，但是如果你发现了另外的名字也不必担心。它是R/3系统持续改进的一个反映，并不是（至少我希望不是）文档中的错误。

In Releases 4.0B and below, to ascertain whether or not an object is a structure you must take one more step. Drill into the **Table name** field in the Technical Information screen to see the Data Dictionary screen for the object.

在发行版4.0B和以前的版本中，为了确定一个对象是否是一个结构，你必须多经过几步。深入技术信息屏幕的表名字段，到数据字典屏幕查看对象（译注：ECC5也是这样）。

Figure 3 The Release 4.0B Data Dictionary Screens for Table “VBAK,” View “VIVEDA,” and Structure “KUAGV”
图三 发行版4.0B中数据字典屏幕显示的表“VBAK”，视图“VIVEDA”和结构“KUAGV”



Dictionary: Table/Structure: Display Fields
SAP
Dictionary: Table/Structure: Display Fields
Append structures...

Name: KUAGV Structure
Short text: Sold-to Party View of the Customer Master Record
Last changed: SAP 10/07/1997 Original language
Status: Active Saved Development class

Field name
KUNNR
NAME1
STRAS
PFACH
PSTLZ

ABAP Dictionary: Display View VIVEDA
SAP
ABAP Dictionary: Display View VIVEDA

Database view: VIVEDA
Short text: Sales Document Item
Last changed: SAP 11/30/1999 Original language DE
Status: Act. Saved Development class IC

Table
VBAK
VBUK
VBAP

Table
VBUK
VBUK
VBUK
VBUK

View fields

View field	Table
MANDT	VBAK
VBELN	VBAK

Table Edit Goto Extras Utilities Environment System Help
SAP
Dictionary: Table/Structure: Display Fields
Append structures...
Technical settings

Name: VBAK Transparent table
Short text: Sales Document: Header Data
Last changed: SAP 06/13/2000 Original language
Status: Active Saved Development class

Delivery class: A Applic. table (master and transaction data)
□ Tab Maint Allowed

Figure 3 shows those screens for the table VBAK, the view VIVEDA, and the structure KUAGV. Corresponding screens in Releases 4.5 and 4.6 look very similar. As you can see, the object type is easy to recognize in these screens.

图三显示了表VBAK、视图VIVEDA、结构KUAGV的屏幕，在发行版4.5和4.6中，相应的屏幕也非常类似。就像你看到的，对象类型在这些屏幕中非常容易辨认。

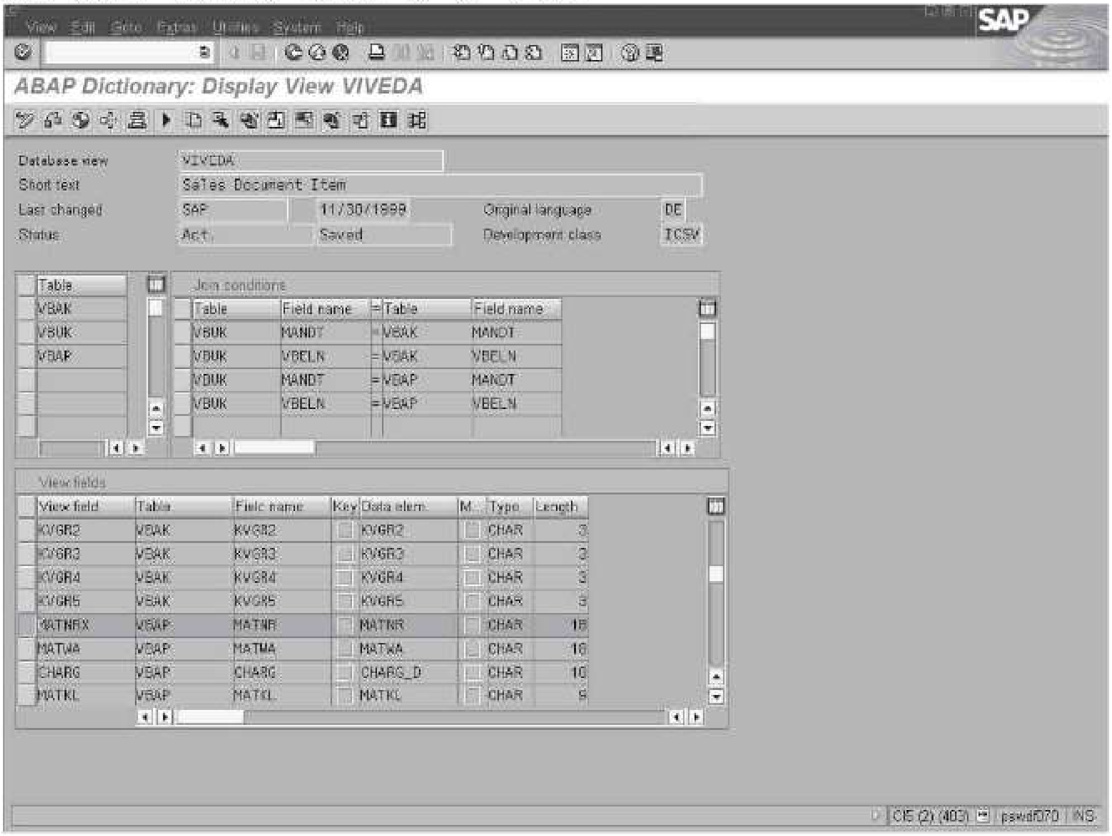
If the underlying object is a table, you’re home free. The information is stored in that table in the field with the same name as the one shown on the Technical Information screen. Success! When you write your report or ABAP Query, or when you program your enhancement, you can refer to the data using these table and field names.

如果潜在对象是一个表，你就相当轻松了。存储在那个表的字段的信息跟在技术信息屏幕中显示的名字完全相同。成功！当你写报表、ABAP查询或者增强功能程序的时候，你可以直接使用这些表和字段名来获取数据。

If the object is a view, then scroll to the **View field** in the view with the same name as the one shown in the Technical Information screen. See **Figure 4**; the **Table** and **Field name** associated with the **View field** is your goal. Success again!

如果对象是一个视图，在字段信息列表中找到与技术信息屏幕显示相同名字的视图字段。看图四，与视图字段关联的表和字段名就是你的目标。又成功了！

Figure 4 Locating the "Table" and "Field name" Associated with the "View field"
图四 定位与"视图字段"关联的"表"和"字段名"


View Edit Goto Extras Utilities System Help
SAP
CI5 (2) (403) pswdf070 INS

ABAP Dictionary: Display View VIVEDA

Database view: VIVEDA
Short text: Sales Document Item
Last changed: SAP 11/30/1999
Original language: DE
Status: Act. Saved
Development class: ICSV

Table
VBAK
VBUK
VBAP

Join conditions

Table	Field name	=	Table	Field name
VBUK	MANDT	=	VBAK	MANDT
VBUK	VBELN	=	VBAK	VBELN
VBUK	MANDT	=	VBAP	MANDT
VBUK	VBELN	=	VBAP	VBELN

View fields

View field	Table	Field name	Key	Data elem.	M..	Type	Length
KVGR2	VBAK	KVGR2	□	KVGR2	□	CHAR	3
KVGR3	VBAK	KVGR3	□	KVGR3	□	CHAR	3
KVGR4	VBAK	KVGR4	□	KVGR4	□	CHAR	3
KVGR5	VBAK	KVGR5	□	KVGR5	□	CHAR	3
MATNRX	VBAP	MATNR	□	MATNR	□	CHAR	18
MATWA	VBAP	MATWA	□	MATWA	□	CHAR	10
CHARG	VBAP	CHARG	□	CHARG_D	□	CHAR	10
MATKL	VBAP	MATKL	□	MATKL	□	CHAR	9

Tables, Views, and Structures

A screen field is associated with a table, view, or structure. While each of these is a Data Dictionary object, only the table actually stores data. This is why the goal of all these techniques is to find the table that actually stores the information that you can see displayed in the screen.

Table	View	Structure
Tables store data. If your screen field points to a table, or you are able to drill down to a table from a view, then you have found the storage location of your data, and you've reached the finish line! This table name and field name are what you need for your report or query.	Views used in screens are linked groups of tables. If the field is in a view, you can drill down to the underlying table that contains the data.	Structures are Data Dictionary objects that have fields but do not carry data. A structure is simply an array of variables defined in the Data Dictionary, and is like empty tables. If the screen field is in a structure, you'll have to keep looking. Remember, only a table actually stores data.

