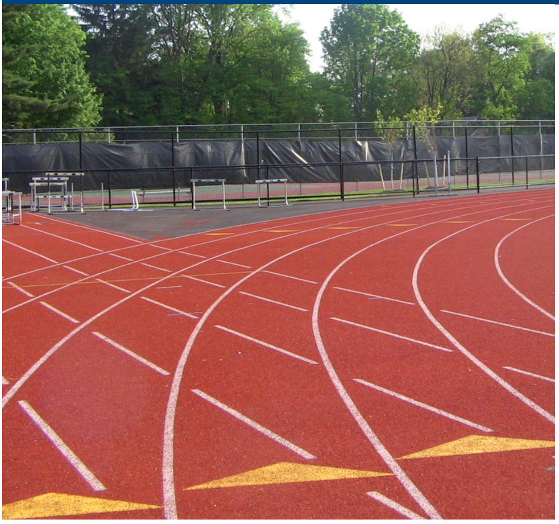


Benchmarking IT application controls

A practical guide for SAP

Ramon de Bruijn RE and Maurice op het Veld RE

Many organizations are implementing cost efficiency programmes to lower the cost of compliance. Besides the risk-based, top-down management assessment and audit approach, moving toward IT application controls in the (SOX) compliance framework also leads to additional cost efficiency. The challenge is, however, to implement a sustainable approach to test the operating effectiveness of these IT application controls on an annual basis. Benchmarking, or 'baselining' as it is also called, the IT application controls can enable that efficient approach to SOX. An ERP package such as SAP can facilitate this benchmarking strategy by clever use of the information and possibilities provided by the system, the specifics of which will be discussed in this article.

Introduction

The introduction of SOX placed enormous pressure on companies. During the first year, the demands to meet the basic requirements of the Act meant that many companies struggled simply to comply. In subsequent years, the issue of cost became more relevant. Now companies are struggling with how to react to IT opportunities and how to cut costs without endangering their compliance.

Because many companies faced serious dilemmas in striking a balance between complying with the regulations and keeping costs down, the SEC (US Securities and Exchange Commission) and PCAOB (Public Company Accounting Oversight Board, to oversee the auditors of public companies) issued new guidelines on what companies and auditors need to do in order to comply with SOX section 404. The new PCAOB standard, Accounting Standard no.5 (AS5), provides a more principle-based guideline on how auditors should conduct their audit of internal control (also known as Internal Control over Financial Reporting, or ICOFR). The new guideline does not change the fundamental requirements that SOX established for management and their auditors to report on the effectiveness of internal control systems. In short, the new guideline enables companies and their auditors to focus on high-risk areas, to concentrate only on areas that could harbour material misstatements, and to use the most practical routes to test the key controls and underlying systems.

R.J.C.H.M. de Bruijn RE
is a senior IT compliance specialist at ASML, the world's leading provider of lithography systems for the semiconductor industry. He is responsible for leadership in IT Compliance processes, deployment and improvement initiatives to foster adherence to IT processes and standards. In his 9-year career as IT Auditor, he has acquired significant experience in IT compliance both as a consultant and manager.

debruijn.ramon@kpmg.nl

M.A.P. op het Veld RE
is a senior manager at KPMG IT Advisory. His activities focus on audit and advisory services in Enterprise Resource Planning (ERP) systems, with SAP in particular. His expertise includes continuous monitoring/auditing, GRC processes/tooling, SOx, control frameworks and optimization projects. He is also a teacher of IT auditing at the TIAS business school and is a guest teacher of the graduate accountancy programme at the University of Tilburg.

ophetveld.maurice@kpmg.nl

The issuance of AS5 was followed by many discussions on how companies should adopt a risk-based control rationalization approach as part of a larger effort towards SOX optimization. These discussions centred on designing and deploying only the most effective and efficient controls to address financial reporting risks. Control rationalization applies a top-down, risk-based approach, eliminates unnecessary controls, uses risk-based testing plans, and optimizes the design of company-level and transaction controls. Other opportunities to further reduce the 'cost of compliance' may include using IT controls in the first place, embedding continuous testing or control framework integration ([Perk07]).

This article explores another opportunity AS5 provides to cut costs without endangering compliance. This is the possibility to use a benchmarking test strategy for automated controls, thereby significantly reducing an organization's and external auditor's efforts relating to testing controls on an ongoing basis. Although the efficient benchmarking strategy is applicable to automated controls in all ERP and IT systems, this article elaborates on practical implementation within a SAP environment.

