

Performance Guide

SAP SEM/BW-BPS 3.1B, 3.2, 3.5

SAP BW-BPS 3.5

Analysis of a Used Case

Technical Background Information

Optimization Proposals

SEM/BW/BW-BPS Statistics in Detail

Last changed on 10.07.03

Valid for SEM/BW 3.1B, from SP3, SEM/BW 3.2, 3.5
BW-BPS 3.5

Changelog:	3
1 General Information About this Guide	4
2 General Prerequisites for the Optimization	4
2.1 Going Live Checks	4
2.2 Relevant SAP Notes Available	4
3 The Analysis Process	4
3.1 Determining a Used Case	4
3.2 Determining the Execution Time for Each Step	5
3.3 Analyzing the Execution Time of a Step: SEM/BW/BW-BPS Statistics	7
3.4 Analyzing the Execution Time for Each Step: Application Statistics	11
4 Assessment of Analysis Results	13
4.1 Optimization	13
5 Background Knowledge	13
5.1 Planning Functions	14
5.2 Manual Planning	14
5.3 WEB	17
5.4 Memory Consumption on the Application Server	17
5.5 CPU Consumption	18
6 Optimization Proposals	18
6.1 Planning Modell	18
6.2 Planning Functions	18
6.2.1 General Recommendations	18
6.2.2 Exit Function	19
6.2.3 Formula Function	20
6.3 Manual Planning	21
6.4 BW	22
6.5 WEB	23
6.6 Variables	23
7 Measuring and Details on SEM/BW/BW-BPS Statistics	24
7.1 Useful Tables	26
7.2 Useful Transactions	26
7.3 Automatic Evaluation of Statistical Data	26
7.4 Memory Consumption and CPU Time	26

Changelog:

18/11/2003	6.1 Planning Modell	Multiarea versus basis area
10/10/2003	6.5 WEB	Load balancing
10/10/2003	6.2.3 Fox	Rule for reading reference data - variables
10/10/2003	6.6 Variables	Buffering
29/10/2002	5.1 / 5.2 / 5.3	Some grafics have been added.
08/10/2002	6.2 Manual Planning	Size of Excel templates
07/10/2002	6.3 BW Optimizations	Close of transactional request triggers upload procedure
07/10/2002	6.3 BW Optimizations	Proposal of BW Aggregates for Planning Level
07/10/2002	6.1.2. Exit function	Comments on the Initialization Module added
10/07/2003	6.3 BW Roll up	No automatic role up any more

1 General Information About this Guide

This document is intended as a guide for optimizing the existing Customizing for SEM/BW/BW-BPS.

2 General Prerequisites for the Optimization

Before the settings in SEM/BW/BW-BPS can be checked, the entire system needs to have parameters set according to the settings recommended by SAP. This is essential in order to ensure that the optimal conditions are fulfilled for analyzing Customizing. The components that are usually integrated with the SEM/BW/BW-BPS system and can thereby influence its performance are an SAP BW system, R/3 Basis, and a Web server (ITS/WAS).

You can obtain recommendations for the settings of these components in two ways: performing a Going Live check and referring to the relevant SAP notes listed below.

2.1 Going Live Checks

Each customer is entitled to three Going Live checks free of charge. However, the onus is on the customer to first request these checks. To do so, create a note for the component XX-SER-TCC. The checks should be performed according to the following time schedule: 4 to 6 weeks prior to going live (General Analysis - GA) When going live (GO) Approx. 6 weeks after going live (General Verification - GV).

2.2 Relevant SAP Notes Available

The notes listed below only cover some of the settings checked in the Going Live check. This is why the check should ideally be scheduled as outlined above.

Current kernel version	SAP notes 19466 and 138704
Database parameters	SAP note 358921 (SEM/BW/BW and Oracle) SAP note 197404 (BW and MSSQL) SAP notes 181945 and 119372 (BW and Informix) SAP note 390016 (BW and DB2/390) SAP note 307077 (BW and DB4/AS/400) SAP note 302429 (BW and DB6/DB2 UDB EEE)
System parameters	SAP note 192658 (recommendations for BW, which also apply to SEM/BW/BW-BPS)
SEM/BW/BW Performance	SAP note 358529 (Overview of performance notes)

3 The Analysis Process

It is generally not possible to check the entire Customizing in one go when searching for the root cause of a performance problem. Instead, it is a matter of analyzing different Customizing aspects at a time, using a number of used cases. Each used case should include a central and significant scenario that is intended to be used later in the operational system. Such a scenario could contain the most important elements of a user's planning session.

During the analysis process, a used case is divided up repeatedly into ever smaller units (as described below) and the resource consumption of each unit is determined. These units are then assessed separately and optimized where possible. The main focus is on the runtimes: the user experiences the runtime directly and this is where a need for improved performance is generally identified.

3.1 Determining a Used Case

A used case should not entail more than 10-15 planning steps. A planning step represents the execution of a planning function (such as the copy function) or of manual planning. Note that a planning sequence is not to

be considered *one* single planning step, even if it seems like one from the perspective of the user. Instead, each planning function contained in the sequence is executed as a step in the used case description. In this section, you find an example of a used case description. The description contains a numbered list of the planning steps and the technical names of the planning area, the planning level, the planning package, the planning method, and the parameter group. These entries are required in order to identify uniquely the planning step executed. In the case of manual planning, the planning method is always 0-MP. The technical name of the layouts corresponds to the name of the parameter group. Furthermore, the date and time when the planning step was executed is noted and the duration of the execution is measured. To simplify the subsequent analysis, it makes sense to calculate the end time (= start time + execution time) as well. Note that, for time measurements and the comparison with the data from the statistical entries, the times can differ from each other. This is the case, for example, if you are in a different time zone to that of the system or the system is set internally to another time zone. In the following example, local time is two hours ahead of normal time: In the measure log, the start time is recorded as 14:16:23 whereas in the statistics log, it is 12:16:23.

Step	Description					Time			
	Area	Level	Package	Method	Param.Grp	Startdate	Starttime	Endtime	Duration
1	4PERF001	4PERFP00	4PERF001	0-MP	4PERFL04	02.07.2002	14:16:23	14:16:34	21,30
2	4PERF001	4PERFP00	4PERF001	0-MP	4PERFL04	02.07.2002	14:16:58	14:17:03	4,92
3	4PERF001	4PERFS00	4PERF001	4PERFF03	4PERFP02	02.07.2002	14:17:28	14:17:34	6,38
4	4PERF001	4PERFS00	4PERF001	4PERFF01	4PERFP01	02.07.2002	14:17:41	14:17:43	1,51
5	4PERF001	4PERFS00	4PERF001	4PERFF03	4PERFP02	02.07.2002	14:17:48	14:17:48	0,09
6	4PERF001	4PERFS00	4PERF001	4PERFF01	4PERFP01	02.07.2002	14:17:52	14:17:52	0,09
7	4PERF001	4PERFS00	4PERF001	0-MP	4PERFL01	02.07.2002	14:17:55	14:18:02	6,96
8									
9									
10									
11									

Total runtime	41,26
---------------	-------

3.2 Determining the Execution Time for Each Step

The used case is now executed and the execution time is calculated for each step and entered into the overview table. This is simplified by using the dialog response time in the SAPGUI (found at the bottom right of the SAPGUI where the system name and other details are usually displayed). The statistics for SEM/BW/BW-BPS should be activated beforehand. To do this, proceed as follows:

1. Execute transaction BPS_STAT0 and then the selection screen (via F8) to display the detail view.

Report Using UPC_STATISTIC2 - Statistics and Hierarchy In...