表、视图和结构

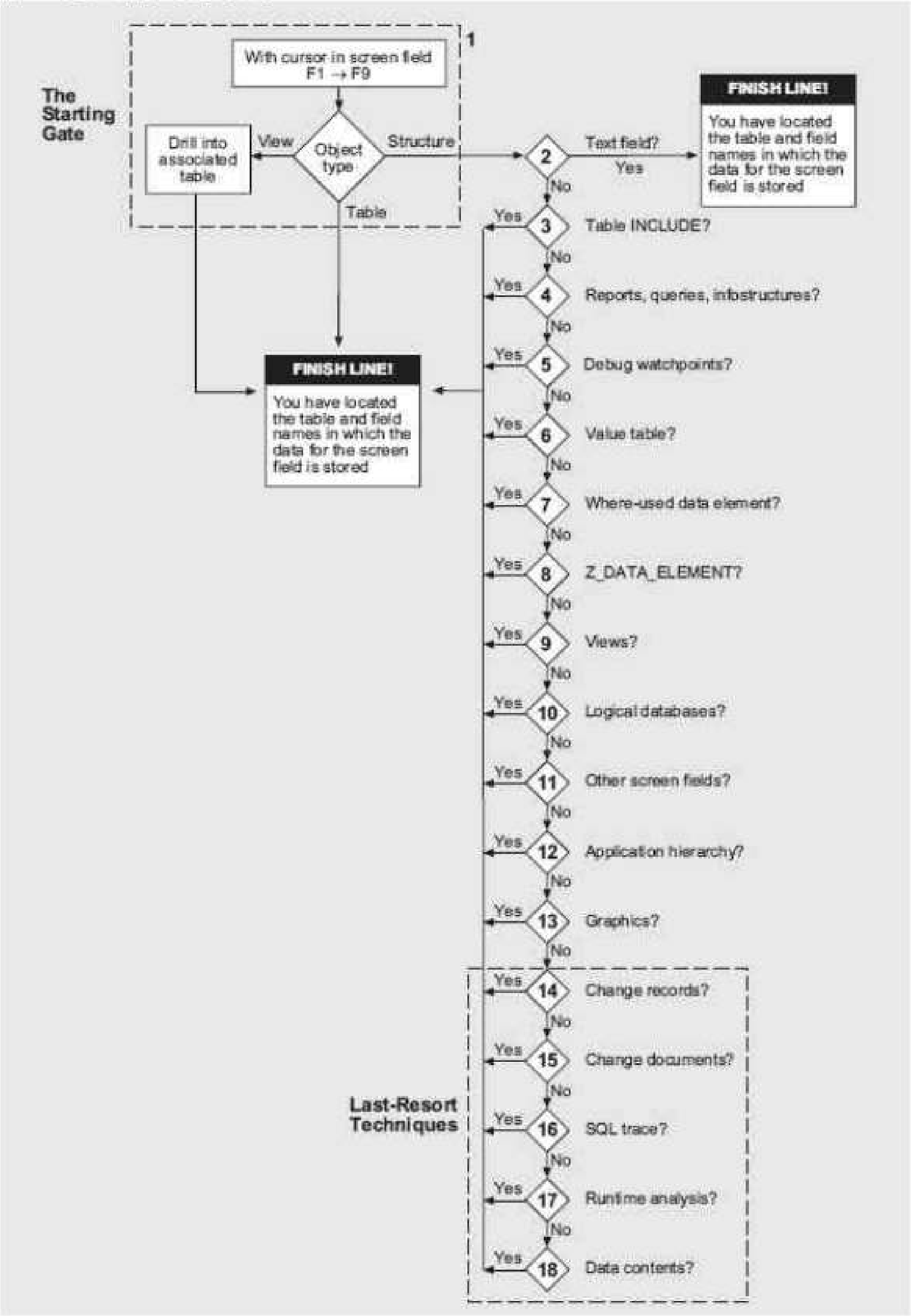
一个屏幕字段关联着一个表、视图或结构。这些都是数据字典对象，但是只有表实际存储了数据。这就是为什么所有这些技巧的最终目标都是要查找存储了你可以看到的在屏幕上显示的信息的表。

表	视图	结构
表存储数据。如果你的屏幕字段指向一个表，或者你可以从一个视图找到表，你就能找到数据的存储地点，你也就达到了终点线！这个表名和字段名就是你的报表或查询中所需要的。	视图在屏幕上用来连接一组表。如果字段在一个视图中，你可以直接深入找到它下面包含数据的表。	结构是包含字段但不存储数据的数据字典对象。结构在数据字典中仅仅是变量定义的队列，就象一个空表。如果屏幕字段在一个结构中，你需要继续查找。记住，只有表真正存储数据。

If the object is a structure, then you have more digging to do, and you must proceed to one or more of the remaining 17 techniques described here. These techniques follow an order that should get you to the finish line in the fewest number of steps. Remember, as soon as you find the associated table, you're done with the search. Look at the flowchart in Figure 5 to see my suggested sequence of techniques.

如果对象是个结构，你还有许多挖掘工作要做，你必须执行下面描述的17个技巧中的一个或多个，这些技巧的排列顺序有助于你以最少的步骤到达重点。记住，当你发现关联表的时候就完成了查找。打开图五中画的流程图，看看我对这些技巧使用顺序的建议。

Figure 5 The Suggested Sequence of Techniques
图五 建议的技巧使用顺序



Racing Toward the Finish Line: Techniques 2-13

向终点冲刺：技巧 2-13

Remember, the finish line, or goal, of this exercise is to locate the table and field names in which the underlying data for a screen field is stored. At this point, we know only that the underlying data object is a structure. Techniques 2-13, which I detail in this section, will most likely hold the answer. If techniques 2-13 do not hold the answer, try techniques 14-18 — the measures of last resort! I suggest you hold off trying these last five tactics until you've exhausted the other possibilities. If none of these 18 techniques work, you have an unusually difficult problem; I wish you good luck. I hope you find your table quickly and easily.

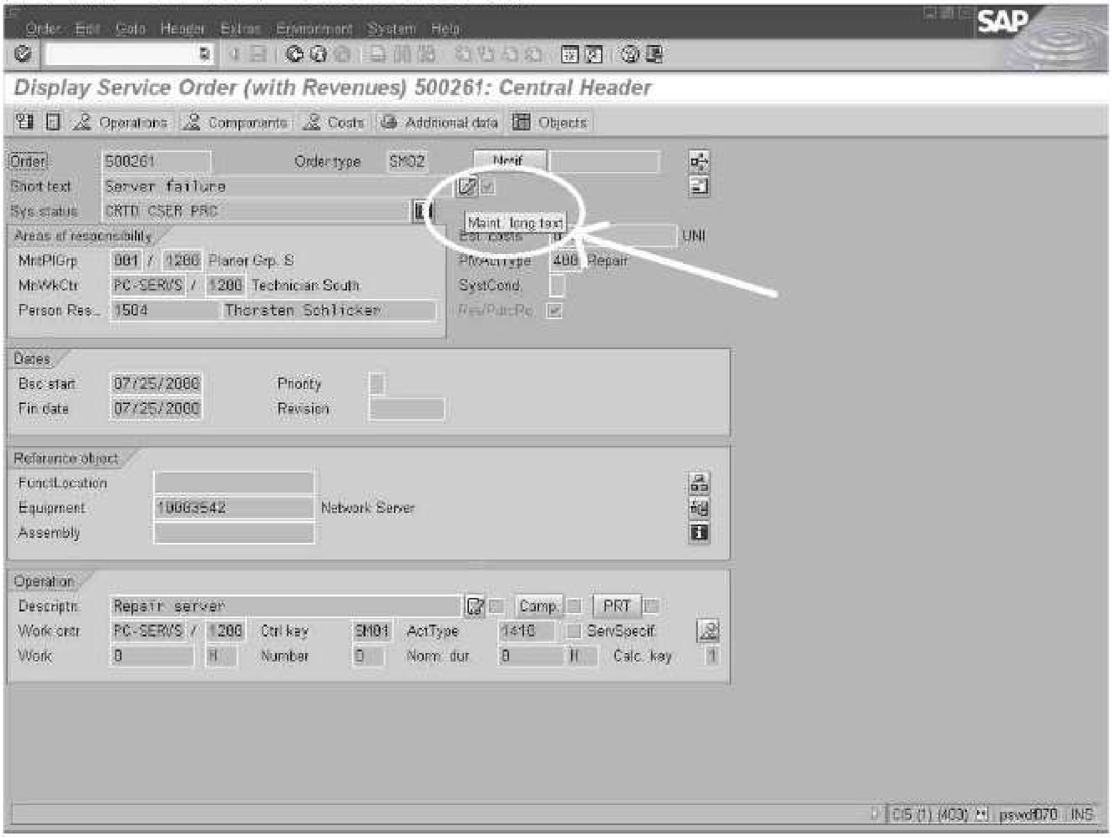
记住，本练习的终点或者说目标，是定位屏幕字段中对应的存储了潜在数据的表和字段名。从这一点来说，我们只知道潜在数据对象是一个结构。我在这一节中详细描述的技巧2至13，很可能包含了答案。如果技巧2至13仍未找到结果，尝试技巧14至18——这是最后的手段了！我建议你在完全排除了其他可能性之后再去尝试这五种策略。如果这18种技巧都不奏效，你就是遇到了一个不同寻常的困难问题，我只能祝你好运。我希望你快速而轻松的找到你的表。