The internal controls environment

Internal or external compliance is about behaviour in accordance with established guidelines, specifications or legislation, such as financial statements for example (focused on by SOX), but may also include intellectual property, privacy, etc. A risk assessment will identify the efficient and effective key internal controls in the main business processes. As the business processes are supported by IT or ERP systems, IT related business controls can be identified. Unfortunately, during the pressure of the first SOX years, it was primarily manual and detective controls that were identified. In the current SOX improvement projects, organizations are trying to find an optimum mix of manual and IT controls to leverage more of the relatively efficient and effective IT controls.

When talking about SOX management assessments or audits, three types of internal controls can normally be distinguished within the business processes (see also Figure 1):

- Manual controls: these are the manual checks performed by people and can include monthly inventory counting, but also an activity such as checking the invoice against the goods receipt packing note and the purchase form before issuing a payment. As mentioned before, this type of control was mainly identified in the early SOX years and takes much time to test only 'a sample' to provide an indication of the operating effectiveness.
- IT application controls: the IT application controls are the opposite of the manual controls. These controls are implemented in the IT or ERP systems and are used every time transactions go through the system. In other words, these controls are enabled and effective for the whole population, as these controls are normally settings in the IT or ERP systems. These controls can be tested in an efficient way, thereby reducing the cost of compliance. Examples of IT application controls are the three-way match, automatic invoicing after goods issue, purchase order approvals, interfaces, authorizations, and segregation of duties.
- IT-dependent manual controls: these controls consist of a manual activity, on one hand, and of an automated activity determined by the system, e.g. the output of a report, on the other. The manual part still requires sample-testing if the organization used the report and the manual follow-up, and the automated part requires testing to determine if the report is reliable. The check by the accounts payable clerk to analyze possible errors and deviations on the invoice payment proposal list is an example of an IT-dependent manual control.

IT General Controls are embedded within IT processes that provide a reliable operating environment and support the effective operation of automated controls (application and IT-dependent manual controls) ([ITGI06]). IT general controls relevant for SOX include:

- Program development
- Program changes
- Access to programs and data
- Computer operations

When taking a benchmarking strategy into account, these IT general controls should be on a required level.

The automated control challenge

The previous section outlined the opportunities of automated controls. Identification of automated controls and taking them into scope for management assessment or external audit constitute important steps in an efficient SOX approach. The next

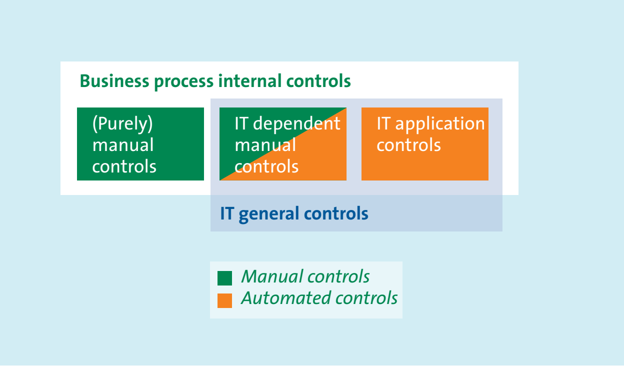


Figure 1. **Types of Internal controls in business processes.**

step is to test the design and operating effectiveness and to 'baseline' the controls. In general, the following ways or combinations can be distinguished ([Perk07]):

- The user-acceptance test: automated controls should be part of a user-acceptance test (UAT). If the UAT is well documented and carried out at the appropriate quality level, these results can be taken into account for testing operating effectiveness.
- Verification of settings / tables / parameters: in the current IT and especially ERP packages (like SAP), the mechanics of automated controls are 'customized' using the settings or parameters. Although it is often difficult to test the completeness of the operating effectiveness using the method, it can be very efficient.
- Trial & error (falsification): test the operating effectiveness of the automated control in a test environment. In other words, does the automated control do what you would expect upfront? Of course, the test and production environment should be equal.
- Audit software (CAAT): audit software can be used to provide evidence that all transactions of a specific business scenario have been processed according the expected script.
- Application code analysis: a very time-consuming way to test the operating effectiveness of the automated control can be to analyze the application code.

There is no standard way to test the design and operating effectiveness of automated controls. Furthermore, methods are sometimes combined, as with the verification of settings and audit software when the IT general controls have some deficiencies, for example, or when absolute assurance is required for very important automated controls.

The assumption is that the automated controls do not change much after the first full test year (this is often a time-consuming change process). Although the test of operating effectiveness can be carried out much more quickly in subsequent years, a substantive effort has to be made by management and the external auditor. The challenge is to determine how the already more efficient reliance on automated controls can be used in an efficient test approach.