UUID 22... **Execute the selection**

Name to

Type to

Current date 02.07.2002 to

Time 12:10:00 to 12:20:00

User FISCHERO to

Planning Area to

Planning level to

Planning Package to

Planning function to

Parameter group to

Total Execution Time to

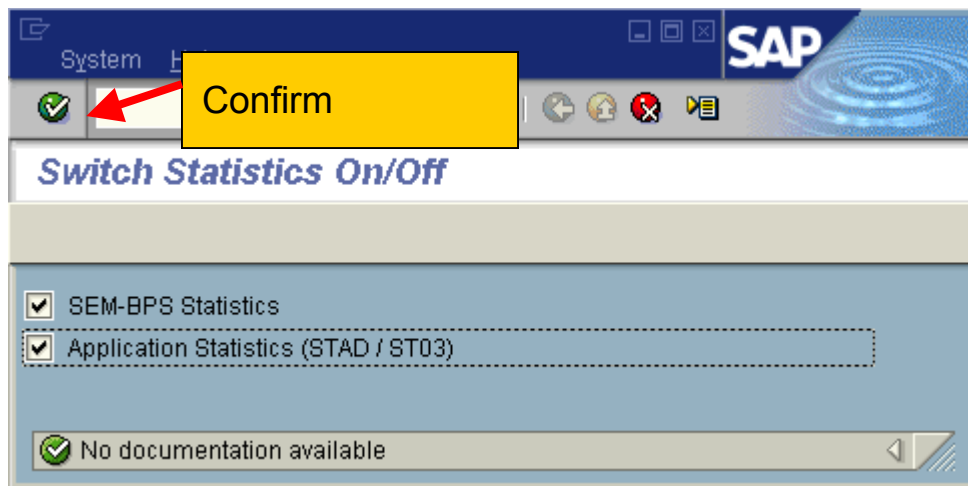
2. Choose the pushbutton with the activation icon (Ctrl+F3).

Statistics SEM-BPS: Single Record Analysis

Activate the statistics

Name	Current date	Time	Pl.package	Pl. func.	Param. grp
MP	02.07.2002	12:16:28	FISCHERO	4PERF001	4PERFL04
MP	02.07.2002	12:16:58	FISCHERO	4PERF001	4PERFL04
PF	02.07.2002	12:17:28	FISCHERO	4PERF001	4PERFF03
PF	02.07.2002	12:17:41	FISCHERO	4PERF001	4PERFF03
PF	02.07.2002	12:17:48	FISCHERO	4PERF001	4PERFF03
PF	02.07.2002	12:17:52	FISCHERO	4PERF001	4PERFF03
MP	02.07.2002	12:17:55	FISCHERO	4PERF001	4PERFL01

3. In the screen that then appears, select the entries for the SEM/BW/BW-BPS statistics and the application statistic and confirm the selection by confirming (choose the pushbutton with the green tick icon).



4. Logon to the system again and perform the steps for the used case. Make a note of the start time and the execution time for the planning step.

Note: To simplify the analysis, you should not work concurrently in the system with the same user name while executing the planning steps. It might otherwise prove difficult to assign the entries from statistics uniquely to the used case.

3.3 Analyzing the Execution Time of a Step: SEM/BW/BW-BPS Statistics

From the technical standpoint, the execution of a planning step comprises a number of different units in sequence. Examples of such units would be database read time, CPU time for processing data records, or data transfer from the back end to the front end.

- The execution of a planning function generally involves reading data from the database (database read time) and then processing the data (CPU time).
- When manual planning is executed, the data, after being read from the database (database read time) and processed (CPU time at the back end), is sent to the front end (data transfer time) and displayed there (transfer and display time).

First, the appropriate entries from the SEM/BW/BW-BPS statistics are assigned to the planning steps. The overview achieved in this way is an essential basis for the subsequent verification or optimization phase because it identifies the units requiring most of the execution time.

Proceed as follows for the assignment:

1. Execute transaction BPS_STAT0. In the selection screen, enter the user name, the date, and the period over which the used case is to be executed. Note that the SEM/BW/BW-BPS statistics records times in GMT standard time and that these times may therefore differ from your local time.

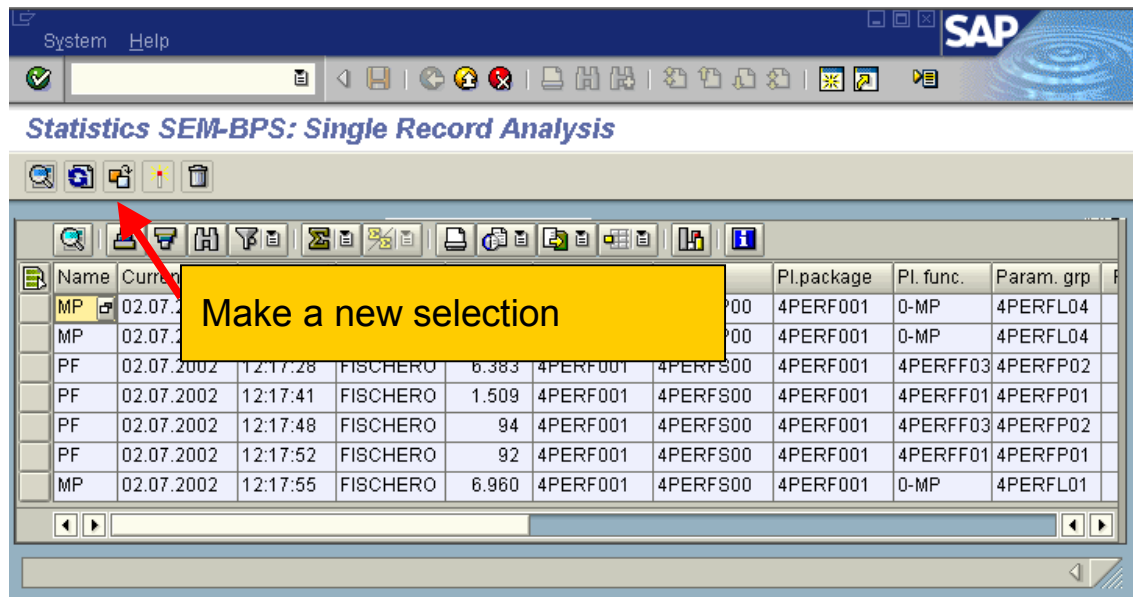
Example: You are working in the time zone GMT+2 and execute a planning function at 14:00. You then have to subtract two hours from this time to obtain the equivalent in GMT (in this case, 12:00).

Recommendation: Save the selection as a variant so that you can call up your settings again later.

2. Sort the displayed records ascending by time.
3. Now check the selection by comparing the technical name and the execution time of the planning steps as they appear in the used case and in the statistics. If you notice in this comparison that some planning steps are missing or that too many are displayed, check the selection that you specified at the beginning. It is likely that the period selected was too small or too large.

Recommendation: An assignment check can also be applied to the runtimes.

Note: The runtimes that you determined and those the were recorded in the statistics may differ slightly. Such discrepancies may occur, for example, because the response time determined using the SAP GUI also contains the communication time between front end and back end.