Technique #2: Text Fields

技巧二：文本字段

Figure 6 Click the "Maintain long text" Icon to Get to the Long Text Screen

图六 点击"维护长文本"按钮到达长文本屏幕



Display Service Order (with Revenues) 500261: Central Header

Order 500261 Order type SM02

Short text Server failure

Sys status CRTD CSER PRC

Areas of responsibility

MntPlGrp 001 / 1200 Planer Grp. S

MnWkCtr PC-SERVS / 1200 Technician South

Person Res... 1504 Thorsten Schlicker

PMActType 400 Repair

Res/PurcRc ☑

Dates

Bsc start 07/25/2000

Fin date 07/25/2000

Priority

Revision

Reference object

FunctLocation

Equipment 10003542 Network Server

Assembly

Operation

Descriptn Repair server

Work cntr PC-SERVS / 1200 Ctrl key SM01 ActType 1410 □ ServSpecif

Work 0 H Number 0 Norm. dur 0 H Calc. key 1

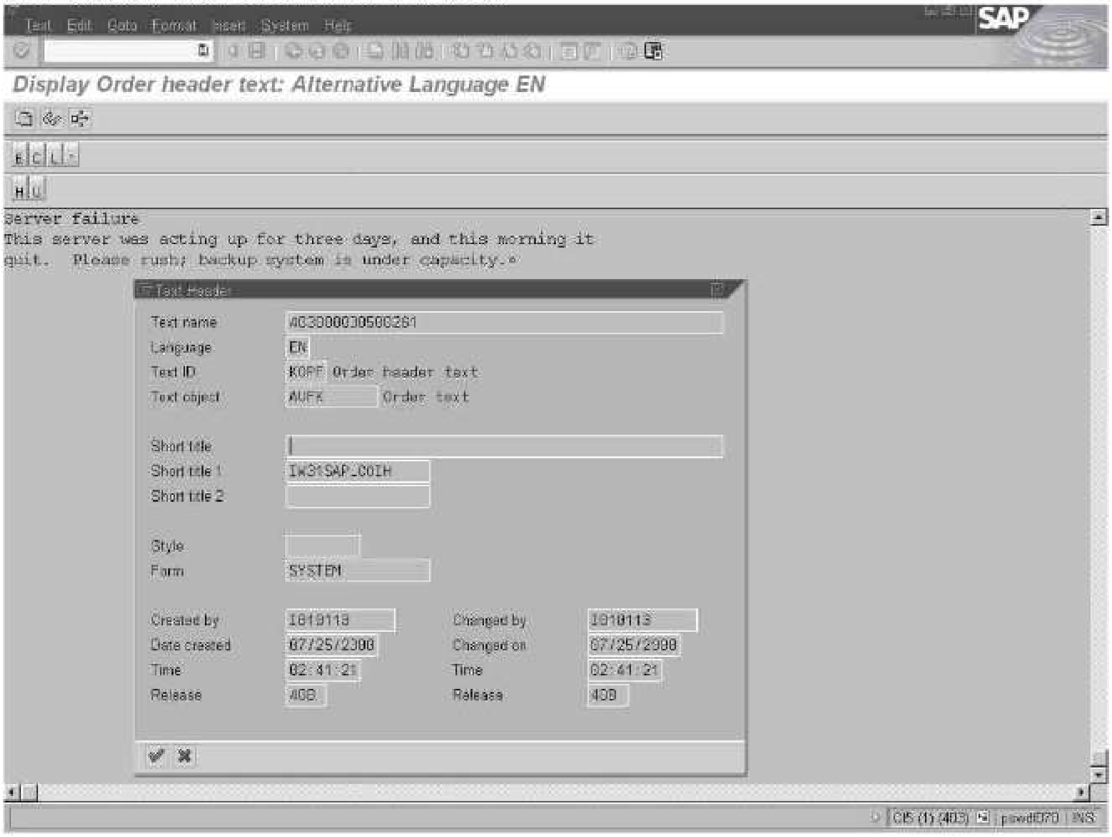
If the screen label is "Short text", or the field name is STTXT, or its data element is CO_STTXT, then you are looking for the source of a text field. As **Figure 6** shows, the Short text field often has a "Maintain long text" icon stuck at the end that may look like a sheet of paper with a pencil. Click on that icon to get to the Long text screen.

如果屏幕标签是“短文本”，或者字段名是STTXT，或者它的数据元素是CO_STTXT，你就按图六所示去找文本字段的源头，短文本字段的后面通常紧接着一个“维护长文本”按钮，看上去象一张纸和一支笔，点击这个按钮打开长文本屏幕。

Texts are stored all over the place in R/3, but frequently in tables STXH and STXL, which are the text header and text line, respectively. Their 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. In the Long text screen, follow **GoTo → Header** (see **Figure 7**) to get the **Text name**, **Text ID**, and **Text object** parameters that you'll need to execute the function module.

在R/3中文本存在各种地方，但是通常是表STXH和STXL中，分别是文本标题和文本行。它们的文本都以raw（例如，非ASCII码）形态存储，所以你必须使用功能模块来取得它们。有几个功能模块来处理这些文本，包括Read_Text，从数据库读文本行。在长文本屏幕，通过转到→表头（看图七）来访问你在执行功能模块时需要的文本名称，文本**ID**和文本对象参数。

Figure 7 Retrieving the Text Name, Text ID, and Text Object Parameters
图七 获得文本名称，文本ID和文本对象参数



Display Order header text: Alternative Language EN

```
Server failure
This server was acting up for three days, and this morning it
quit.  Please rush; backup system is under capacity.
```

Text name	403000030500261		
Language	EN		
Text ID	KOPF Order header text		
Text object	AUFK Order text		
Short title			
Short title 1	IW31SAP_COIH		
Short title 2			
Style			
Form	SYSTEM		
Created by	1010113	Changed by	1010113
Date created	07/25/2000	Changed on	07/25/2000
Time	02:41:21	Time	02:41:21
Release	40B	Release	40B

CI5 (1) (403) | pswdI070 | INS

If the field is not a text field, then you must continue to search.

如果这个字段不是一个文本字段，你必须继续查找。

Technique #3: "INCLUDE" Table

技巧三："INCLUDE"表

A structure may contain one or more INCLUDEs. An INCLUDE may actually be a table or another structure. So, if the structure you are working with includes an INCLUDE, and that INCLUDE *is* a table that contains your field, then you have won again. 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* a table that contains the field KSCHL. That table and field are your goal, and we need not go any further. If the structure doesn't contain an INCLUDE, or if the INCLUDE is not a table, then we must keep searching. The next technique you should apply is described in technique 4 because it's easy and, quite likely, will get you your answer.

一个结构可能包含一个或多个INCLUDEs。一个INCLUDE可能是一个实际的表或其他结构。所以如果你的工作结构包含了INCLUDE，而这个INCLUDE*是*一个包含了你所需字段的表，就又胜利了。例如，在销售订单项目概览中，条件类型列出的是KOMV-KSCHL，但是KOMV是一个结构。KOMV包含了INCLUDE KONV，它*是*一个包含了KSCHL字段的表。这个表和字段是你的目标，我们就不需要再做其他的了。如果这个结构并不包含INCLUDE，或者INCLUDE不是一个表，我们必须继续查找。你应该使用的下一个技巧在技巧四中说明，因为它简单，并且很可能，将获得你的结果。

Technique #4: Reports, Queries, Infostructures

技巧四：报表，查询，信息结构

If you know of any ABAP Queries, already-existing reports, or Infostructures that use the field you need, then you can dig into them to find the source table. Before you started this search, you may have found a report or query that didn't provide the result that the Operations Manager needs, but that uses the field you are looking for. Now that you have the field name from the Technical Information screen, you can review those reports and queries to see if the search has already been done for you. If you find your field in one of those reports or queries, you can examine the code and find the table that it uses to pull that field; you will have successfully concluded your search.