> The challenge is how the already more efficient reliance on automated controls can be used in an efficient and sustainable test approach

Benchmarking – the theory

The Special Topics appendix of AS5 deals with the benchmarking of automated controls. It states, '*Entirely automated application controls are generally not subject to breakdowns due to human failure. This feature allows the auditor to use a "benchmarking" strategy.*'

And even though AS5 specifically addresses the possibility of using a benchmarking test strategy, it is not new, as section E122 of Auditing Standard No. 2 (AS2) specifically acknowledges benchmarking as a testing strategy permitted by the standard.

Traditionally, benchmarking referred to measuring a product or service according to specified standards in order to compare it with one's own product or service. For example, the NASDAQ may be used as a benchmark against which the performance of technology stock can be compared. Another example is comparing or benchmarking IT costs with other business units or organizations.

Benchmarking as mentioned in AS5 is different in the sense that it implies a testing strategy for audited automated controls in subsequent-year tests. Benchmarking as such involves documenting and testing controls embedded in an organization's applications and key reports, in order to determine whether they have maintained their integrity over time. In other words, if you are feeling well, you do not have to go the doctor every year or undergo a full-body scan. This approach is attractive since a full audit of controls in the first year without re-auditing them in subsequent years (unless a major change is made) represents a significant cost-saving opportunity.

AS5 set the ground rules for using a benchmarking test strategy. In short, if IT general controls related to change management are effective, and the automated control has been tested in the past, annual testing is not required. The benchmark only should be established periodically.

Preconditions

The following preconditions for using a benchmarking strategy must be taken into consideration:

1. The general controls over program changes, access to programs, and computer operations should be tested effectively when establishing the baseline (AS5 item B29).
2. The application should operate in a stable environment and there is only a limited number of changes (AS5 item B31).
3. The control should be matched to a specific program within the application (AS5 item B31).

4. There must be information regarding the programs in the production process to prove that controls within the program have not changed (AS5 item B31).
5. The benchmark must be re-established after a period of time (AS5 item B33).

Item 1 is self-evident, since the IT general controls provide a basic assurance regarding the performance of the automated controls.

For item 2, the frequency of changes is a strong indicator whether or not an automated control might be well-suited for benchmarking. Again this will be obvious, as benchmarking only works for automated controls that have not changed since setting the baseline.

Items 3 and 4 are less obvious, since much software no longer works with compilation dates for example. Complex ERP systems now cover large-scale business processes within the dynamic environments of today's companies, so items 3 and 4 are certainly a challenge. This will be addressed specifically with respect to SAP later in the article.

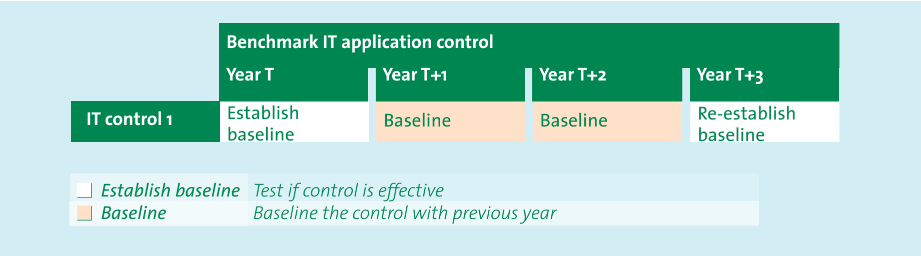
Finally, item 5 is again self-evident, although it provides additional possibilities when companies integrate baseline activities into the extended scope of user-acceptance testing.

Defining the benchmarking test strategy

When defining a benchmarking test strategy, the following actions could be taken into account:

Step 1. Define the scope of automated controls for benchmarking

First and foremost, it is important to define the scope of automated controls that are well-suited for benchmarking. The controls should be either IT application controls or IT-dependent manual controls, where the IT part is the part to be benchmarked (the manual part should be tested for operational effectiveness each year). Generally, these controls are configurable parameters, custom-built routines, or queries that ensure the complete, accurate, timely and proper processing and reporting of (financial) transactions. Because automated controls often depend on more than one factor to work effectively, it is important to consider not only the front-end functionality but also the associated dependencies. For instance, an automated approval control in SAP R/3 may be configured to automatically approve three-way matched invoices below a specific Euro threshold. However, this control depends on the accuracy of the established threshold value (SAP setup or customization) and the assurance that only authorized individuals can access the configuration information. Similarly, as reports generated by an application depend on the integrity of the source data and the reporting system's logic, it is critical to consider both when assessing the integrity of a report used to perform an IT-dependent manual control. This combined group of front-end automated controls and dependencies may be the starting point of your baseline scope definition.