- Copy all rows and columns to an ExcelSheet for further processing. Note that the standard layout does not display all columns. For this reason, it is important to first display all columns before copying the data. Change the display settings using the *Change Layout* function that you access via *Choose Layout*.

Instead of being copied, the data can also be exported directly into an Excel table. To do this, use the option *Export -> Spreadsheet -> Normal Table*.

In the following section, you find an Excel sheet showing how the gathered information could look. The aspects discussed in the following sections relate to the example data in this Excel sheet. The table display has been altered for the purposes of this document, whereby the lines in the table have been split and the column title font made italic.

The significance of the individual columns is described in the respective F1 Help documentation on each column or in the chapter *Measuring and Details on SEM/BW/BW-BPS Statistics* below.

Step	Name	Type	Date	Time	User	Plng Area	Pl. level	Pl.package	Pl. func.	Param. grp	Runtime	ReadBWRecs	BWReadTime
1	MP	EXCEL	02.07.2002	12:16:23	FISCHERO	4PERF001	4PERFP00	4PERF001	0-MP	4PERFL04	21.383,00	101,00	9.220,00
2	MP	EXCEL	02.07.2002	12:16:58	FISCHERO	4PERF001	4PERFP00	4PERF001	0-MP	4PERFL04	4.923,00	0,00	0,00
3	PF	08	02.07.2002	12:17:28	FISCHERO	4PERF001	4PERFS00	4PERF001	4PERFF03	4PERFP02	6.383,00	201,00	158,00
4	PF	07	02.07.2002	12:17:41	FISCHERO	4PERF001	4PERFS00	4PERF001	4PERFF01	4PERFP01	1.509,00	0,00	0,00
5	PF	08	02.07.2002	12:17:48	FISCHERO	4PERF001	4PERFS00	4PERF001	4PERFF03	4PERFP02	94,00	0,00	0,00
6	PF	07	02.07.2002	12:17:52	FISCHERO	4PERF001	4PERFS00	4PERF001	4PERFF01	4PERFP01	92,00	0,00	0,00
7	MP	EXCEL	02.07.2002	12:17:55	FISCHERO	4PERF001	4PERFS00	4PERF001	0-MP	4PERFL01	6.960,00	0,00	0,00

Step	Name	PF Read R.	PF R.Chngd	PF: R. New	MP Trans	MP Start	MP dataput	BW ZeroRec	BW Calls	PF Ref.Dat	MPNo.Cells	MP LogSc	UUID 22 character
1	MP	0,00	0,00	0,00	670,00	3.507,00	676,00	101,00	1,00	0,00	183,00	16.729,00	0002X5BFm
2	MP	0,00	0,00	0,00	228,00	3.936,00	242,00	0,00	0,00	0,00	63,00	615,00	0002X5BFm
3	PF	100,00	100,00	0,00	0,00	0,00	0,00	101,00	1,00	0,00	0,00	0,00	0002X5BFm
4	PF	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0002X5BFm
5	PF	100,00	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0002X5BFm
6	PF	100,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0002X5BFm
7	MP	0,00	0,00	0,00	239,00	4.457,00	328,00	0,00	0,00	0,00	375,00	1.911,00	0002X5BFm



Statistik-extract.xls

3.4 Analyzing the Execution Time for Each Step: Application Statistics

The application statistic data that can be analyzed using transactions STAD (analysis of individual records) or ST03 describe the planning step from the point of view of the operating system. In this way, the statistics obtain information such as how many CPU seconds were required or how long the work process took. Users wanting to use this information in an analysis need to have experience in performance optimization. Such information falls outside of the scope of this guide.

If you want to perform an ordinary analysis aimed at optimizing SEM/BW/BW-BPS Customizing, you generally do not need the information contained in this section and can continue with chapter 4.

The following step-by-step guide shows how the entries from the application statistics can be assigned to the planning steps from SEM/BW/BW-BPS statistics.

1. Start transaction STAD and make the selections as instructed in the comment fields in the following screenshot:

The screenshot shows the SAP STAD (Statistical Records) transaction screen. The title bar indicates 'System Help' and 'SAP'. The main title is 'Statistical Records'. Below the title, there is a section 'Select statistical records (for the whole R/3 System)'. Under this, there is a 'Select the display mode:' section with three radio buttons: 'Show all statistic records, sorted by start time' (selected), 'Show all records, grouped by business transaction', and 'Show business transaction summ'. Below this, there are input fields for 'Start date' (02.07.2002), 'Read time' (00:10:00), 'Start time' (14:10:00), 'User' (fischero), 'Transaction' (*), 'Program' (*), 'Task type' (*), 'Resp. time' (>=), 'DB req.time' (>=), 'CPU time' (>=), 'Bytes req.' (>=), and 'DB changes' (>=). At the bottom, there is a 'Tools' section with two checkboxes: 'Include statistics from memory' (checked) and 'Include application statistics' (checked). There are also buttons for 'Server selection' and 'Further options'. Three yellow callout boxes with red arrows point to specific fields: '1. Start time and date' points to the 'Start time' field, '2. Time interval in which the statistical data is to be displayed' points to the 'Read time' field, and '3. Set indicator so that the system also reads the application information' points to the 'Include application statistics' checkbox.

1. Start time and date

2. Time interval in which the statistical data is to be displayed

3. Set indicator so that the system also reads the application information

2. The entries highlighted with colors correspond to the planning steps and each entry describes the complete execution.

Workload: Statistical Records

System: QE5 Number of RFCs which responded (without errors): 4 (4)
 Analysed time: 02.07.2002 / 14:16:23 - 02.07.2002 / 14:26:23 Time frame: +/- 00:02:00
 Display mode: All statistic records, sorted by time Application statistic is included

Started	Server	Transaction	Program Function	T Scr. Wp	User	Response time (ms)	Time in WPs (ms)	Wait time (ms)
		BPS0	U*	D	FISCHERO	0		
14:16:23	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 1	FISCHERO	12.649	12.626	
14:16:23	1s0319_QE5_24	BPS0	SBSEM_BPS_MP_EXEC	D 0100 1	FISCHERO	21.398	21.386	1
14:16:40	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 1	FISCHERO	4.315	326	
14:16:48	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 0	FISCHERO	34	19	
14:16:50	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 0	FISCHERO	971	864	
14:16:52	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:16:54	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:16:54	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:16:55	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:16:58	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:16:58	1s0319_QE5_24	BPS0	SBSEM_BPS_MP_EXEC	D 0100 2	FISCHERO	4.931	4.924	
14:17:13	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	101	31	
14:17:18	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	309	121	
14:17:23	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:17:24	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:17:26	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:17:28	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:17:28	1s0319_QE5_24	BPS0	SBSEM_BPS_PF_EXEC					
14:17:39	1s0319_QE5_24	BPS0	UPC_FW_CALL					
14:17:41	1s0319_QE5_24	BPS0	SBSEM_BPS_PF_EXEC	D 0100 2	FISCHERO	1.513	1.507	
14:17:41	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	1.828	1.701	
14:17:48	1s0319_QE5_24	BPS0	SBSEM_BPS_PF_EXEC	D 0100 2	FISCHERO	94	94	
14:17:48	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	381	252	
14:17:52	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	403	273	
14:17:52	1s0319_QE5_24	BPS0	SBSEM_BPS_PF_EXEC	D 0100 2	FISCHERO	92	92	
14:17:54	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	102	47	
14:17:55	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	7.031	2.501	
14:17:55	1s0319_QE5_24	BPS0	SBSEM_BPS_MP_EXEC	D 0100 2	FISCHERO	6.970	6.961	
14:18:13	1s0319_QE5_24	BPS0	UPC_FW_CALL	D 0100 2	FISCHERO	1	1	

3. There are two ways in which the planning steps can be assigned. On the one hand, the start time from the application statistics (shown in the first column) can be compared with that in the SEM/BW/BW-BPS statistics. This is simplified by the fact that the program name (in the third column) contains the type of the planning step. Here, MP stands for manual planning, PF for planning function, and so forth. The program name is created using the following logic: SB + SEM/BW_BPS_ + (type of planning step) + EXEC.