如果你知道ABAP查询，现存报表或者信息结构使用了你需要的字段，你可以挖掘它们来找到源表。在你开始查找以前，你可能发现一个报表或查询没有提供那位业务经历所需的结果，但是它使用了你要找的字段。现在你可以从技术信息屏幕中获得字段名，你可以看看这些报表和查询是否已经替你找到了所需的字段。如果你在这些报表或查询中找到你的字段，你可以研究代码找到列出那个字段所使用的表，你将成功的结束查找。

If you don't find your table and field here, then you must continue searching. Technique 5 is a powerful one for those who know how to use it, and can very often find the result.

如果在这里还没有找到表和字段，你必须继续查找。技巧五在我们所知的这些方法中是一个强有力的工具，经常可以找到结果。

You may have found a report or query that uses the field you are looking for. Now that you have the field name, you can review those reports and queries to see if the search has already been done for you. If you find your field in one of those reports or queries, you can examine the code and find the table that it uses to pull that field, which successfully concludes your search.

你可能发现一个报表或查询包含了你要找的字段。现在已经有了字段名，你可以看看这些报表或查询看看是否已经替你找到了所需的字段。如果你在这些报表或查询中找到了字段，你可以检查代码找到列出那个字段的表，你将成功的结束查找。

Technique #5: Debug Watchpoint

技巧五 调试观察点

If you are searching in Release 4.0B or above, and want to go to the heavy artillery immediately, then get the **Program name** (under “Screen data”) from the Technical Information dialog box and debug the program. Go to **Tools → ABAP Workbench → ABAP Editor → Program=[the program name] → Debugging**. Click on the **Watchpoint** button on the task bar and enter the program and field names into the **Local watchpoint** fields, then click on **Enter** and **F8**. Any change in the variable triggers the watchpoint, presenting you with the code immediately below the line that changed the variable. Read the code above the watchpoint carefully, and you can probably find the name of the Data Dictionary object that the field is in. Again, it may be a table, view, or structure, but you are getting very close.

如果你在4.0B及以上版本中，想要立刻深入了解，不妨直接从技术信息对话框中获取程序名称（在“屏幕数据”下），然后进行调试。到工具 → **ABAP** 工作台 → **ABAP** 编辑器 → 程序=[输入程序名称] → 调试。点击人物栏上的观察点按钮，在本地观察点字段中输入程序和字段名称，然后按回车或**F8**。这些变量的变化都会触发观察点，立即把你引导到改变变量的下一行代码。仔细读观察点上面的代码，你也许会发现数据字典对象所在字段的名字。或许它仍然可能是表、视图或者结构，不过你已经非常接近了。（译注：在ECC5中，已经不必这么麻烦了，程序运行到任何时候在事务码栏输入/h就可以启动调试，不过对于标准程序来说，用此方法找到所需表和字段基本上是Mission Impossible，标准程序的层次嵌套实在太多了，而且读写数据库基本上都是在函数中，要一层一层的追踪到函数，需要极大的耐心和毅力。）

Ken Greenwood (author of *Teach Yourself ABAP/4 in 21 Days*) says he always jumps immediately to this technique if the Technical Information screen doesn’t produce the table. Ken further suggests that this technique finds the table right away — with the only exception being the IMPORT statement. If the watchpoint doesn’t trigger, then debug again using a breakpoint on IMPORT and you will find it in a maximum of three steps.

Ken Greenwood（《21天自学ABAP/4》的作者）说，他经常在技术信息屏幕不显示表名称的时候立即使用此技巧。Ken还提到这个技巧总能找到表——只有使用IMPORT语句时是个例外。如果观察点没有被触发，就在IMPORT上再次用断点调试，这样最多只需要三步就能找到它。

If you can’t find the table reference you need in the code, then you must continue your search, and the next natural technique for you to use is described in technique 6.

如果你无法在代码中找到你需要的表，还必须继续查找，下一个简单的方法就在技巧六中说明。

Technique #6: Value Table

技巧六 值表

If debugging didn’t give you what you need or you don’t want to debug a program, then back out to the Technical Information dialog box and double-click on the **Data element** field under “Field data”. This will show you the value table, if there is one. If you’re just looking for master data like customer name — whose field name is KUNNR — then a value table will probably be defined, and it will give you what you want. You’ll find that transaction data like Order Quantity — whose field name is KWMENG — has no value table, and you get to keep looking. From here on, techniques 7-13 all have about the same likelihood of revealing your table name. Use them in any order you like. You’ll probably find that you like some more than others, so use those first.

如果调试无法给你想要的，或者你不想调试一个程序，就返回技术信息对话框，双击“字段数据”下的数据元素字段。如果有值表的话，这样就会把值表显示出来。如果你只想找主数据，例如客户名称——字段名是KUNNR——值表通常都会被定义，它会给你想要的。你会发现事务数据如订单数量——字段名是KWMENG——没有值表，你得继续找。从下面开始，技巧7至13对你找到表名都有类似的启迪作用，可以按任意顺序使用它们。你可能发现自己比较喜欢某些方法胜过其他的，那就优先使用它们。

（译注：很遗憾，在ECC5里面没有找到Value Table，不知道它究竟是个什么东西，有知道的朋友请跟我联系，谢谢。）

If debugging didn’t give you what you need or you don’t want to debug a program, then back out to the Technical Information dialog box and double-click on the Data element field under “Field data”. This will show you the value table, if there is one.

如果调试没能给你想要的或者你不想调试一个程序，就返回技术信息对话框，双击“数据字段”下的数据元素字段。如果有值表的话，就会把值表显示出来。

Technique #7: “Where-Used” Data Element

技巧七 “所用处”数据元素

When R/3 moves data from one table (or structure) to another, it often (not always) moves it into a target field having the same data element as the source field. From the Technical Information dialog box, double-click on the **Data element** field under “Field data” and follow **Utilities → Whereused List → Table fields → Enter** (see **Figure 8**). 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.

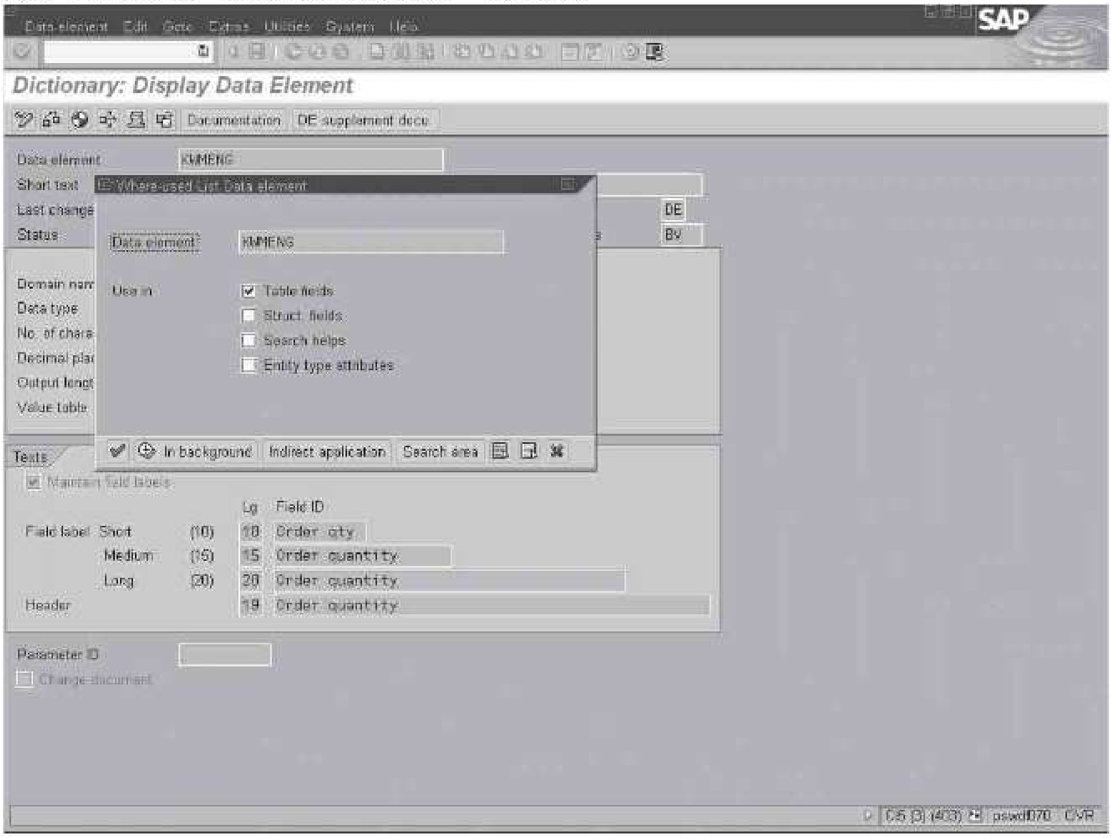
当R/3把数据从一个表（或结构）移动到另一个中，它经常（但也并非总是）把它移动到与源字段有相同数据元素的一个目标字段中。从技术信息对话框中，双击“字段数据”下的数据元素字段，然后点实用程序 → 所用处清单 → 表格字段 → 回车 （参见图八）。这将列出所有使用这个数据元素的表、视图和结构。如果你的字段是经常使用的，这个列表会十分庞大而难以找到数据存储位置。例如，在4.0B版中KUNNR字段会列处743个对象。一个不常用的字段会列出很少的对象，你可以从列表中找出数据源。

Note also that the list includes structures; sometimes, lots of structures. When you drill into a field name in the list, you’ll see the table display that will identify it as a structure or a transparent table. If it’s a structure, you'll have to keep looking.

