Script-controlled Backups in SAP HANA Database

Documentation of a template shell script for controlling backup execution

Richard Bremer, Product Management SAP HANA

Data Backups in SAP HANA Database can be invoked in many different ways. One such way is to invoke a data backup from the command line of the operating system of the SAP HANA Database server. For such purposes, the creation of a Linux shell script that controls backup execution is advisable. This document describes an example script for the Linux bash shell that can trigger the execution of data and configuration backups of an instance of SAP HANA Database. The script additionally contains functionalities to assist with (log-) backup housekeeping.

Contents

About this Shell Script

The backup script provided with SAP Note 1651055 serves as an example of how one can run a complete backup of SAP HANA Database as recommended in the SAP HANA Backup and Recovery Guide which is available in the SAP Library (http://help.sap.com/hana_appliance/) and on SAP Service Market Place (http://service.sap.com/hana/).

This document describes the backup script of version 04 and SAP Note 1651055 version 1. The script version is defined in line 2 of the backup script.

Important note regarding support for the backup script

The backup script is provided as part of a consulting note. It does not represent SAP-delivered software, and is hence also not supported by SAP.

Version 04 of this script is the final version. There is no intent to make any future modifications.

This script does not replace a professional backup solution. Where possible, backup solutions supporting HANA, e.g. via its BACKINT interface, should be used.

Features

Currently, the backup script can trigger a database backup using SAP HANA Database's backup functionality; and it can create backup copies of the SAP HANA Configuration Files.

Backups can be written into files with a configurable file name to which the current weekday is appended, i.e. names such as
`<backup_path>/<base_name>_Tue_<file_name_extension>;`
where `<backup_path>` and `<base_name>` can be configured in the parameterization of the shell script and `<file_name_extension>` is automatically created by SAP HANA Database. In this mode, each backup file will be stored for one week and will then be overwritten. This mode is well suited for scheduling backups e.g. via the Linux cron daemon and is the default behavior.

As an alternative to creating one uniquely named backup file per weekday, the script can create series of <n> uniquely named backup files (<n> being a configurable number), leading to a backup retention time of <n> days.

Backups can optionally written to fully configurable file names, i.e. to files of the form
`<backup_path>/<base_name>_<suffix>_<file_name_extension>;`
where all parts of the file name are as above, and `<suffix>` can be provided via a command line parameter.

The backup script measures the time needed to create the database backup. After completing a backup, the script will determine the size of the created backup files. Together with the time stamp of the point in time of starting the backup, this information is appended to a csv file which can be used for monitoring size and run time of the database backups. As an optional feature, the script can write this information to database tables as well.

Special run modes of the script (command line options -ld and -ll) help determining such log backup files (available since SPS 3 / revision 20 of SAP HANA database) that can be safely moved or deleted (based on the oldest data backup that may still be recovered). The script will not delete the log backup files, it only generates a list of candidates for deletion. It is the task of the database administrator to verify that the list is correct and to perform the movement or deletion of the log backup files manually.

Changes in Version 04

- Adjusted the script to a change in the hdbsql syntax that was introduced in hdbsql with SPS 8 / revision 80 and would cause the listing of log backups to fail
- Moved script configuration to dedicated configuration file. This makes it possible to prepare script configurations for multiple SAP HANA systems and to run them via a single instance of the backup.sh script. The name of the configuration file can be hardcoded in the backup script or passed via a command-line parameter.
- Introduced new command line switch "-cd" to delete log backups from catalog (and disk if so configured) using the `BACKUP CATALOG DELETE` command. This switch will usually be combined with switches -od -cd to automatically determine the backup ID of the oldest data backup that still resides on disk in its original location.
- Inifile customizations can be logged by the script. If logging to DB tables is enabled, we log all DB parameters that are not on their DEFAULT setting into a dedicated logging table – together with the BACKUP ID of the data backup. And we also write SQL statements to set all customized parameters into the script log file. These customizations are recorded for all inifiles of the entire DB system, in particular for all hosts of a scale-out landscape.
- Cleaning up all temporary files that are created in /tmp during script execution.

Important information

The backup script is provided without any support. To help you remember this, we have introduced several switches within the source code of the script that will remind you of this fact. The script will not work unless you have changed the corresponding sections of the script, thus acknowledging that you understand the reasons for the individual switches.

When running the script for the first time – with a valid script configuration – the script will stop at several locations, printing instructions on the command line on how you can acknowledge the given switch.