Example: In the case of manual planning, the program name is therefore SBSEM/BW_BPS_MP_EXEC.

4. The other variant relies on a unique key (GUID) being quoted with the matching entries for the SEM/BW-BPS statistics and the application statistics.

You find this GUID – along with other application information - in the dialog box that you can call up simply by clicking on the program name.

Complete Description of a Function

Text: SBSEM_BPS_MP_EXEC

Tech Text: SBSEM_BPS_MP_EXEC

Function type: SB

Application Info: Details with BPS_STAT0:188SFOnY4XdX00002X5BFm

Init Key: IIBPS0

Open Key: SBSEM_BPS_MP_EXEC

GUID that creates the link to the SEM/BW-BPS statistics

Enter this GUID in the selection screen for the SEM/BW-BPS statistics (transaction BPS_STAT0). The relevant data record is then displayed.

The screenshot shows the SAP selection screen for the report 'UPC_STATISTIC2'. The title bar indicates 'Report Using UPC_STATISTIC2 - Statistics and Hierarchy In...'. The selection criteria are as follows:

Field	Value	Operator	Field	Value
UUID 22 character				
Name		to		
Type		to		
Current date	02.07.2002	to		
Time	12:10:00			
User	FISCHERO			
Planning Area				
Planning level		to		
Planning Package		to		
Planning function		to		
Parameter group		to		
Total Execution Time		to		

4 Assessment of Analysis Results

4.1 Optimization

After the execution time is now available at a very detailed level, it can easily be determined in which units this is mainly spent. As a result, the prerequisite is created to begin with the optimization work target-related, in which these units will be edited.

You will find different information that you need for optimization in the following section. That is general recommendations and notes for the areas planning function, manual planning and database access on the one hand, and background information on the effect executing the planning steps has in the system, and how the Customizing of planning functions or manual planning influences this effect on the other hand. It is often sufficient to follow the recommendations and notes for a first optimization. However, for more extensive processing, it is essential to include the background information as only then can customer-specific features be taken account.

As a rule, the recommendations and notes almost automatically arise from the background information so that it is a good test of understanding to explain these with the background information.

5 Background Knowledge

The following subsections describe from the system's view what happens when a planning function or manual planning is executed. The information is not necessary to apply the optimization proposals which are presented in section 6. However, the information helps to understand how the recommendations work and its starting points. It also enables you to make further special optimizations of your Customizing on the basis of a deeper understanding of the processes.

You will find information on the area of memory and CPU consumption, which are independent of the type of planning step executed, summarized in extra subsections.

5.1 Planning Functions

We assume that the transaction BPS0 was started again and you have navigated to the planning function that you want to execute. What happens now during the first execution?

The selection, which is defined in the planning level and the planning package, is used to read the data records from SAP BW that should be changed. Conditions which are available in the Customizing of planning functions are not taken into account here.

Now it is determined whether the planning functions require further data and if necessary this is read from BW. This is generally reference data which is not part of the level or package selection. The data read in this way is not locked exclusively for the user, since only read access is required.

Example: With the copy function, data records that are specified in the 'copy from' fields and are not contained in the package selection are read from SAP BW.

All read data is written into an internal data buffer directly after the read access, which is in the memory area of the application server. If the function is executed a second time, the request would then be served directly from this buffer without having to read data from BW.

Only then does the actual planning function start its work. This means that it checks all data records against the conditions and then changes, generates or deletes the desired data records. Each of these changes is written back into the buffer and is then available for subsequent calculations or for display in manual planning. (When saving new or changed data records again, the exits 'Characteristic derivation' and 'Combination validation' are called up to derive or validate them – see SAP Note 387524. Furthermore, the system checks whether these records are locked by data slices).

The following illustrations show the order of steps explained in the text above. The greyed read accesses show that the transaction data is read from the buffer and not from the BW.

Initial Start of a planning function:



Re-execute a planning function:



5.2 Manual Planning

We assume that transaction BPS0 was started again and you have navigated to the layout that you want to execute. What happens now during the first execution?

First the transaction data is read from BW. However, the selection used there generally does not correspond exactly to the package selection, but is created according to the following logic:

1. The package selection is used for characteristics which are in the header of the layout.
2. The values defined (in the layout) are used for characteristics which are in the data columns (even if the indicator 'Comparison column' is set for the column!).
3. The logic depends on the definition of the key area for characteristics in the key area. In case the rows are defined individually, the values defined in the layout are used (also with 'Comparison columns'). In case the rows to be displayed are derived from the existing transaction data, the package selection is used.

On the assumption that the call of manual planning takes place in the change mode, the selected data is locked exclusively for the user and written to a data buffer. A further component (planning processor) now

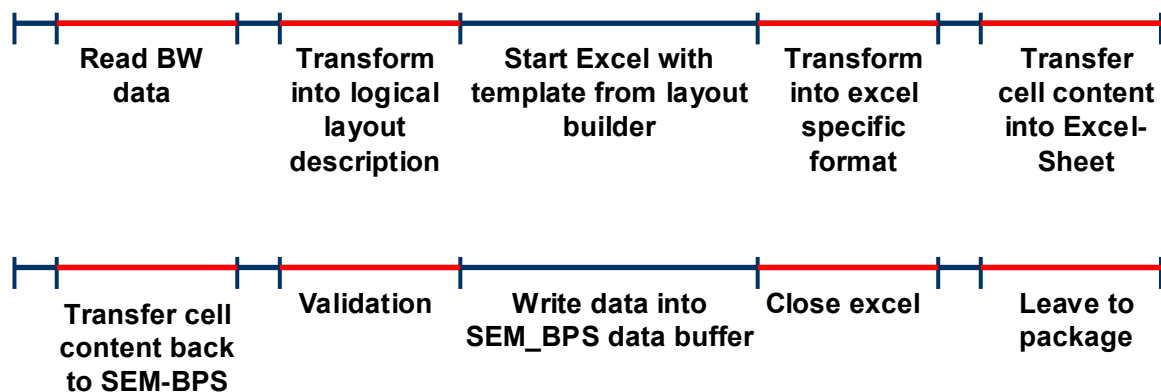
prepares the data which is first available in a simple table, in accordance with the definition of the planning layouts and presents the result in the front end.

This process covers the following steps:

1. The data from the data buffer is converted into a convenient form for further processing and stored there for the duration of usage of manual planning.
2. The first header combination is determined. The transaction data which belongs to this header combination generally only presents a partial quantity of the total requested transaction data. A further conversion is made for this partial quantity into an internal (logical) presentation of the screen content (logical screen).
3. The display tool at the front end is started and the layout template is opened in the case of Excel.
4. A new format is generated from the internal (logical) presentation of the screen content, which was generated in step 2 and is required for the first screen layout. This new format is required for the corresponding display tool (Excel/ALV). With Excel the possibility exists to fill further customer-specific data cells with an exit.
5. Now the data cells are sent to the front end and transferred into the display tool.
6. Provided that the automatic execution of Visual Basic macros has been activated in the layout, these are now executed.

