



Thomas Stutenbäumer

# Practical Guide to SAP ABAP®

## Part 2: Performance, Enhancements, Transports

- ▶ **Developer influence on performance**
- ▶ **Modifications and enhancements to SAP standard**
- ▶ **SAP access and account management techniques**
- ▶ **SAP Transport Management System**

# Table of Contents

<b>Preface</b>	<b>7</b>
<b>1 The requirements and design of a solution</b>	<b>11</b>
1.1 A requirement from practice	11
1.2 Results achieved so far	12
1.3 What still has to be implemented	14
<b>2 The influence of developers on performance</b>	<b>15</b>
2.1 Performance analysis	16
2.2 Table indexes and performance	27
2.3 Program development and performance	32
2.4 Conclusion	38
<b>3 Changing and enhancing standard SAP applications</b>	<b>41</b>
3.1 Modifications	42
3.2 User exits	53
3.3 Event function modules with FQEVENTS	70
3.4 Enhancements for the business partner with the area menu BUPT	80
3.5 Enhancements of services for objects	96
3.6 Enhancing the CIC	107
3.7 Enhancements with the Enhancement Framework	126
<b>4 Access protection and authorization in SAP</b>	<b>159</b>
4.1 SAP authorization concept	159
4.2 Authorizations in ABAP developments	161
4.3 Missing authorizations	163
<b>5 Customizing SAP</b>	<b>165</b>
5.1 Transaction SPRO	166
5.2 Current settings	172

<b>6</b>	<b>SAP Transport Management System</b>	<b>179</b>
6.1	The central role of table TADIR	186
6.2	The Transport Organizer	187
6.3	Releasing transport requests	194
6.4	Transporting table content	195
6.5	Adding objects to transport requests	197
6.6	Deleting transport requests	200
6.7	Collective transports	201
6.8	Transport Organizer tools	204
6.9	Avoiding typical transport errors	206
6.10	Cross-transports	207
<b>7</b>	<b>Troubleshooting</b>	<b>217</b>
7.1	Searching for errors in ABAP code	217
7.2	Errors in SAP coding	222
7.3	Implementing SAP Notes	227
<b>8</b>	<b>SAP Service Marketplace</b>	<b>237</b>
8.1	Activating single sign-on (SAP passport)	239
8.2	Developer keys and SSCR keys	242
8.3	Investigation with SAP Service Marketplace	248
8.4	Reporting an SAP incident	251
<b>9</b>	<b>Epilogue</b>	<b>263</b>
<b>A</b>	<b>The Author</b>	<b>266</b>
<b>B</b>	<b>Example programs</b>	<b>267</b>
<b>C</b>	<b>Useful SAP transactions</b>	<b>268</b>
<b>D</b>	<b>Useful SAP database tables</b>	<b>270</b>
<b>E</b>	<b>Index</b>	<b>271</b>
<b>F</b>	<b>Disclaimer</b>	<b>274</b>

## 2 The influence of developers on performance

**The performance of SAP systems is a widely discussed subject. In SAP environments with mass data in particular, badly developed applications and insufficiently maintained databases cause such a poor system response that the use of the application becomes unreasonable. Therefore, it is worthwhile dedicating some time to the performance.**

The concept *performance* refers to the achievement of an IT system. IT systems are made up of different components: hardware (server and clients), networks (Internet and intranet with active and passive components), middleware, databases, and applications. To guarantee the desired performance, the IT service provider has to consider all components. As a result of these efforts, the user sees only the response time of the application.

There are a lot of publications on the market that cover the optimization of SAP systems—for example, by means of load-balancing (the equal distribution of the server activity over several application servers).

In the following I limit myself to the important contribution of ABAP developers to increasing performance.

On one hand, the job of an IT supporter involves *performance monitoring* for the current productive environment. In larger SAP installations, the implementation of a task force concerned exclusively with the system performance pays off. This task force can be made up of one employee from the SAP Basis team and one SAP developer. On the other hand, the IT supporter has to guarantee that customer developments deliver sufficient performance. This can be achieved, for example, by a team running performance tests before new applications go live. The causes of any performance losses have to be eliminated.

Needlessly long program runs cause high energy consumption—due to the high CPU load—and also tax the patience of the user. In practice, I have seen a lot of programs which urgently needed a performance opti-

mization. Lack of knowledge and technical sensitivity on the part of the developer are often the reason behind long runtimes.