	Benchmark IT application control			
	Year T	Year T+1	Year T+2	Year T+3
IT control 1	Establish baseline	Baseline	Baseline	Re-establish baseline

☐ *Establish baseline* *Test if control is effective*
▨ *Baseline* *Baseline the control with previous year*

Figure 2. **Example of benchmark strategy for automated controls.**

When defining the scope, also consider efficiency. Given the work effort involved in establishing and maintaining the baselines, it is important for organizations to assess whether or not yearly testing might be a more efficient approach than benchmarking. For example, if testing a particular IT control can be established by a simple screen-print of configuration settings from the target system, this could cost less effort than providing evidence that the configuration control has not changed. To achieve the advantages of a baseline approach, organizations must limit their baseline scope to selected automated controls that require significant effort in testing. To ensure the benchmarking test strategy is beneficial, the following factors must be considered in advance:

- The cost of the annual operating effectiveness testing of automated controls.
- Whether and when changes to the applications, infrastructure, related infrastructure and controls are planned (an example is an upgrade of SAP R/3).
- Whether there are any current deficiencies in IT general controls or application/automated controls.

Step 2. Validate that an initial baseline of scoped controls has been established

When defining the scope of the benchmarking tests, it is important to ensure that the existing key automated controls have already been identified and documented as part of the overall SOX approach. The baseline approach is less feasible for organizations that have known deficiencies in their IT general controls, especially in security and application change management. Before a benchmarking test strategy can redeem its promise, organizations must have demonstrated the effectiveness of their IT general controls.

Step 3. Define rotation schedule

Once the baseline has been established, the benchmarking test strategy provides evidence for a number of years. As mentioned, the benchmark must be re-established after a period of time (AS5 item B33). Although AS5 sets no rules, three years constitutes a good practice for an appropriate frequency for re-establishing baselines. It is also important to re-establish baselines where significant changes occur within the applications. When significant changes occur, integrating baseline activities into an extended scope of user-acceptance testing should only be considered when the application operates in a stable environment and a limited number of changes are expected after these significant ones.

Rather than re-establishing all baselines every few years, it may make sense to adopt a rotation schedule whereby a portion of automated controls are tested each year, thereby spreading the effort over several years. By carefully planning and extending the scope of user acceptance testing, the benchmarking test strategy provides a sustainable and efficient approach. As stated previously, the UAT must be well documented and carried out by (IT) staff with knowledge of process controls as well as of the functionality of the system.

Integrating baselining into the UAT process thus helps further reduce overall compliance costs, and facilitates early deficiency identification and rectification. In this way, you can begin to reap the long-term rewards of a benchmarking test strategy while the applications continue to maintain their integrity over time.

The timing within a year of the rotation schedule is something to consider as well. Since there are three options, it is important to carefully plan them within the year. Testing to establish a baseline is best done early in the year (first quarter), so that if inaccuracies are detected there is time to rectify the control. The Baseline test itself, however, is best planned late in the year (end of third quarter) because little change can be expected in the rest of the year. And the User-Acceptance Test is mostly dependent on the delivery date of the automated system being developed and tested.

Benchmarking in a SAP environment

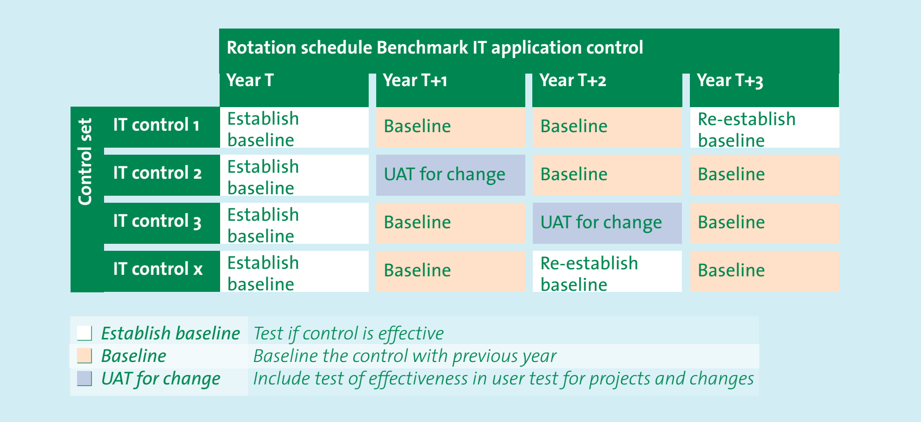
As described, benchmarking can be used to conclude that an automated control is effective without having to repeat the specific tests of the operation of the automated control. The nature and extent of the evidence that the auditor should obtain to verify that the control has not changed may vary depending on the circumstances, including the strength of the company's program change controls.

A known challenge will be the AS5 requirement regarding the 'compilation date', which is not explicitly available in SAP. On this issue, the article will provide a practical solution for SAP with respect to the two types of automated controls that have been distinguished.

Although the following sections are more related to SAP, they nevertheless give guidelines for benchmarking automated controls in other ERP packages.