The following illustration shows the order of steps executed by the system if a manual planning is started (first line) and a planning package is chosen after some values have been changed. (second line).

Initial start:



For the duration of manual planning, the display tools (Excel/ALV) only communicate with the planning processor using the internal general display of the layout (logical screen). Examples of this communication are changing the header combination or checking newly entered values.

After exiting manual planning (for example, by opening another layout), the data is written again into the data buffer and the memory space required by the planning processor is released. In this way, the planning processor only stores the data of the active layout, while the display tools only store the data of the current logical screen.

The data of the logical screen is transferred to the front end PC after formatting for the respective front end using SAPGUI-RFC. In the case of Excel in Place, the data is then in a front end component that is delivered with the SAPGUI. This data is then transferred into MS Excel using OLE automation calls. Transferring the data in this last step is quicker, the quicker the front end PC is. If very large datasets are transferred, then the memory available in the PC is also relevant. Therefore, if there is not sufficient working memory on the PC, areas of this are stored on the hard disk, and thus make the execution time worse.

Further Details on Point 1:

You can imagine the result of the selection of transaction data in the data buffer as a simple table. The entries of this table contain exactly the characteristics in the key and exactly the key figures in the data part that are available in the planning area. Further tables are filled for the duration of manual planning, starting from this table.

One of these tables (table 1 or gr_c->to_dat1) contains, for example, exactly the characteristic combinations of the selected data in the key area, and another table (table 2 or gr_c->to_dat2) all characteristic combinations in combination with the key figures occurring in the planning layout. The size of these tables results according to the following formulas:

- Table 1: Number of entries = number of records in the data buffer
- Table 2: Number of entries = number of entries from table 1 * number of used key figures in the layout.

Both of these tables were presented as they are generally the largest tables after the transaction data buffers and therefore should also be taken into account when reducing the working memory or the CPU requirement. But since these tables are only stored for a layout (in contrast to the transaction data buffers which contain all data read in the planning session), they are generally much smaller than the transaction data buffer.

Further Details on Point 2:

A logical screen is built for a header combination. This contains exactly the data displayed in the display tools. The logical screen refers in this way to exactly the part of table 1, whose entries belong to the current header combination and the part of table 2, whose entries belong to the used header combination and the used key figures (an entry belongs to it exactly if the characteristic values of the header combination agree with those of the table entries). It depends on the form of the planning layout whether a row or even a cell in the layout corresponds to a table entry in table 1 (per fixed header combination). At any rate every cell in the layout corresponds to an entry in table 2.

Examples:

Selected characteristics: Version, fiscal year, product;

Selected key figures: Sales quantity, revenue

Available in the package: Three versions, two fiscal years and five products.

Data is available for all combinations.

Layout1:

Version, fiscal year in the header, therefore $3 * 2 = 6$ header combinations, as well as 6 possible logical screens; product in the key and sales quantity and revenue in the data columns. Therefore, you have 5 rows; one row corresponds to an entry in table 1 per header combination; altogether there are $6 * 5 = 30$ entries in table 1 and $30 * 2 = 60$ entries in table 2.

Version (3 characteristic values)
Fiscal year (2 characteristic values)

	Sales quantity (key figure)	Revenue (key figure)
Product 1		
Product 2		
Product 3		
Product 4		
Product 5		

Layout2: Version in the header, therefore, there are 3 header combinations, thus 3 possible logical screens; product in the key and the combination of two fiscal years with the two key figures in the data columns. Thus you have 5 rows; one row corresponds to two entries in table 1 per header combination; altogether there are $3 * 5 * 2 = 30$ entries in table 1 and $30 * 2 = 60$ entries in table 2.

Version (3 characteristic values)

	Sales / 2000 (key figure / fiscal year)	Sales / 2001 (key figure / fiscal year)	Revenue / 2000 (key figure / fiscal year)	Revenue / 2001 (key figure / fiscal year)
Product 1				
Product 2				

Product 3				
Product 4				
Product 5				

5.3 WEB

Business Server Pages (BSP) are generated on the basis of the Customizing settings made in the WEB Interface Builder (transaction BPS_WB). The pages generated can be displayed and where necessary modified in the Object Navigator (transaction SE80 => Repository Browser => BSP Applications). Such changes are overwritten during repeated generation.

The runtime of the WEB Interface Builder uses planning objects that have been created in the Planning Workbench (transaction BPS0). If, for example, you now make changes to a planning function (without changing its name), it is not necessary to regenerate the WEB application because the planning objects are referenced only.

Communication between the WEB application and the planning framework occurs almost exclusively via the SEM/BW-BPS API. One exception to this is when you use Excel WEB components instead.

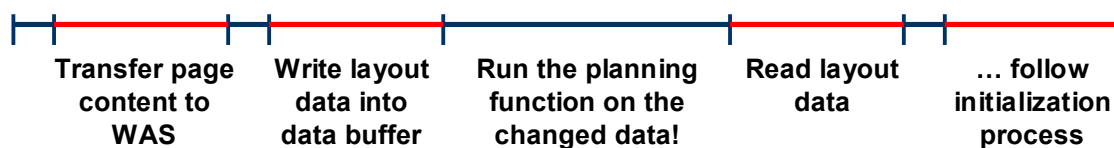
A WEB application contains different objects in which values can be changed. Examples of such objects are selectors or entry fields for variables or layouts. If, for example, a planning function is to be triggered, you need to establish the order in which the changes made are to be transferred to SEM/BW-BPS. With the default sequence, the layout information is written first of all to SEM/BW-BPS, and then the variable values, and finally the planning function is executed.

The following illustration shows a typical data flow of a first call of a Web Planning Application and the execution of a planning function.

Calling the URL leads to an initialization process ...



Input processing (execution of function):



For more information about the WEB Application Server (SAP WAS), see the documentation in the Application Help.

5.4 Memory Consumption on the Application Server

The total memory consumption is made up of a static and a variable part. The static part essentially consists of starting cross-application planning (transaction BPS0) and covers approx. 3 to 5 MB. The variable part is mostly determined by the number of data records, which are read from BW and/or have been generated in a planning function. The rule also applies here that it is advantageous to keep the number of selected or generated records as small as possible.

The buffer, which manages the transaction data (and covers the major part of the working memory), is only released when you have exited the transaction. This generally means that when using new data the memory occupied increases (accumulative effect). In general it makes sense to save the results after ending a planning section, and then to start the transaction again afterwards.

The (theoretically) minimal memory consumption can be estimated by determining the number of records read (transaction and reference data) and multiplying these by the technical length (see SAP note 0407563 for details on this). This number must still be multiplied by the factor $1 + 1 + 1 + 0.5 = 3.5$. This factor results because when executing a planning function this data must be stored several times. The individual summands are explained below:

- A database copy at the time of the selection from SAP BW is stored in the data buffer.
- The planning function or manual planning executes changes to a copy of the transaction data. This is necessary since if the function could not be executed without any errors, the initial data cannot be changed.
- The results will be stored as delta to the database copy. This is necessary since when saving transaction data in SAP BW, the changes to the previous value (delta logic) must be entered. On the assumption that all data was changed, we receive a share of 1. If only half of it was changed, then we would receive a share of 0.5.
- To quickly access the data in the data buffer, index tables are created that contain the key part of the database table. On the assumption that the length of the key part roughly corresponds to half of the data record length, we still receive a share of 0.5.