In this chapter, I will first explain the options for *performance analysis* in SAP systems. I will then demonstrate the influence of *table indexes*. And last but not least, I will explain *programming techniques* which guarantee high performance. In the explanations I use the example program ZCU\_ABAP\_PERFORMANCE. This program uses different programming techniques to select the contract accounts, contracts, and installations for business partners from an Oracle database. The test system contains 58,232 business partners, 111,653 contract accounts, and 197,812 contracts.

## 2.1 Performance analysis

There are different tools for performance analysis in SAP systems. Examples include:

- ▶ Performance trace with transaction ST05
- ▶ ABAP runtime analysis with transaction SAT
- ▶ Single transaction analysis with transaction ST12

In the following I will explain the first two tools.

### 2.1.1 Analysis with transaction ST05

Transaction ST05 is probably the best known tool for performance analysis. As an alternative to using the transaction code, you can access the tool via the pull-down menu SYSTEM • TOOLS • PERFORMANCE TRACE. This tool allows you to perform different performance analyses:

- ▶ SQL trace: records database access to programs and transactions
- ▶ RFC trace: records access to RFC function modules (remote function call) across several SAP systems
- ▶ Enqueue trace: records object locks

- ▶ Buffer trace: records access to the table buffer
- ▶ HTTP trace: analyzes the data interface between the SAP server and the web application server when using the Web Dynpro technology

The performance analysis allows you to record one or several of these aspects. The detailed explanation below is limited to the *SQL trace*, that is, the recording of database accesses.

In an additional screen session we use the example program ZCU\_ABAP\_PERFORMANCE and select the programming technique VIEW. In a program run, 2,295 records are selected for 300 selected business partners (table BUT000) with their contract accounts (table FKKVKP) and their contracts (table EVER).

Before starting the program, use transaction ST05 to access the performance analysis on another screen (see Figure 2.1).

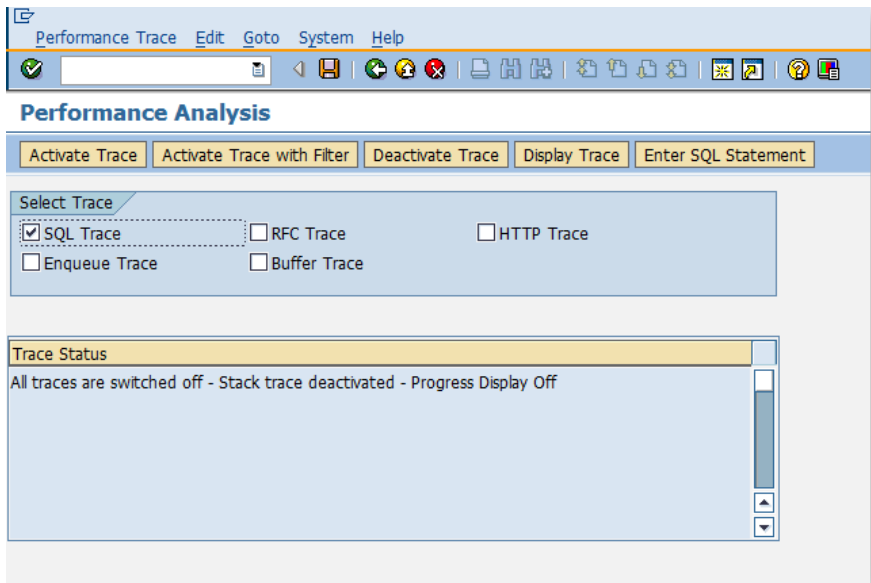


Figure 2.1: Selection screen of transaction ST05 (performance trace)

After selecting the SQL trace, start the performance recording of database accesses for the current user by clicking **Activate Trace**.

## No parallel program runs during performance analysis




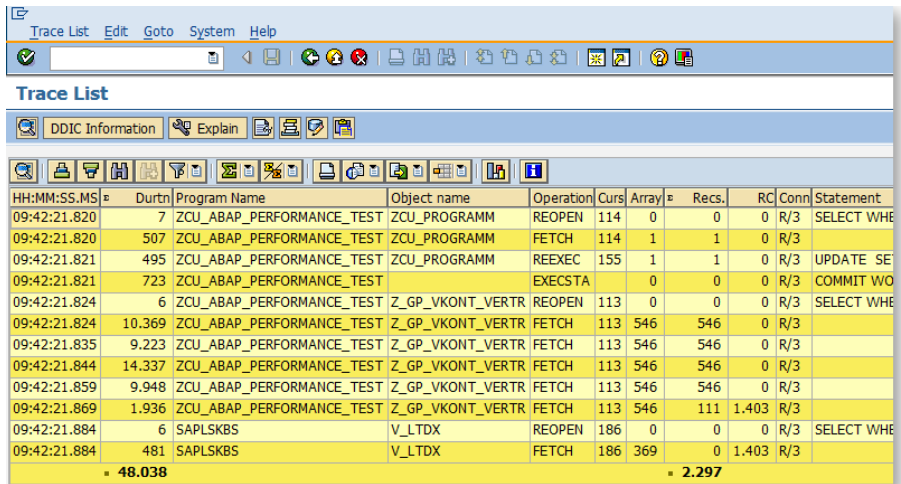
When using the performance analysis, make sure that no further programs are running on the SAP system under your user name (e.g., background programs), otherwise the analysis will return incorrect results.