IT application controls

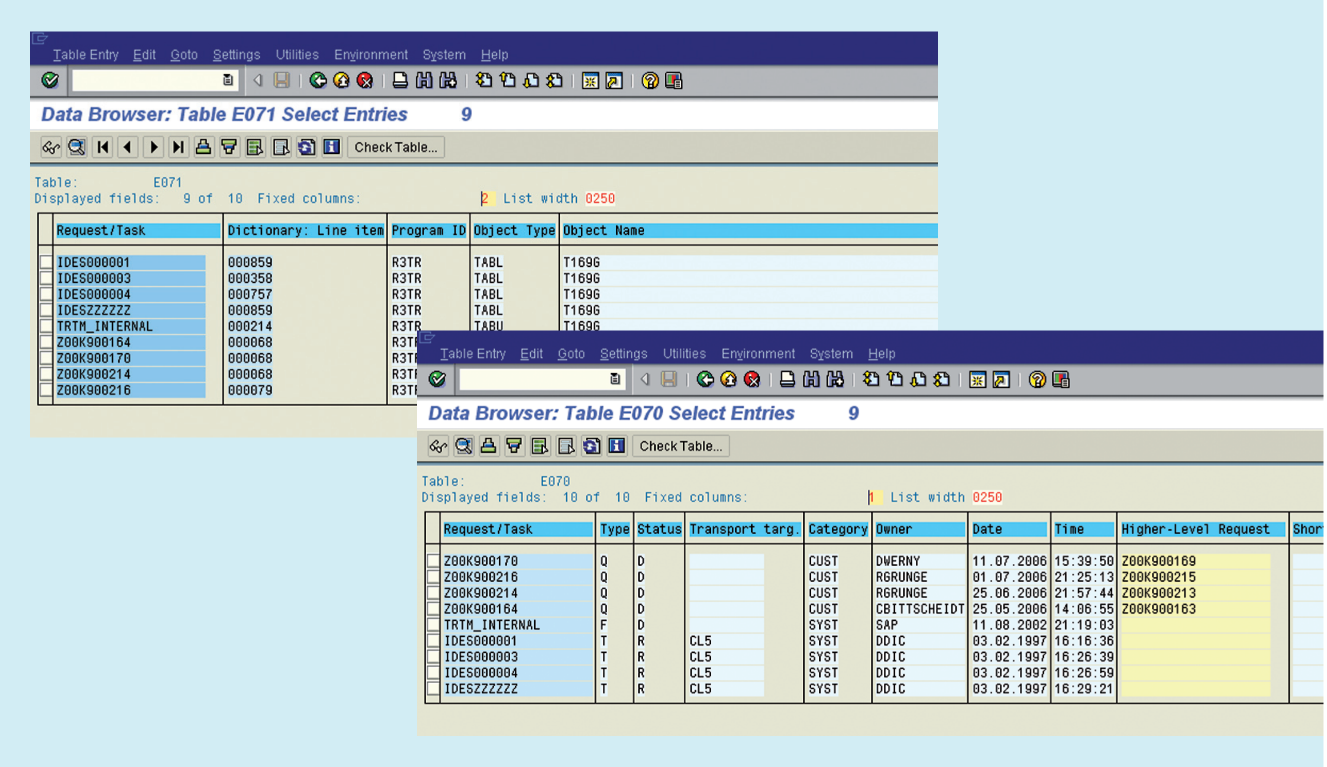
In SAP, many IT application controls have already been set up in the customization process (SAP transaction code *SPRO*). These settings are usually stored in tables. For example, the



		Rotation schedule Benchmark IT application control			
		Year T	Year T+1	Year T+2	Year T+3
Control set	IT control 1	Establish baseline	Baseline	Baseline	Re-establish baseline
	IT control 2	Establish baseline	UAT for change	Baseline	Baseline
	IT control 3	Establish baseline	Baseline	UAT for change	Baseline
	IT control x	Establish baseline	Baseline	Re-establish baseline	Baseline

□	*Establish baseline*	*Test if control is effective*
🟧	*Baseline*	*Baseline the control with previous year*
🟦	*UAT for change*	*Include test of effectiveness in user test for projects and changes*

Figure 3. **Example of a benchmark rotation schedule.**



Data Browser: Table E071 Select Entries 9

```
Table:            E071
Displayed fields:   9 of  10  Fixed columns:          2  List width 0250
```