5.5 CPU Consumption

Executing a planning function cannot be distributed to several CPUs. From the point of view of a single end user, the optimal runtime (this is achieved when a user has a CPU available at 100%) depends on the calculation speed of a CPU, not on the total number of CPUs of the system. This becomes important when many users are working on the system simultaneously.

The CPU time required generally depends on the complexity of the planning function, on the number of data records edited, on the use and number of data slices, and on the use or implementation of derivation or combination exits.

6 Optimization Proposals

6.1 Planning Modell

In BPS you can use multi areas to combine several basis areas. This is definitely needed if data should be transferred or access from several basis areas by one function. When creating planning functions accessing only one basis area they should be defined in this basis area and not in the multi area creating a certain overhead. This leads on the one hand to longer execution times (e.g. mapping between the data structure of the multi and the basis area) and on the other hand to a higher memory consumption as the data structure of the multi area contains all characteristics and keyfigures that are part of any basis area (note that the structure is independent from the setting of the planning level).

6.2 Planning Functions

6.2.1 General Recommendations

Executing specialized functions is generally quicker than executing more general functions (for example, using the copy function instead of the formula function to copy data).

The data records that have been changed by a planning function are checked for their validity before they are written into the data buffer (for example, a check regarding data slices, combination validation) and possibly derived. These activities increase the execution time of the planning function. Pay particular

attention to a high-performing implementation when using combination validation and check, since the ABAP coding will run through once per data record.

In order to estimate the execution time that will be used in the exits, the function should be executed twice, once with the exit activated, once with the exit deactivated.

Assuming that you always execute three predefined planning functions one after the other, which edit the same data records. This then means that from the system's view, the data record was changed three times and therefore the validity was checked three times. Combining specialized functions in one general function reduces this number and as a result the time that has to be spent on this check. The performance disadvantage of a general function can possibly be more than compensated for by the lower number of required checks and subpackages. One disadvantage of this optimization is, for example, a lower clarity of Customizing and that the functions can only be executed en bloc.

Example: A copy function, a revaluation, and a formula function edit the same transaction data and are always executed one after the other. As a rule, combining them into one formula function will reduce the execution time.

The larger the number of data records transferred to the planning function for editing, the higher the required CPU time is. Data records that have been excluded through conditions, increase the required CPU time as every record must be checked first regarding the conditions. A comparison of the values for 'read records' with 'changed records' and 'regenerated records' (these values are displayed in the status line after executing a planning function) delivers a clue to the 'effectiveness' of the function. Note that at present reference data is not displayed in the status line and, therefore, you have to look in the SEM/BW-BPS statistics to determine this number.

Example: A revaluation function has read 1500 records and changed 100. This is a poor relationship as 1400 records have been checked unnecessarily for the conditions as well as read by BW. In this example it would be appropriate to enter the selection in the package or the level so that (ideally) only 100 records would have been read.

6.2.2 Exit Function

The aspect of execution speed should be taken into account during implementation. You will find a collection of examples in the following section that explain how a certain task could be implemented so that it is high-performing.

1. With the loop over the transaction data table xth_data, the line structure and the individual components of the line structure must be dynamically referenced using the assign command. The assign can take place inside or outside of the loop – endloop command. The implementation mentioned last is more high-performing as a rule. Outside means that the assign commands are applied once to the line structure. If this line structure is now refilled within the loop command, the system will automatically update the assigns.

```
[...]
field-symbols <ls_data>.
create data lr_data like line of xth_data.
assign lr_data->* to <ls_data>.

[...]
assign component:
'S_CHAS-0FISCYEAR' of structure <ls_data> to <0FISCYEAR>,
'S_CHAS-0VERSION' of structure <ls_data> to <0VERSION>.

[...]
Loop at xth_data into <ls_data>.
[.do something with 0Version and 0Fiscyear..]
Endloop.
```

2. When reading an individual data record from the table xth_data, note that it concerns a table of the type hashed. For this reason, it should always be accessed with a complete key. If this does not happen then the hash algorithm cannot be used and the system automatically chooses sequential reading as an alternative, which as a rule leads to a considerably longer runtime.

3. When reading database tables in a planning function of the type exit, note that as a rule with one execution (user action) the planning function calls up the function module that contains the source text you created very often (in extreme cases once per data record). The number of calls is influenced by the selection in the characteristics in the table 'fields to be changed'. For this reason, it can make sense to implement the initialisation module of the planning function, as this is always called up and only once. There are examples in the function group upfx.
4. In the initialization module you can perform time consuming steps that has to be done exactly once and that doesn't change the transaction data itself. To transfer information or data from the initialization to the exit function module you could define them in the same function group and use global tables defined in the top include.

6.2.3 Formula Function

Skilful formulation of the algorithm can save CPU time with the formulas as with a programming language. There are a series of recommendations in the examples for formula language (F1 help in the formula editor).

Below is a summary of the most important information.

1. If you have to access an operand several times, it makes sense to note the value of this operand in a variable and to continue working with this variable.
2. Only include the characteristics from the planning level in the quantity of characteristics to be changed if the characteristics are really edited, or these are needed to select reference data (see the example of price planning in the F1 documentation).

If you need characteristic values of characteristics that are not part of the characteristics to be changed, you can use the OBJV() function (see the example of price planning in the F1 documentation).

3. Weigh up whether it is necessary to nest FOREACH loops or whether it is possible to use constructs of the type FOREACH VAR1, VAR2. The difference appears as follows in the formula:

a.) nested FOREACH loops

```
FOREACH VAR1
  FOREACH VAR1
    [... calculation ...]
  ENDFOR
ENDFOR
```

b.)

```
FOREACH VAR1, VAR2
  [... calculation ...]
ENDFOR
```

The difference between both variants is that in the first case (nesting) all values for Var1 and Var2 are determined from the transaction data and then all combinations are formed for the calculation. In the second case, all combinations of Var1 and Var2 are determined from transaction data. The difference is explained once more in the following example. Only two or three characteristic values have been included here for reasons of clarity.

Example: Assuming you use two characteristics each with 3 and 2 characteristic values. If the two FOREACH loops are nested, then the outer loop will run through the inner loop twice for every step, and thus results in $3 \times 2 = 6$ program runs. This makes sense if all combinations are required. When using FOREACH VAR1, VAR2, the number depends on the available combinations and thus results in 3 program runs, and as a result in a considerably lower number than in the first case.

Transaction data record	Characteristic 1	Characteristic 2
1	A	X

2	B	X
3	C	Z

Combinations with nesting	Characteristic 1	Characteristic 2
1	A	X
2	A	Z
3	B	X
4	B	Z
5	C	X
6	C	Z

Combinations without nesting	Characteristic 1	Characteristic 2
1	A	X
2	B	X
3	C	Z

- In general, reference data is read to evaluate formulas. The operands and expressions are examined that are on the right side of the assignment indicator '=' in a formula to determine the selection condition for the reference data. If this concerns characteristic values, then it is checked whether these are included in the selection condition of the package. Is this not the case, then the characteristic value is added to the selection condition. The selection condition for the characteristic value is deleted, if the variable was set using a function TMVL (generate characteristic values of a time characteristic) or ATRV (read attribute values).