Now start program ZCU\_ABAP\_PERFORMANCE in another session. As soon as the program displays the selected records, change to the session where the SQL trace performance analysis is running. Click **Deactivate Trace** to stop the performance recording.

Next, start the display of the SQL trace by clicking **Display Trace**.

Figure 2.2: Selection screen for displaying the SQL trace

On the subsequent screen (see Figure 2.2), you can restrict the selection of the performance trace. On this selection screen, the period of the last performance trace of the user is defaulted. Accept the selection without any changes by clicking . The log of the database accesses of the last program run appears.



HH:MM:SS.MS	Durtn	Program Name	Object name	Operation	Curs	Array	Recs.	RC	Conn	Statement	
09:42:21.820	7	ZCU_ABAP_PERFORMANCE_TEST	ZCU_PROGRAMM	REOPEN	114	0	0	0	R/3	SELECT WHE	
09:42:21.820	507	ZCU_ABAP_PERFORMANCE_TEST	ZCU_PROGRAMM	FETCH	114	1	1	0	R/3		
09:42:21.821	495	ZCU_ABAP_PERFORMANCE_TEST	ZCU_PROGRAMM	REEEXEC	155	1	1	0	R/3	UPDATE SE	
09:42:21.821	723	ZCU_ABAP_PERFORMANCE_TEST		EXECSTA		0	0	0	R/3	COMMIT WO	
09:42:21.824	6	ZCU_ABAP_PERFORMANCE_TEST	Z_GP_VKONT_VERTR	REOPEN	113	0	0	0	R/3	SELECT WHE	
09:42:21.824	10.369	ZCU_ABAP_PERFORMANCE_TEST	Z_GP_VKONT_VERTR	FETCH	113	546	546	0	R/3		
09:42:21.835	9.223	ZCU_ABAP_PERFORMANCE_TEST	Z_GP_VKONT_VERTR	FETCH	113	546	546	0	R/3		
09:42:21.844	14.337	ZCU_ABAP_PERFORMANCE_TEST	Z_GP_VKONT_VERTR	FETCH	113	546	546	0	R/3		
09:42:21.859	9.948	ZCU_ABAP_PERFORMANCE_TEST	Z_GP_VKONT_VERTR	FETCH	113	546	546	0	R/3		
09:42:21.869	1.936	ZCU_ABAP_PERFORMANCE_TEST	Z_GP_VKONT_VERTR	FETCH	113	546	111	1.403	R/3		
09:42:21.884	6	SAPLSKBS	V_LTDX	REOPEN	186	0	0	0	R/3	SELECT WHE	
09:42:21.884	481	SAPLSKBS	V_LTDX	FETCH	186	369	0	1.403	R/3		
							<b>48.038</b>				<b>2.297</b>

Figure 2.3: SQL trace of a program run with the programming technique VIEW

The performance log in Figure 2.3 shows that in total, the selection of the records takes 48,038 microseconds (0.000001 seconds). Hence, the program run lasts approximately 0.048 seconds.

On certain lines you can navigate directly to the record source code. Select a line and double-click one of the buttons shown in Figure 2.4. This allows you to get additional information from the trace log.

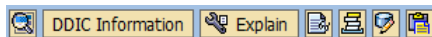



Figure 2.4: Button strip in SQL trace

The  Explain button, for example, shows the SELECT statement recorded for the corresponding line (see Figure 2.5).