还需要注意，这个列表中也包括了结构，有时候还有很多个结构，当你在列表中查找时，你将会看到表显示为结构或者透明表，如果这是个结构，你还要继续找。

（译注：当然你一定也看到了，在查看所用处清单时，会让你选择在哪些数据对象中查找，完全可以不选择结构，这样出来的就都是表了。）

Figure 8 "Where-used List Data element" Request Screen in Release 4.0B
图八 4.0B版中的"所用处清单数据元素"请求屏幕



Data element: KMMENG

Use in
☑ Table fields
□ Struct. fields
□ Search helps
□ Entity type attributes

Data element: KMMENG

			Lg	Field ID
Field label	Short	(10)	10	Order qty
	Medium	(15)	15	Order quantity
	Long	(20)	20	Order quantity
Header			19	Order quantity

☑ Maintain field labels

Parameter ID

□ Change document

C05 (3) (403) pswdf070 OVR

Technique #8: "Z_DATA_ELEMENT"

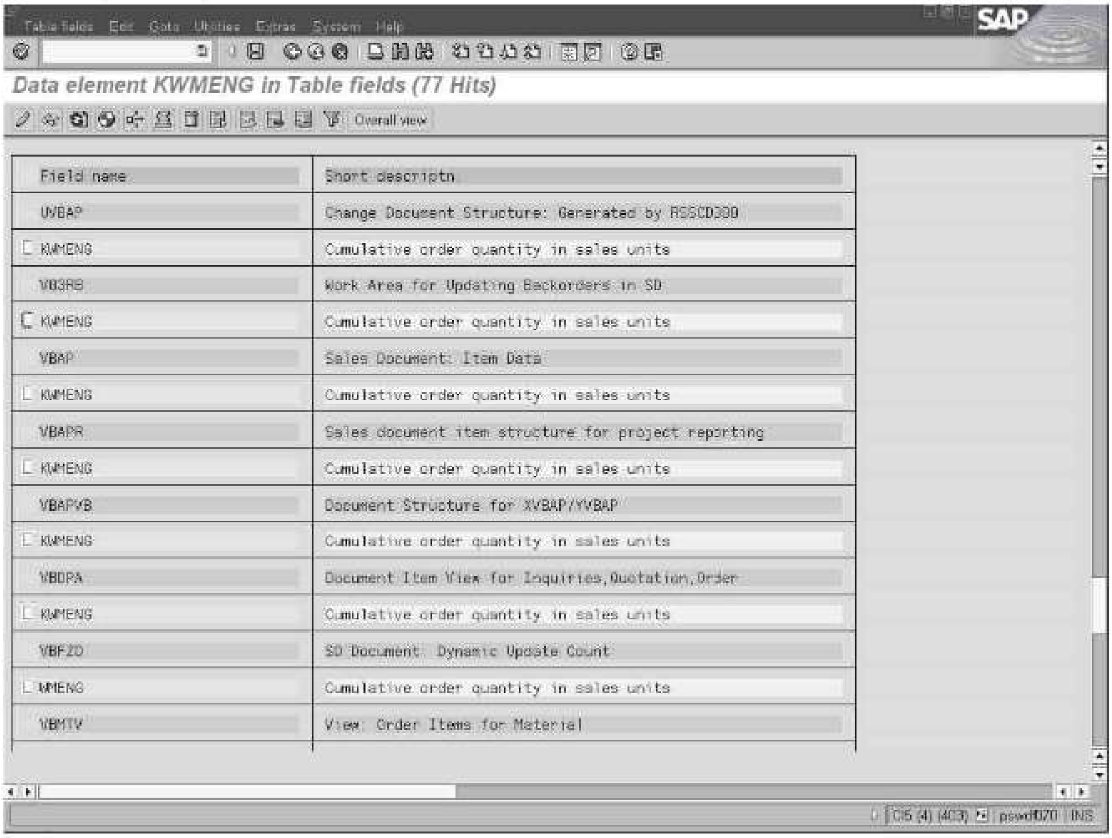
技巧八 "Z_DATA_ELEMENT"

Since the "Where-used" list for a data element generally contains so many (non-data carrying) structures, the list is not terribly useful for our purpose. Take a look at **Figure 9**, for example. It has 77 hits.

由于一个数据元素的"所用处"清单经常包含了太多（不包含数据）的结构，这个清单并不十分好用。看看图九，它包含了77个提示项。

Figure 9 Data Element "Where-used" List

图九 数据元素"所用处"清单



Data element KWMENG in Table fields (77 Hits)

Field name	Short descriptn
UVBAP	Change Document Structure: Generated by RSSCD300
□ KWMENG	Cumulative order quantity in sales units
VB3RB	Work Area for Updating Backorders in SD
□ KWMENG	Cumulative order quantity in sales units
VBAP	Sales Document: Item Data
□ KWMENG	Cumulative order quantity in sales units
VBAPR	Sales document item structure for project reporting
□ KWMENG	Cumulative order quantity in sales units
VBAPVB	Document Structure for XVBAP/YVBAP
□ KWMENG	Cumulative order quantity in sales units
VBDPA	Document Item View for Inquiries,Quotation,Order
□ KWMENG	Cumulative order quantity in sales units
VBFZD	SD Document: Dynamic Update Count
□ WMENG	Cumulative order quantity in sales units
VBMTV	View: Order Items for Material

You can create the Data Dictionary view Z_DATA_ELEMENT, which will show only the data tables that use your data element.

你可以创建数据字典视图Z_DATA_ELEMENT，它只显示使用了你的数据元素的数据表。

Create the view with **Tools → ABAP Workbench → Dictionary → Object name=Z_DATA_ELEMENT → Views → Create**, then enter the values shown in **Figure 10**.

使用工具 → **ABAP** 工作台 → 字典 → 对象名称**=Z_DATA_ELEMENT** → 视图 → 创建 来创建视图，然后输入图10中显示的数据。