	Request/Task	Dictionary: Line item	Program ID	Object Type	Object Name
□	IDES000001	000859	R3TR	TABL	T169G
□	IDES000003	000358	R3TR	TABL	T169G
□	IDES000004	000757	R3TR	TABL	T169G
□	IDESZZZZZZ	000859	R3TR	TABL	T169G
□	TRTM_INTERNAL	000214	R3TR	TABU	T169G
□	Z00K900164	000068	R3T		
□	Z00K900170	000068	R3T		
□	Z00K900214	000068	R3T		
□	Z00K900216	000079	R3T		

Data Browser: Table E070 Select Entries 9

```
Table:            E070
Displayed fields:  10 of  10  Fixed columns:          1  List width 0250
```

	Request/Task	Type	Status	Transport targ.	Category	Owner	Date	Time	Higher-Level Request	Shor
□	Z00K900170	Q	D		CUST	DWERNY	11.07.2006	15:39:50	Z00K900169	
□	Z00K900216	Q	D		CUST	RGRUNGE	01.07.2006	21:25:13	Z00K900215	
□	Z00K900214	Q	D		CUST	RGRUNGE	25.06.2006	21:57:44	Z00K900213	
□	Z00K900164	Q	D		CUST	CBITTSCHEIDT	25.05.2006	14:06:55	Z00K900163	
□	TRTM_INTERNAL	F	D		SYST	SAP	11.08.2002	21:19:03		
□	IDES000001	T	R	CL5	SYST	DDIC	03.02.1997	16:16:36		
□	IDES000003	T	R	CL5	SYST	DDIC	03.02.1997	16:26:39		
□	IDES000004	T	R	CL5	SYST	DDIC	03.02.1997	16:26:59		
□	IDESZZZZZZ	T	R	CL5	SYST	DDIC	03.02.1997	16:29:21		

Figure 4. **Benchmarking IT application controls using transport logs.**

tolerance limits for the three-way matches are stored in the SAP table called '*T169G*'. These settings are the current settings, in other words, there is no compilation date or audit trail on the previous settings (including dates). However, for the 'compilation date' required by AS5, there are two practical solutions in SAP:

Enabling logging on the specific customizing table

Standard SAP does not log all changes to the SAP tables. To allow SAP logging of specific SAP tables, the following steps should be performed.

First, the '*REC/CLIENT*' client setting or parameter should be set. Be careful when switching the *REC/CLIENT* in the client settings to Yes, because standard SAP will then log many tables including transactional tables. This will cause SAP to produce excessive logging, possibly slowing down the system and using considerable disk space.

The second step to select the SAP tables for logging is very important. The *DD09L* transaction code will enable you to include or exclude SAP tables in the logging. Besides selecting the SAP tables with the settings for the IT application controls to benchmark in scope (in the example *T169G*), logging other critical systems tables (such as *T000*, *T001*, *TCURR*, etc.) is also recommended. The transactional SAP tables should be excluded from logging, in order to prevent excessive logging.

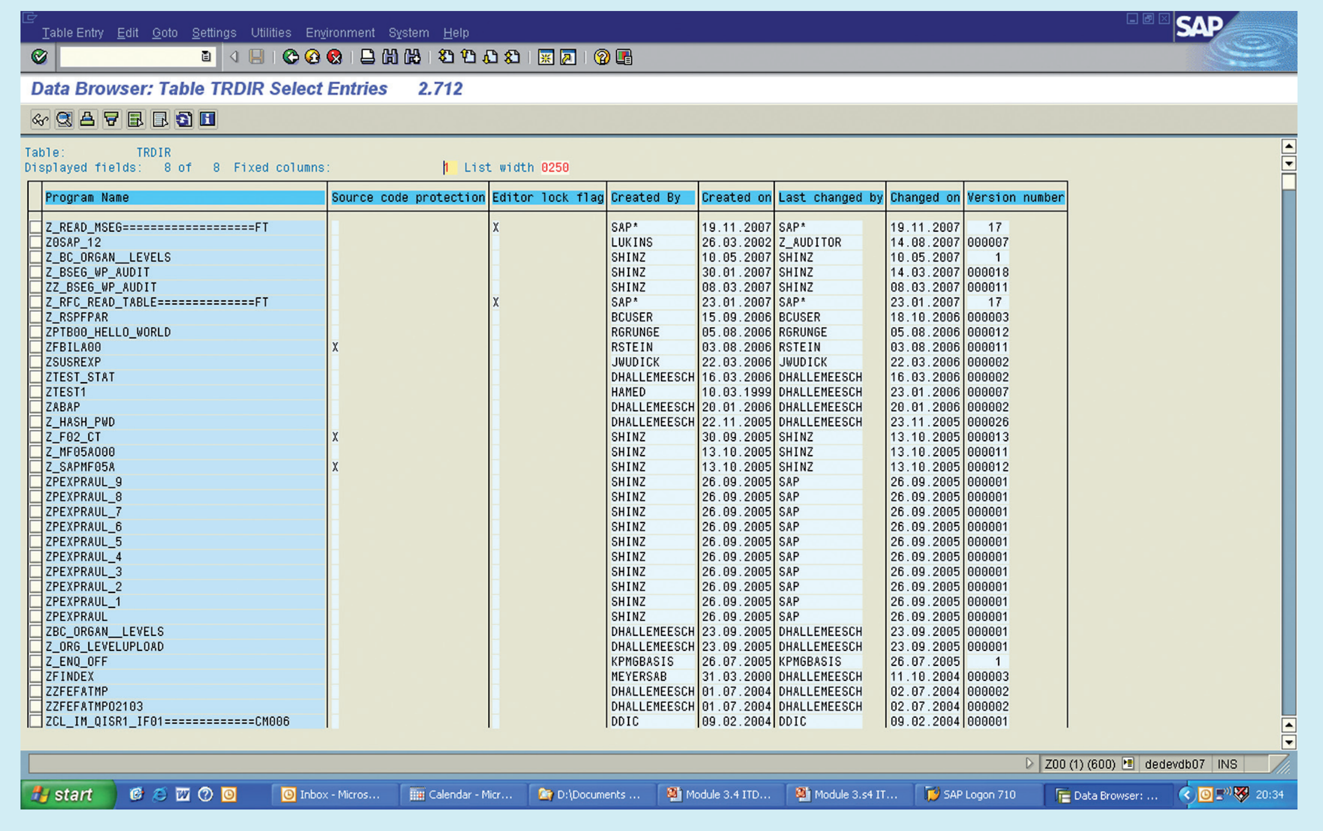
Finally, the logging results for a certain period can be viewed using the *RSTBHIST* program. The program will give you a list of the current and previous setting, including the date of change. If the setting of an IT application control has not been changed since the last baseline, this control is suitable for benchmarking.