The selection condition of the package is only adjusted for the characteristics, which have been selected as characteristics to be changed of the function.

6.3 Manual Planning

The package selection is to be set up for the header characteristics of the planning layout, so that you receive exactly the combination that a user really edits in the planning session. As a result, you can avoid that many records or cells that are not needed are hidden behind header combinations that are not used, and thus take up unnecessary system resources. See the section Background Knowledge => Manual Planning for an explanation about how the number of records can be determined that are hidden behind a header combination.

Using data slices and combination validations increases the execution time as each data cell must be checked whether it is ready for input (data slices) or allowed (combination). Consider that the number of data cells on the front end can differ greatly from the number of data records.

Example: You use the form-based mode that proposes possible data records with the help of existing master data. Several hundred data records can be made available for display without a single transaction data record existing. Every data record that exists for such a data record will be checked!

A slow front end PC can greatly increase the response time as CPU time is needed to form the planning layout.

The Excel template into which the transaction data is imported and which, for example, contains the style information and graphics in a specific layout, is transferred to the front end PC the first time the planning layout is called up and saved locally. Every further call of a planning layout then uses the local Excel template, provided this is still valid, and has not been changed on the central application server.

As the size of the Excel template influences the transfer time needed you should try to reduce the size as much as possible. Here is a list of possible reasons that might lead to an astonishing high size:

- Graphics with high resolution.
- Large Visual Basic coding.
- Cells excel has marked as 'in use' although they are empty.
- Undo and history information from working in the excel sheet or in the VBA editor.

Solutions for the first two items are to reduce the resolution to needs (e.g. a digital photo provides much more details than you can on the screen.) and reduce of the transferred coding (e.g. use Add-ins). For the last two points Microsoft provides a detailed solution description on prerequisites and how to decrease the size:

- <http://support.microsoft.com/default.aspx?scid=kb;en-us;q123269>
- <http://support.microsoft.com/default.aspx?scid=kb;en-us;q280157>

One way of determining the current size is to enter the third screen of the layout builder and save the file to your local PC. In the properties of this file you will find the actual size.

The network speed is important; with normal LANs the influence is not relevant. With a WAN, which can usually transfer a much lower quantity of data per second, the network can greatly reduce the performance (for example, a modem connection, ISDN). In this case, you should restrict the quantity of the data contained in the layout to a sensible size.

Several round trips are necessary (transferring data, querying an attribute on the front end, reaction to this on the back end, setting an attribute on the front end etc.) during data transfer from the back end to the front end and back. Every round trip has a certain latency; in WAN this latency can also reduce performance. In certain cases, it can make sense to switch off the progress indicators sent by the system using the SET/GET parameter 'SIN' in order to reduce the number of roundtrips. To do this, this value must be set to '0' in the user settings.

6.4 BW

If you use an InfoCube for planning, it must be a transactional InfoCube.

To achieve optimum read performance, the InfoCube should be compressed and the InfoCube statistics updated regularly. In the context of compression, "regularly" means often enough to ensure that no more than five transactional requests are left uncompressed at any one time.

When creating and updating the statistics for transactional InfoCubes, apply the same procedure as for normal InfoCubes. Ensure for this that the F fact table is not empty when the statistics are created. This requirement is linked directly to the fact that the Optimizer is more sensitive to out-of-date statistics if too few values are available. Moreover, especially when filling empty InfoCubes, updates should occur more frequently because the number of values changes dramatically when expressed as a percentage. Independently of the settings in the Administrator Workbench, the data is rolled up into any existing aggregates each time a request is closed.

The navigation path is as follows: Start transaction RSA1 => Data Targets => Select InfoCube => Manage. Choose the "Performance" tab page. The different activities can be started from this tab page. For more extensive measures, consult an SAP BW consultant or a database expert.

When uploading data into BW the system can perform specific actions before and after the load process. A common action is to drop the index before and to rebuild it after this process. This must not be done if a transactional InfoCube is used for planning as the close of a transactional request is treated as a data load process. The read performance will be significantly lower if no index exists.

The creation of aggregates can improve the read performance. In the SAP Note 560369 you will find a report that shows proposals for aggregates based on existing planning levels. The report will be delivered with SEM/BW 3.1B support packages 6.

From BW 3.0B SP 15 onwards the system will consider the flag 'automatic roll up' and you can decide whether the roll up should take place automatically when a request is closed or not.

To further analyse the execution of a BW data read access you can use the report upc_bw_selection_debug. It offers a subset of the RSRT debug flag set like aggregates found, execution plan etc.

6.5 WEB

A WEB application generally consists of a large number of interface elements such as layouts, planning functions, and variable selectors. These objects are closely linked with one another. Depending on the influence that the individual element has, these links can complicate the analysis. Hence, the response time for the selection of one and the same variable depends on the WEB application used and this can consequently generate misleading statements such as “The selection of the variable has differing response times” even though the time differences are due to the environment settings. To exclude side effects of this nature, only the interface element under examination should be used in the application where possible.

Example: The application consists of a layout and a planning function. The execution time that the end-user sees after choosing the pushbuttons is made up of not just the execution time of the planning function itself but also that for transporting the layout content to SEM/BW-BPS (including all validations) and that for transporting the changed data back to the front-end. The response time is therefore less in applications that only contain the pushbutton for the planning function.

If layouts are distributed across several tab pages, only those layouts that are visible are processed. This reduces the runtime for executing or refreshing the function.

To keep the number of page updates to an absolute minimum, only the necessary events should be activated. This is particularly essential for variable selection.

Example: The layout used should have its header specified by three variable selectors. If the triggering of an event is activated in the attributes for the selectors, the whole page is updated after each selection. In this case, it is preferable to deactivate the events and add a “Refresh” pushbutton. This means that the page is only updated once upon completion of the entire selection.

When accessing WEB planning applications there should be a load balancing process in place. If you like to use the web dispatcher or the standard load balancing please keep in mind that the URL displayed in the WEB builder does not reflect any settings but contains the id of the application server you are currently working on.

End user adding the URL after the redirection process to their favourites will bypass the load balancing process and will make effecting load balancing more difficult.

6.6 Variables

Part of the total execution time is the resolving of variable values. They are a generic service that is normally called several times during an execution.

When using a variable with replacement type exit you should buffer the return values if possible. This saves the time to prepare the result values. Examples can be found in the how to paper: ‘How to ... Variables of type exit’. Variables of other replacement types are in general buffered by the system automatically. With SEM/BW 3.1B SP14 this buffering has been completed to cover all types (refer to SAP note 629775 for more details). Additionally the variables values can be buffered with in each execution of a function. It can be switched on by adding the following line to the data base table upc_dark2: upf_var_buffer_on = X.

The usage of the replacement type authorization could lead to very large selection criterias. As the BPS doesn't support the usage of wild cards or exclusion, the system will resolve the selection defined in the authorization object into a set of single values. Hierarchy nodes or super user permission are good candidates leading to a large set (e.g. ‘*’ for costcenter will lead to a complete list of cost centers defined in the master data). The effects on runtime are:

- Time to resolve the selections based on master data
- Read data from BW (large sql statements)
- Number of entries on the enqueue server
- Time to perform locking checks
- ...