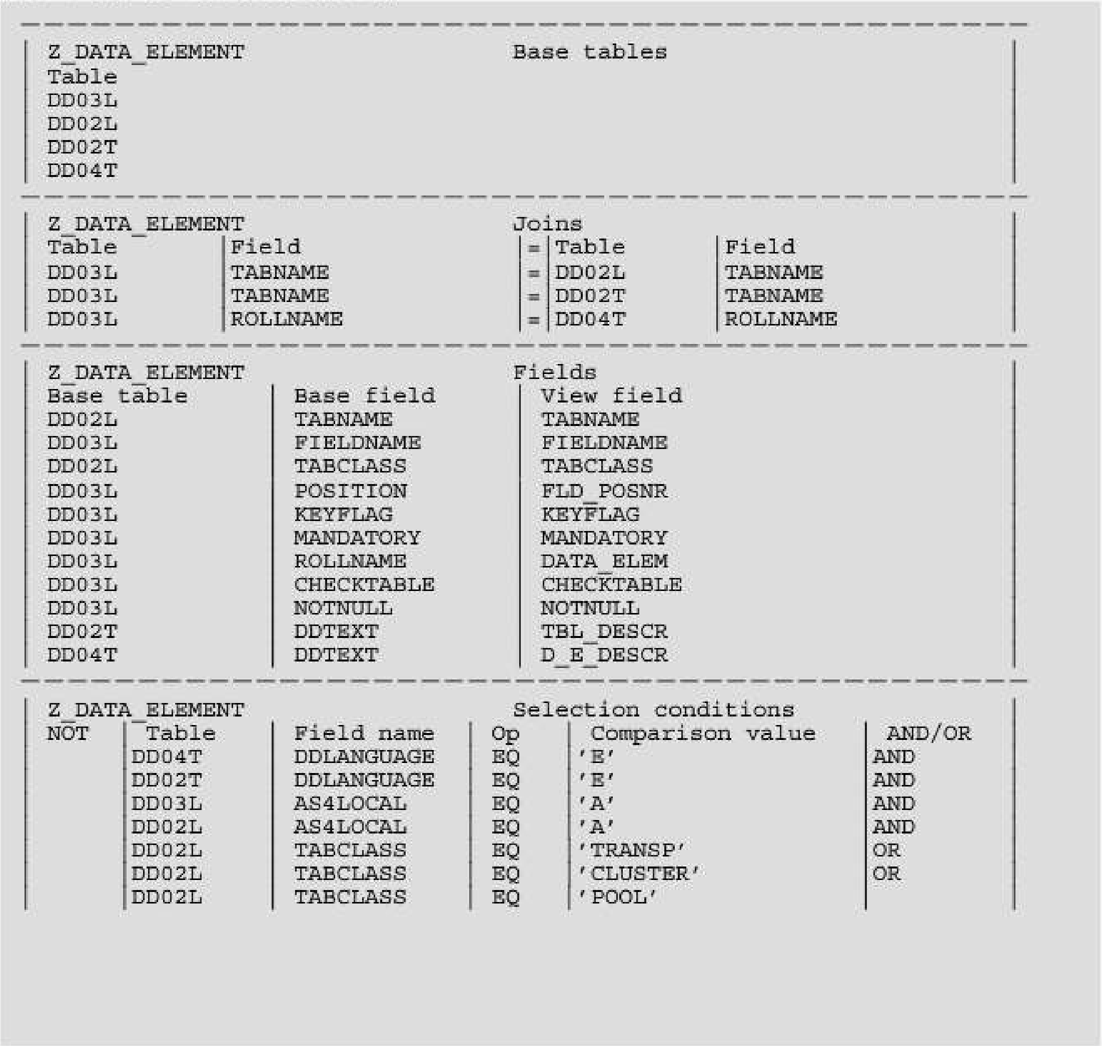
Use this tool with **Tools → ABAP Workbench → Dictionary → Object name=Z_DATA_ELEMENT → Views → Display → Utilities → [Display data or Table contents] → DATA_ELEM=[the name of the data element you are searching for] → [Execute or F8]**.

使用工具 工具 → **ABAP** 工作台 → 字典 → 对象名称**=Z_DATA_ELEMENT** → 视图 → 显示 → 使用程序 → **[**显示数据或表内容**] → DATA_ELEM=[**你要找的数据元素的名称**] → [**执行或按**F8]**.

This view will give you a list of only the tables that use your data element.

这个视图将给出只包含使用你的数据元素的表的清单，而没有视图或结构。

Figure 10 Components of a View for Searching Data Elements
图十 查找数据元素所需的视图组件



```
 ------------------------------------------------------------------------
| 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 conditions                 |
| NOT  | Table      | Field name   | Op  | Comparison value   | AND/OR  |
|      |DD04T       | DDLANGUAGE   | EQ  | 'E'                | AND     |
|      |DD02T       | DDLANGUAGE   | 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'             |         |
```

译注：这是利用SAP现有的知识库做成的一个视图，其实是个定制开发而成的产物，跟技巧七所说的所用处清单的方法没有太多差别。SAP系统的所有应用表的结构，都存储在知识库（Repository）中，知识库的相关知识请参考SAP标准教程BC400。

Technique #9: Views
技巧九 视图

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 screen. 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 KWMENG in the VBAP table.

事务数据通常存储在几个分层连接的表中，R/3可能会定义一个或多个视图来处理事务。你可以通过查询这些视图来发现其他相关的表。这些表里可能会存储你需要的数据。例如，假定你想从“销售订单：单行概览”屏幕中得到订单数量数据。它被显示为RV45A-KWMENG，但是RV45A是一个结构。你已经知道订单编号是VBAK-VBELN字段，于是你可以找VBAK表的所用处清单所对应的视图，视图VIVEDA连接着VBAK、VBUK和VBAP这几个表，可以发现KWMENG在VBAP表中。（译注：在ECC5中，VIVEDA只关联VBAK和ABAP两个表，去掉了VBUK。）

Technique #10: Logical Databases

技巧十 逻辑数据库

Tables that are related in a business process are frequently connected in logical databases. The VBAK “Where-used” list for logical databases, for example, includes AAV, which also shows the connection to VBAP.

与商务过程相关的表通常都与逻辑数据库相连。例如，在“所用处”清单中VBAK的逻辑数据库，包含了AAV，它也显示出了与VBAP的连接。

（译注：这跟前面的技巧类似，都是所用处清单的不同用法。）

Technique #11: Other Screen Fields

技巧十一 其他屏幕字段

One or more of the other fields in the same transaction 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 RV45SKUNNR, but you find that RV45S 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** or **Standard Order** is listed as VBAK-VBELN. When you look up the VBAK table, you find that it also includes KUNNR.

相同事务中的一个或多个其他字段可能直接与它们的数据库表相连。如果你可以发现事务的头表，它很可能包含了你需要的字段。例如，假定你需要销售订单初始屏幕的售达数据，它显示的是RV45S-KUNNR，但是你发现RV45S是一个结构，它的数据元素是KUNAG，但是在4.0B版中使用那个数据元素的有276个表，所以这里面找到它是很艰苦的。在相同的屏幕，订单或者标准订单列为VBAK-VBELN，当你查看VBAK表时，会发现它包含了KUNNR字段。

（译注：有时甚至不是在当前屏幕，还可以在当前程序的其他屏幕来查找相关的字段。）

Technique #12: Application Hierarchy

技巧十二 应用程序层次

In 4.0B or above, go to **Tools** → **ABAP Workbench** → **Overview** → **Data Browser**. Press **F4** in the *empty* **Table name** field, then click on **SAP Applications** → **Application Hierarchy**. Drill down through the menu tree to find the tables associated with your application. Your field may well be in one of those tables.

在4.0B或以上版，使用 工具 → **ABAP** 工作台 → 概览 → 数据浏览器. 在*空的*表名字段上按 **F4**，然后按 **SAP** 应用程序 → 应用程序层次. 在菜单树里面找与你应用程序相关的表。你的字段或许正好在这些表当中。

（译注：应用程序层次是个很庞大的结构，如果能以此生成一个大的表格倒是不错的。）

Technique #13: Graphics

技巧十三 图形

In 4.0B or above, go to **Tools → ABAP Workbench → Dictionary**, then type in the name of a table you believe is related to your field. Follow the path **Display → [Extras or Utilities] → 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 you may be able to find where your field is located.

在4.0B以上版本，使用 工具 → **ABAP** 工作台 → 字典，然后输入你认为与字段相关的表名。按照菜单路径 显示 → **[**附加或应用程序**]** → 图形**.** 你将会看到你选定表和其他关联表的关系图。你可以使用这些看到与一个业务对象关联的整个表家族，也可以看到你所需要的字段。

（译注：使用图形有时候会得到过于庞大的表连接图，而这些关联主要是跟主数据的，多数时候使用效果并不是特别好。）

（译注：总结前面的这些做法，除了使用Z_DATA_ELEMENT外，其他都是用的系统现有功能，找到的相关表通常都比较多，只有对相关业务熟悉了才能比较快的找到表，对不熟悉的模块，找表仍是个非常艰苦和细致的工作。）

The Last Resort: Techniques 14-18

终极手段：技巧14-18

These are the techniques of last resort, which you should turn to only if you've exhausted all other possibilities.