Using the SAP transport log

All transports to the production client are logged in SAP system tables, including the last transport (i.e. change) dates. The details of the changes to object classes in the transports are logged in the *E071* SAP table. Object classes are also SAP tables, and enable to identify the last change dates to the settings of the IT application controls stored in those tables.

In the example (see Figure 2) you will find that table *T169G* has been transported a number of times to the production system. The Request/Task field can be linked to the E070 table to find the corresponding transport date.

The last transport of the *T169G* table in the example shown in Figure 4 was 11 July 2006. If the baseline testing for this IT application controls was established after this date in 2006, the



Data Browser: Table TRDIR Select Entries 2.712

```
Table:          TRDIR
Displayed fields:   8 of   8  Fixed columns:          1  List width 0250
```

	Program Name	Source code protection	Editor lock flag	Created By	Created on	Last changed by	Changed on	Version number
□	Z_READ_MSEG==================FT		X	SAP*	19.11.2007	SAP*	19.11.2007	17
□	Z0SAP_12			LUKINS	26.03.2002	Z_AUDITOR	14.08.2007	000007
□	Z_BC_ORGAN__LEVELS			SHINZ	10.05.2007	SHINZ	10.05.2007	1
□	Z_BSEG_WP_AUDIT			SHINZ	30.01.2007	SHINZ	14.03.2007	000018
□	ZZ_BSEG_WP_AUDIT			SHINZ	08.03.2007	SHINZ	08.03.2007	000011
□	Z_RFC_READ_TABLE==============FT		X	SAP*	23.01.2007	SAP*	23.01.2007	17
□	Z_RSPFPAR			BCUSER	15.09.2006	BCUSER	18.10.2006	000003
□	ZPTB00_HELLO_WORLD			RGRUNGE	05.08.2006	RGRUNGE	05.08.2006	000012
□	ZFBILA00	X		RSTEIN	03.08.2006	RSTEIN	03.08.2006	000011
□	ZSUSREXP			JWUDICK	22.03.2006	JWUDICK	22.03.2006	000002
□	ZTEST_STAT			DHALLEMEESCH	16.03.2006	DHALLEMEESCH	16.03.2006	000002
□	ZTEST1			HAMED	10.03.1999	DHALLEMEESCH	23.01.2006	000007
□	ZABAP			DHALLEMEESCH	20.01.2006	DHALLEMEESCH	20.01.2006	000002
□	Z_HASH_PWD			DHALLEMEESCH	22.11.2005	DHALLEMEESCH	23.11.2005	000026
□	Z_F02_CT	X		SHINZ	30.09.2005	SHINZ	13.10.2005	000013
□	Z_MF05A000			SHINZ	13.10.2005	SHINZ	13.10.2005	000011
□	Z_SAPMF05A	X		SHINZ	13.10.2005	SHINZ	13.10.2005	000012
□	ZPEXPRAUL_9			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_8			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_7			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_6			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_5			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_4			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_3			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_2			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL_1			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZPEXPRAUL			SHINZ	26.09.2005	SAP	26.09.2005	000001
□	ZBC_ORGAN__LEVELS			DHALLEMEESCH	23.09.2005	DHALLEMEESCH	23.09.2005	000001
□	Z_ORG_LEVELUPLOAD			DHALLEMEESCH	23.09.2005	DHALLEMEESCH	23.09.2005	000001
□	Z_ENQ_OFF			KPMGBASIS	26.07.2005	KPMGBASIS	26.07.2005	1
□	ZFINDEX			MEYERSAB	31.03.2000	DHALLEMEESCH	11.10.2004	000003
□	ZZFEFATMP			DHALLEMEESCH	01.07.2004	DHALLEMEESCH	02.07.2004	000002
□	ZZFEFATMP02103			DHALLEMEESCH	01.07.2004	DHALLEMEESCH	02.07.2004	000002
□	ZCL_IM_QISR1_IF01============CM006			DDIC	09.02.2004	DDIC	09.02.2004	000001