# E Index

## &

&SAP\_EDIT 195

## A

ABAP stack 219

Authorization

- Access management 159
- Account management 159
- Authorization management 159
- Missing authorizations 163
- Objects 159
- Profiles 159
- Roles 159

## B

BAdI

- Call BAdI 145
- Classic 144
- Components 144
- Construction 146
- Default implementation 148
- Definition 147
- Fallback class 147
- Implementation 144
- Implementing class 147
- Runtime behavior 149
- Standard SAP 144
- Technology 143

BOR object

- BOR method 112
- BOR type 108
- Create an instance 119
- Generate 118
- Release 121
- Status of components 121
- Test 119

BUPT

- Area menu 80

- Dataset 87
- Events 91
- Field group 85
- Fields 86
- Screens 89
- Section 89
- Views 86

## C

CIC

- Context menu 107
- Customer Interaction Center 107

Current settings 172

Customizing 165

- Activity importance 170
- Change log 170
- Client-independent 165
- Client-specific 165
- IMG 167
- SPRO 166
- Tables 165
- Transport request 165

## D

Database utility 31

Development package

- SAP packages 183

Development Package

- Customer packages 183

Dirty assign 93

Dump 217

## E

Enhancement

- Composite 128
- Create implementation 139
- Dynamic spot 130
- Enhancement types 130

- Explicit 127
- Framework 126
- Implementation 131
- Implicit 128
- Implicit enhancement point 136
- Information System 128
- SE20 128
- Section 128
- SPAU\_ENH 143
- Spot 128
- Static spot 131
- Enhancements
  - Call customer function 55
  - Components (SMOD) 54
  - Projects (CMOD) 53
  - Search components 58
  - Search projects 65
  - Services for object 96
- Error handling 217

## F

### FQEVENTS

- Customer function module 77
- Event function modules 70
- Events 71
- Example function module 75
- Standard function module 76
- Table TFKFBC 80
- Table TFKFBM 80
- Table TFKFBS 80

## I

- Import queue
  - Add requests 214

## M

- Message maintenance 224
- Modification 42
  - Adjustment 42
  - Browser 42
  - Developer key 45
  - Modification Assistant 49

- Modification procedure 46
- Release change 42
- SSCR key 45
- Undo 52

## P

- Performance 15
  - Database access 19
  - Index 27
  - Monitoring 15
  - Performance analysis 16
  - Performance trace 16
  - Principles 38
  - Runtime analysis 21
  - SQL trace 17
- Programming technique 32
  - Nested loops 33
  - Nested select 38
  - Pre-selection 36
  - Table fetch 36

## S

- SAP Community Network
  - SCN 249
- SAP Knowledge Base 249
- SAP note
  - SNOTE 228
- SAP Note 227
  - Can be implemented 232
  - Change adoption 234
  - Implementation 227
  - Implementation status 234
  - Note Assistant 227
  - Note Browser 228
  - Release Information 229
  - Reset 236
- SAP Service Marketplace 237
  - Developer key 242
  - SAP Passport 242
  - Single sign-on 239
  - SSCR key 242
- SAP Support
  - Incident priority 257

- Incident Wizard 252
- Opening system 260
- Report an incident 252
- SAP component 256
- SAP Note 250
- SAP Support Portal 242
- SAP System
  - Client concept 180
  - Logical 179
  - Physical 179
- Screen
  - Layout editor 48
- Screen Exit
  - Customer subscreen 61
- Services for object
  - CL\_GOS\_SERVICE 100
  - Service type 105
  - Subservice 106
  - Superclass 100
  - Table SGSOATTR 105
- T**
- Table index 16, 30
  - Attributes 28
  - Customer-specific index 27
  - Delete 31
  - Standard SAP index 27
- TADIR 186
  - Original system 45
  - Repair flag 45
- Transport
  - Cross-transport 207
  - Development package 181
  - Emergency transport 185
  - Export queue 184
  - Financial rules 184
  - Going live 185
  - Import options 45
  - Import queue 184
  - Internal control system 184
  - Release procedure 184
  - Repair 192
  - Table content 195
  - Table TADIR 186
  - Transport directory 208
  - Transport files 208
  - Transport Organizer* 187
- Transport error 206
  - Foreign developments 206
  - Overtaker 206
  - Overwriter 206
  - Transport tool 207
- Transport management
  - Organizer tools 204
- Transport Management System
  - 213
  - TMS 180
  - Transport layer 182
- Transport request
  - Collective transport 201
  - Customizing 180
  - Deletion 200
  - Display 190
  - Include objects 197
  - Key entries 194
  - Local development objects 182
  - Modifiable 187
  - Objects in requests 204
  - Released 187
  - Request types 188
  - Workbench 180
- Transport Request
  - Import options 215
- U**
- User exit 53
  - Client-specific 67
  - Release-stable 53
  - Screen exit 53