Possible solutions for this area of problems: chose the replacement type exit and implement your own authorization lockup. Add a check that only resolves the selections into single values if they contain wild cards.

Examples and more detailed information can be found in the how to paper: 'How to ... Variables of type exit'.

7 Measuring and Details on SEM/BW/BW-BPS Statistics

Once transaction BPS_STAT0 has been started and the selection has been executed, an overview of the executed planning steps appears on the left-hand side. When you double-click the line or choose the "Details" pushbutton, all details concerning this record are displayed hierarchically in the right-hand side of the screen. This hierarchy generally contains more information that the record displayed in the overview. What the individual hierarchy nodes stand for is described in the table below.

The screenshot shows the SAP BPS_STAT0 interface. On the left, a table lists planning steps. A yellow box labeled '1. Select the record' points to the first row (MP, 02.07.2002, 12:16:58, FISCHERO, 4.923). A second yellow box labeled '2. Choose "Details" or double-click the selected record' points to the 'Details' button in the toolbar. A third yellow box labeled '3. Details view for the selected record' points to the right-hand pane, which displays a hierarchical tree of attributes for the selected record.

Na...	Tagesdatum	Uhrzeit	Benutzer	Laufzeit	Pl.gebiet	Pl.ebene	Pl.paket	Pl.fkt.
MP	02.07.2002	12:16:58	FISCHERO	4.923	4PERF001	4PERFP...	4PERF001	0-MP
PF	02.07.2002	12:17:28	FISCHERO	6.383	4PERF001	4PERFS...	4PERF001	4PERFF03
PF	02.07.2002	12:17:41	FISCHERO	1.509	4PERF001	4PERFS...	4PERF001	4PERFF01
PF	02.07.2002	12:17:48	FISCHERO	94	4PERF001	4PERFS...	4PERF001	4PERFF03
PF	02.07.2002	12:17:52	FISCHERO	92	4PERF001	4PERFS...	4PERF001	4PERFF01
MP	02.07.2002	12:17:55	FISCHERO	6.960	4PERF001	4PERFS...	4PERF001	0-MP
MP	02.07.2002	12:16:23	FISCHERO	21.383	4PERF001	4PERFP...	4PERF001	0-MP

The details view on the right shows a hierarchy of attributes for the selected record (MP, 02.07.2002, 12:16:23, FISCHERO, 21.383, 4PERF001, 4PERFP..., 4PERF001, 0-MP):

- MP
 - P_AREA: 4PERF001
 - T_STAMP: 20020702:121623:671
 - P_LEVEL: 4PERFP00
 - P_PACKAGE: 4PERF001
 - P_METHOD: 0-MP
 - P_PARAM: 4PERFL04
 - T_BRUTTO: 21383
 - T_NETTO: 471
- MP_LOG_SCR
 - T_STAMP: 20020702:121624:117
 - T_BRUTTO: 16729
 - T_NETTO: 7509
- BW_DATA_READ
 - T_STAMP: 20020702:121626:977
 - P_AREA: 4PERF001
 - TYPE: E
 - RECS_READ: 101
 - RECS_DEL: 101
 - T_BRUTTO: 9220
 - T_NETTO: 9220
- MP_START_VIEWER
 - TYPE: EXCEL
 - T_STAMP: 20020702:121640:864
 - T_BRUTTO: 3507
 - T_NETTO: 3507
- MP_DATA_PUT
 - T_STAMP: 20020702:121644:377
 - T_BRUTTO: 676

Generally, a node is represented by a yellow folder icon and the detailed information for the node (attributes) is represented by a sheet icon .

Here is a list of the node names used and what they stand for:

Explanation

MP	Manual Planning
PF	Planning Function
API	Callup of the SEM/BW/BW-BPS API
EXT	Other Callups
WEB	WEB Application

Manual Planning

MP_DATA_PUT	This is the total time for converting the data from the logical description into a suitable format for the display tool and for transporting or displaying the data on the front-end PC.
MP_START_VIEWER	Time the Viewers (Excel) is started on the front-end PC
MP_LOG_SCR	Conversion of the database table into a logical description of the layout
MP_DATA_TO_FRONT	This is the total time for transporting the cell content to the front-end PC and for transferring data to the local instance of MS Excel.
MP_ABAP_EXIT	ABAP Exit for filling the Excel cells with data.

Business Information Warehouse (SAP BW)

BW_DATA_READ	Read access to SAP BW
--------------	-----------------------

WEB Applications

WEB_INPUT_PROC	Input processing by the WEB application
WEB_INIT.	Initialization of the WEB application

Attribute Name: Manual Planning

CELLS_TRANS	Number of transported data cells (including dummy cells and cells added in the user exit)
CELLS_PP	Number of data cells prepared by SEM/BW/BW-BPS

Attribute Names: Planning Functions

RECS_READ	Number of data records from the planning package
RECS_READ_REF	Number of reference data records
RECS_NEW	Number of data records generated
RECS_CHG	Number of data records changed
RECS_DEL	Number of data records deleted
TYPE	Contains additional information ... BW_DATA_READ: Lock type, such as "E" for exclusive lock NAME = "API" : Name of the function module NAME = "PF" : Type of planning function

Attribute Name: WEB Applications

PAGE	Name of the WEB application
T_PREV_CALL	Response time of the last dialog step of the WEB planning session.

Attributes for General Use

P_AREA	Planning area
P_LEVEL	Planning level
P_PACKAGE	Planning package
P_METHOD	Planning method
P_PARAM	Planning parameter
T_BRUTTO	Total runtime
T_NETTO	Total runtime minus runtime used for subordinate elements
T_STAMP	Timestamp specifying the start time

7.1 Useful Tables

- The assignment between the names of an area and the appropriate InfoCube/BW is stored in the table UPC_BW_AREA. Multi-planning areas are not contained in this table.
- Table UPC_STATISTIC3 displays the statistical results.
- Table UPC_STATISTIC2 contains the hierarchy information
- Table RSDDSTAT contains the BW statistical results.

7.2 Useful Transactions

- BPS_STAT01 Statistics for SEM/BW/BW-BPS
- BPS_TRACE Records all callups of the SEM/BW/BW-BPS API (function group upc_api)
- STAD Dialog statistics of the R/3 System (incl. application statistics)
- ST03 Including overviews of the dialog statistics
- SMICM Monitoring transaction for the Internet Connection Manager
- ST05 Various traces (RFC, SQL, ...)
- SE30 ABAP Trace
- SM04 Memory usage

7.3 Automatic Evaluation of Statistical Data

To perform an initial analysis of a problem, you need an overview of the system and the system behavior. Comparing the middle-range transfer rates for transaction data per InfoCube, for example, can provide important clues when verifying whether the whole system is slow or whether there is just a problem with a particular InfoCube. This obviously has an effect on the subsequent actions you take.

The individual statistical records (transaction BPS_STAT0) do not provide a direct overview of this information. This is why the evaluation transaction BPS_STAT1 has been created. It calculates average values for the different areas. The information it provides includes the following:

- Categorization of planning functions by runtime and number of processed data records
- Average/maximum user activities in hour-specific average
- Manual planning and planning functions per user
- Read transaction data per area

7.4 Memory Consumption and CPU Time

These variables can be determined using the transaction STAD. Since work with this detailed knowledge with R/3 requires analysis and tuning transactions, contact your SAP consultant for more detailed information.