这些技巧是最后的手段，当你用尽了前面所有努力时不妨试试。

Technique #14: Change Records

技巧十四 修改记录

R/3 allows for changes that are made to many documents to be recorded in change records, and frequently the change information includes the name of the table and field affected. If change recording is turned on, 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. Depending on the R/3 release, here are some examples of menu paths to change records:

R/3允许在变更记录中记下许多变更文档，变更信息经常包含了受影响的表和字段的名字。如果变更记录被打开，你可以在一个文档中修改你要寻找的字段，然后读变更记录来找到表和字段名。根据不同的R/3版本，下面有几个变更记录的菜单路径。

• In Plant Maintenance or Service Management *Technical Objects*: **Extras → [Display Changes or Environment] → Display Changes** *Order*: **Extras → Order documents → [Changes or Extras] → Display Changes**

Notification: **Extras → Notification documents → Changes**

Contracts & Planning: **Environment → Changes**

• 在工厂维护或者服务管理*技术对象*：附加 → [显示变更或环境] → 显示变更 *订单*： 附加 → 订单 → [变更或附加] → 显示变更

通知：附加 → 通知文档 → 变更

合同和计划编制：环境 → 变更

- **In Sales and Distribution**

  Order: **Environment → Changes**

  Delivery: **Environment → Changes**

- 在销售和配送

  订单：环境 → 变更

  配送：环境 → 变更

If more than one change is shown, drill into the change you're interested in (generally the most recent). 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.

如果显示了多个变更，选择你感兴趣的那个（通常是最近的）变更继续深究。表TCDOB和TCDOBT列出了变更文档创建的所有可能对象类。你可以从那里得到你的程序是否支持变更文档的线索。

（译注：对业务不熟悉，没看懂这个方法。）

Technique #15: Change Documents

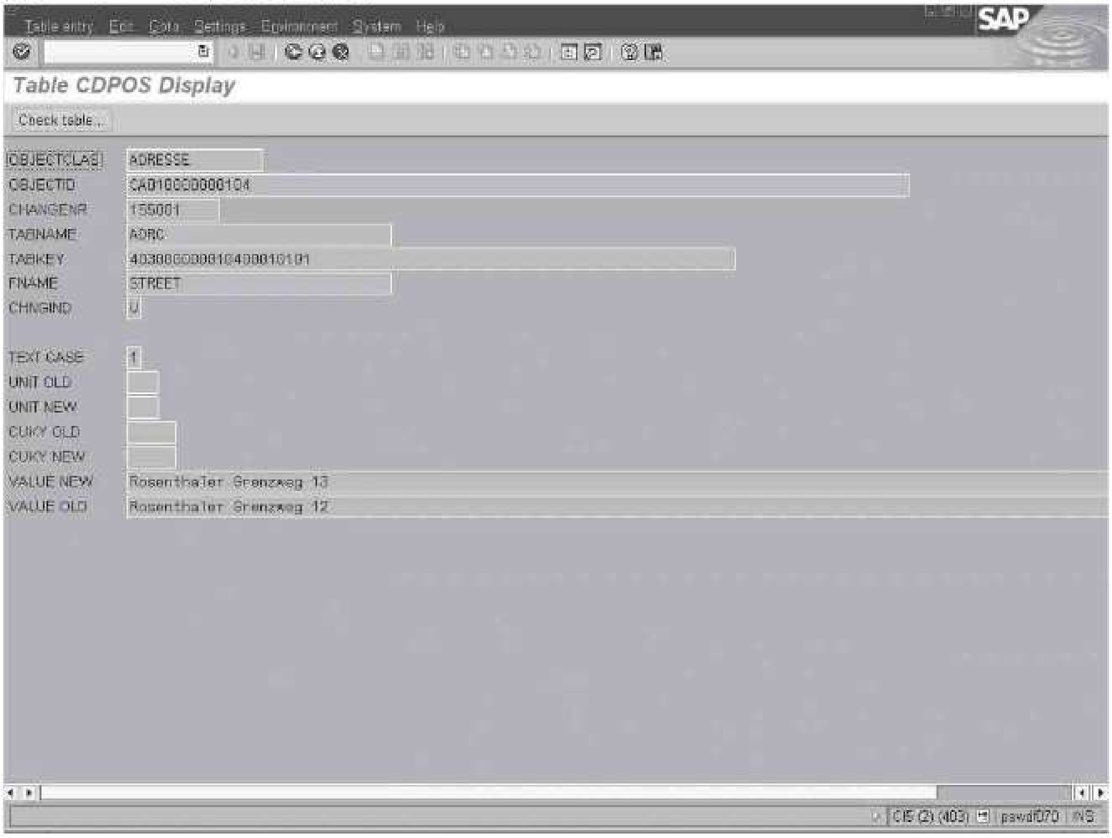
技巧十五 变更文档

If you can't find the change record 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: "Change Document Header" and "Change Document Position"!) as follows. After you have changed and saved the field you're searching for, open CDHDR in the ABAP Workbench dictionary, then 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 (**Figure 11**). This record contains the names of the table and field affected, as well as the new and old values, so you can ensure that you're looking at the right record.

如果你在事务里无法找到变更记录，跟着就看看变更文档表CDHDR和CDPOS（现在是英文和德文命名规则的混合："变更文档标题"和"变更文档位置"）。你改变和保存了你搜寻的字段后，在ABAP工作台字典中打开CDHDR，然后按照 实用程序 → 表内容 来找变更记录。在选择屏幕中输入今天日期、时间段和你的用户名。可能得出一个匹配记录。用 **Ctrl+C** 拷贝CHANGENR的值，用它从CDPOS中获得适当的记录（参考图十一）。这个记录包含了受影响的表和字段名，还有变更前后的值，这样你就可以确定你找到的是正确的记录。

（译注：系统既然有表保存这些变更记录，就应该有个事务码来查询分析它，直接用se16查看表内容，似乎有点不合适。如果谁知道有查看变更文档的事务码，请加以补充。）

Figure 11 Change Table Illustrations in Release 4.0B
图十一 在4.0B版中变更表的例子



OBJECTCLAS	ADRESSE
OBJECTID	CA0100000000104
CHANGENR	155001
TABNAME	ADRC
TABKEY	403000000010400010101
FNAME	STREET
CHNGIND	U
TEXT CASE	1
UNIT OLD	
UNIT NEW	
CUKY OLD	
CUKY NEW	
VALUE NEW	Rosenthaler Grenzweg 13
VALUE OLD	Rosenthaler Grenzweg 12

Technique #16: SQL Trace
技巧十六 SQL跟踪

If all else fails, it's time for some serious archeology. In one session, go to your transaction but don't run it yet. In another session, go into **Tools → ABAP Workbench → Test → SQL Trace → Trace On → Ok**. Switch to your transaction session and run your transaction. Switch back to the trace session and click **Trace off → List trace → [Execute or 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. Use **System → List → Save → Local file → unconverted → Enter → File name=[c:\mypath\filename.doc] → Transfer**. Open "filename.doc" in Word, 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.

如果其他方法都失败了，就得用点严肃的考古方法了。一种方式，到你的事务里但先不要运行；另一种方法，到 工具 → **ABAP** 工作台 → 测试 → **SQL** 跟踪 → 打开跟踪 → 确定。切换到你的事务并开始运行。再切换回跟踪任务并点击 结束跟踪 → 跟踪清单 → **[**执行或**OK]**。你将得到一个包含了所有参说的SQL调用列表。这会比较便于下载到一个字处理文档里进行搜索。使用 系统 → 清单 → 保存 → 本地文件 → 不改变 → 回车 → 文件名=**[c:\mypath\filename.doc]** → 传输。在Word里打开文件"filename.doc"，搜索你的字段名，注意与它关联的表。要确定哪个表确

实存储了数据还包括一定的侦测工作。这种方法需要对SQL命令和原理的理解。

（译注：做ABAP肯定要对SQL命令有一定了解，所以这个不成问题，而且区分哪些是查询哪些是写入也很容易，因此这种方法用的还是比较多。）

Technique #17: Runtime Analysis

技巧十七 运行时分析

Alternatively, use runtime analysis with **System** → **Utilities** → **Runtime Analysis** → **Execute** → **[Transaction=your_transcode or Program=your_program_name]** → **F8**. After you post the transaction or complete the program, the system will return you to the Runtime Analysis screen. Choose **Analyze**, then **Tables** or **Table Hit List** (**F6**) to see all the transparent and pool tables that were accessed during the transaction. **Hit List** (**F5**) will show you all the program calls and specify the ABAP program that was running. Sometimes the Where-used list from the "Structure" display with **Programs** checked will list a program in which the field is associated with the database. Compare the programs in that list to those you found in the Runtime Analysis Hit List to narrow your search.