Figure 5. **Benchmarking IT-dependent manual controls using change logs.**

IT application control has not been changed since then. In other words, the setting is the same and the IT application control is suitable to be benchmarked for SOX purposes.

IT-dependent manual controls

IT-dependent manual controls in SAP are mainly the standard or custom-built SAP reports. The manual testing part of this IT-dependent manual control is dependent on the integrity of the SAP report. The purpose of benchmarking these reports is to identify whether or not the report has been changed since the last established baseline.

An easy way to identify the change dates of the SAP report is to use the *TRDIR* SAP table. The changes dates for all SAP reports and programs (including the custom-made, normally identified by a Z or a Y) can be found in this table (see Figure 5). This table shows the program name (i.e. SAP Report) and the last change date of that report. For example Z_RSPFPAR was created on 16 September 2006 and changed on 18 October 2006. This means that the report has not been changed in 2007 or 2008. If the baseline was established in 2007, this IT-dependent manual control is suitable for benchmarking.

Another example shows that although the ZoSAP_12 report was created in 2002, the report was changed in 2007. This report is therefore not suitable for benchmarking. A new baseline should be established first, before applying benchmarking in the future.

As mentioned previously, a well-executed and documented user-acceptance test could provide the required evidence for establishing baselines, which will really embed the compliance in the IT processes and further decrease the effort required.

Benchmarking automated controls can be seen as a step towards continuous monitoring using the upcoming GRC tooling

The above-mentioned methods for benchmarking the automated controls in a SAP environment are only possible in a well-controlled IT environment with effective IT general controls. However, in an environment with IT general control deficiencies, the methods explained above can still give the auditor insight into the changes to key IT application and IT-dependent manual controls. This information can be used in a risk assessment to choose a suitable IT management assessment or IT audit.

In conclusion

In the article entitled 'Testing application controls' by Van der Perk ([Perk07]) it was stated that the identification of IT controls in the business control frameworks is an important aspect of sustainable compliance. In a stable IT environment (limited IT general control deficiencies) using the benchmark strategy for automated controls, this can even lead to further reductions in the cost of SOX compliance.

This article gave a practical guidance for management and auditors on how to set up a benchmarking strategy for testing automated controls in their compliance testing. Organizations that want to embed the transparent benchmarking rotation schedule using UAT for re-establishing the baseline will need to have an already mature IT compliance organization.

One of the main challenges for benchmarking within a SAP environment is to identify the compilation dates. These compilation dates are not available in SAP, but the benchmark can still be conducted using the different change logs in SAP, as mentioned in this article. Another challenge is to identify the SAP tables that contain the IT application control settings.

In our opinion, the transparency needed to implement the benchmarking strategy can be seen as a step toward continuous monitoring of the effectiveness of automated controls using the upcoming GRC tooling.

References

[Brou06] P.P.M.G.G. Brouwers RE RA, M.A.P. op het Veld RE, and A. Lissone, *Tool based monitoring en auditing van ERP-systemen, van hebbeding naar noodzaak*, Compact 2006/2.

[ITGI06] IT Governance Institute (ITGI), *IT control objectives for Sarbanes Oxley*, 2006, (www.itgi.org).

[PCAOB AS2] www.pcaobus.org/Standards/Standards_and_Related_Rules/Auditing_Standard_No.2.aspx.

[PCOAB AS5] www.pcaobus.org/Standards/Standards_and_Related_Rules/Auditing_Standard_No.5.aspx.

[Perk07] L.J. van der Perk RA en P.N.M. Kromhout, *Testen van applicatiecontroles*, Compact 2007/3.