还可以选择，使用运行时分析，用 系统 → 实用程序 → 运行时分析 → 执行 → [事务=你的事务码 或 程序=你的程序名] → **F8**。在你提交了事务码或者完成程序的时候，系统会返回运行时分析屏幕。选择分析，然后是表或者表命中清单**(F6)**来查看在事务执行过程中访问到的透明表和池表。命中清单(F5)将显示出所有运行时使用的程序调用和ABAP程序。有时被选择的程序会同时显示"结构"的所用处清单并列出字段关联数据库的程序。比较那个清单中的程序和你在运行时分析命中清单的程序来缩小你的搜索范围。

（译注：在ECC5中，路径为 工具 → **ABAP**工作台→ 测试 → 运行时分析 ）

Technique #18: Data Contents

技巧十八 数据内容

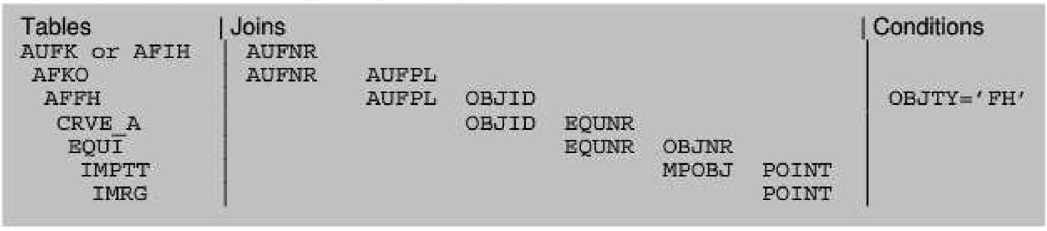
Finally, extract some data from the set of tables associated with your transaction, and copy those records into a text file. Use the "Search" function in your word processor to find data flows by content where the source and destination fields may have different names and data elements. R/3 often uses internal numbers that you'd never think of, such as Object ID and Object Type, to link records. 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.

最后，从你的事务中关联的表中提取出一些数据，并且把这些记录复制到文本文件中。在你的字处理软件中使用"查询"功能通过内容找出数据流，。R/3经常使用你无法想象的内部编码，例如对象ID和对象类型，来连接记录。然而它们的值相当与众不同，当你从几个事务中提取数据后，你也许可以通过搜索这些值（忽略字段名）而发现数据流的路径。

For example, to find Measurement documents associated with a service order, we needed to trace the connections shown in **Figure 12.** There was nothing obvious about which fields linked the data.

例如，为了找到与服务订单关联的计量单位文档，我们需要按图十二中显示的跟踪连接。这里不会有任何明显的关于字段连接数据的东西。

Figure 12 Data Path from Service Order to Measurement Document
图十二 从服务订单到计量单位文档的数据路径



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

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.

我们在R/3中创建一个服务订单和几个计量单位文档。然后我们提取出以服务订单号开头的记录，然后追踪各种连接以找出所有记录的关联。当使用SE11功能显示每个表的时候，我们用 实用程序 → 表内容 来摘取出数据。

We exported the records as text documents using **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 by following the value of the contents. OBJID, OBJNR, and MPOBJ contain internal numbers that "mean" nothing to the transaction, yet they linked the records.

我们使用 系统 → 清单 → 保存 → 本地文件 来把这些记录导出成文本，然后合成一篇Word文档，然后使用Word的"查找"功能通过追溯内容的值来从表中找到数据。OBJID、OBJNR和MOPBJ包含了对事务毫无用处的内部编号，然而它们与记录相关。

（译注：这种方法看上去就感觉头晕，我也没有具体实践过，一般来说，使用SQL跟踪就足够了，修改值后查看SQL跟踪的数据库更新，总能找到目标字段。）

Helpful Hints

有益的提示

√ Many transactions have header information and detail or item lines in separate linked tables. 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 and VBAP. When you are looking for linked tables containing the data for a transaction, this tip may help you identify one if you have the other.

√ 许多事务通过分离的连接表记录了头信息、细节或行项目。记住R/3是一个德国产品，"Kopf"是德语中的"头"，而"position"可以翻译成德语的"细节"或"项目"。你将经常发现头/项表的名字中包含一对"K"和"P"，例如VBAK和VBAP。当你为一个事务查找包含数据的连接表时，这个提示可能帮助你在有一个表时快速找到另一个。

√ There are small differences between R/3 Releases 3.1I, 4.0B, 4.5B, and 4.6B for the menu paths and directions used in this article. For example, Release 3.1I menus refer to the ABAP/4 (Workbench, Editor, etc.) whereas higher releases refer to ABAP. The Release 4.0B **ABAP Workbench → ABAP Editor** menu path shows up in Release 4.6B with the extra step **ABAP Workbench → Development → ABAP Editor**. The figures in this article were taken from a 4.0B system unless there was an important difference in appearance on another release. Many installations today are still using Release 3.1, and most are running 4.0 or below, so those images are important. The corresponding 4.5 and 4.6 screens are virtually identical to Release 4.0B except for Figure 2, so there's no need to show the other figures in those releases as well.

√ 本文中使用的菜单路径和用法在R/3系统的3.1I、4.0B、4.5B、4.6B等版本中会略有不同。例如，在3.1I的菜单提到ABAP/4（例如工作台、编辑器等）而更高版本仅仅写ABAP。4.0B版的**ABAP** 工作台 → **ABAP** 编辑器 菜单路径在4.6B版中扩展了一步变为**ABAP** 工作台 → 开发 → **ABAP** 编辑器。这些特性在本文中都是以4.0B系统为主，除非在其他版本中有很明显的不同。现在许多系统仍在使用3.1版，多数都是4.0或者以下版本，所以这些图片很重要，相应的4.5和4.6版的屏幕除了图二之外都和4.0B版的一样，所以不需要列出其他版本的图片了。

√ Release 3.1I and below can't execute technique 5 because they don't have the watchpoint functionality in their debuggers, or techniques 12 and 13 because they don't have the application hierarchy or graphics functions.

√ 3.1I版及更早版本无法使用技巧五，因为它们的调试器中没有观察点功能，也无法使用技巧十二和十三，因为它们没有应用程序层次和图形功能。

Conclusion

结论

SAP's three-tiered architecture is the backbone of an R/3 infrastructure. It's what makes an R/3 system flexible, reliable, and open, all at the same time. It's also what makes locating underlying data, at times, so inherently difficult. Hopefully the techniques presented here will help. Opt for the ones that you find easiest and most convenient to use. Leave the others as techniques of last resort. Some fields may still resist these techniques; there are no doubt other ways yet to chase them down. If you know of other techniques (or corrections to these techniques), please e-mail me at dennis.barrett@sap.com.

SAP的三层结构是R/3基础结构的中枢。它使R/3系统同时具有灵活性、可靠性和开放性。但同时它也造成定位潜在数据时的固有困难。希望这里介绍的技巧会有帮助。选择使用其中你觉得简单和方便的方法，把其他技巧作为终极手段。有些字段可能在上述技巧下仍无法现形，毫无疑问还有其他

办法可以找出它们来。如果你知道其他技巧（或者纠正上述技巧），请通过电子邮件联系我：dennis.barrett@sap.com。

Acknowledgements

感谢

Thank you to the many SAP and partner consultants who gave me suggestions and feedback, and to those who reviewed this article.

感谢所有给我提供过建议和反馈的SAP及其合作伙伴的顾问们，并感谢阅读本文的读者。

*Dennis Barrett is an applications consultant with SAP America who focuses on Service Management/Customer Service and the Service Provider solution. He has been consulting in computers for over 15 years, always blending applications and programming. He's a certified ABAP programmer and often facilitates communications between business process owners and programmers. Dennis is the author of "SAP R/3 ABAP/4 Command Reference," published by Que/MacMillan, and wrote "Customer Service ABAP Tables for Programmers," published this spring in the PM/CS e-Newsletter, which is sent to all SAP consultants in Plant Maintenance and Customer Service. Dennis maintains a personal Web site at **www.mindspring.com~dennis.barrett**, which includes copies of the PM/CS articles as well as white papers and other items of interest to ABAP programmers and CS consultants. You can reach him at dennis.barrett@sap.com.*

本文翻译完成于2007年5月。希望能够对 ABAP 开发的初学者提供一些帮助。

非常感谢我从事 SAP 开发这一年来为我提供 ABAP 技术指导的几位顾问老师：毛俊宏、冯耀武、谭奇勇、胡跃明、邱加亮、吴均，也非常感谢 QQ 群 5381003（SAP ABAPer 6-1）、13976797（SAP SD MM PP）中各位高手的帮助。

请大家对文章提出改进意见，您可以直接修改和补充本文档的中文翻译并继续发布，但请保留原作者的英文原稿，便于读者理解分析。

欢迎各位与我联系，交流 ABAP 开发技巧和心得。

E-mail:	qiangsheng@gmail.com
MSN:	qiangsheng@hotmail.com
QQ:	1011370
Blog:	比较懒，从来没写过，😜

强晟

2007年5月8日