There will be no support for the script or its functionality offered via SAP support tickets. If you run into issues using the script, please raise a question in the relevant forums on SAP Community Network (SCN).

Limitations

While the script can make copies of the full configuration files, it only copies the files of one server (the host specified in the script configuration). There is no intention to change this behavior.

The retention time method will not work perfectly across the year-end. The reason is that it is simply based on a modulus operation on the "day-of-year".

Parameterization

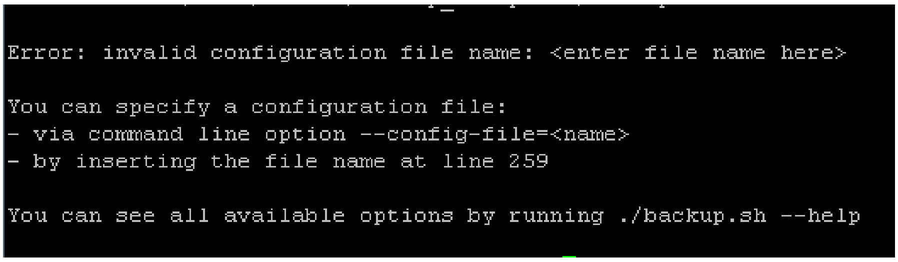
The details of the script execution are controlled by parameters in a configuration file; and by command line parameters. This Paragraph deals with the parameterization in the configuration file. Command line options are discussed below in the section on running the script.

Specifying the configuration file

The name of the configuration file to be used can be specified

- Either via command-line parameter --config-file=<file_name>
- Or it can be hardcoded in the backup script.

If the configuration file is not specified at all, the script will immediately terminate with an error message as printed below



```
Error: invalid configuration file name: <enter file name here>

You can specify a configuration file:
- via command line option --config-file=<name>
- by inserting the file name at line 259

You can see all available options by running ./backup.sh --help
```

If a configuration file name is hardcoded in the script and given on the command line, the command line input takes precedence.

Generic information on script parameterization

Before the backup script can be used, the parameterization has to be adjusted. Parameters must be given in the form:

```
<name>=<value>
```

Please note that there must not be any white space on either side of the ‘=’ operator. The name of parameters must not be changed. Capitalization is relevant. Many parameters refer to other parameters, for example in path names or file names. In this case, the parameter referred to is denoted in the form ${<name>}. As an example, the SAP System ID of the SAP HANA Database system is stored in a parameter named SID. The path at which the software executables of SAP HANA Database System with SAP System ID TNS are installed is provided via a parameter SIDPATH:

```
SID=TNS
SIDPATH=/usr/sap/${SID}
```

All available parameters are described in the following sections. The first column of each parameter table denotes whether or not the parameter must be changed:

- ☑ means that the parameter **must** be adjusted to your particular installation
- ⦸ denotes a parameter that **may be** adjusted to you needs if so desired
- ☒ means that typically this parameter refers to a default setting of SAP HANA Database that is **very unlikely to be changed** in any database installation.

Do check all parameters to make sure your backup parameterization is correct.

Database System Specific Information

In this section, parameters describing the SAP HANA Database system are defined.

Change	Name	Default	Description
☑	SID	---	SAP System ID of the SAP HANA Database System

Change	Name	Default	Description
☑	INSTANCE	---	Instance number of the SAP HANA Database System
☑	HOSTNAME	---	Hostname (the local host name of the database server. Do not use “localhost”, do not use the IP address and do not use the fully qualified (<hostname>.<domain>) name. Example: for database server “hana42.my_company.com”, the hostname is hana42)
☒	SIDPATH	/usr/sap/${SID}	The directory into which the binaries of the SAP HANA database system have been installed
☒	INSTPATH	${SIDPATH}/HDB ${INSTANCE}	The directory containing the instance data of the SAP HANA Database

Target directory and output file name for backup

In this section, the output locations for the database backup and for the backup copy of the configuration files are defined.

Change	Name	Default	Description
☑	BACKUP_BASE_DIRECTORY	${INSTPATH}/backup	Base path into which the backup script will write all backups. Sub-directories will be created below this path for data and configuration file backups. This directory - must be mounted in the file system path of your SAP HANA Database server; - should be located on a dedicated partition or device; - should not be on the same partition as the binary installation of SAP HANA Database, the log volume or the data volume of SAP HANA Database. It should also not be inside of the root partition of your Linux installation.
⦸	BACKUP_DATA_DIRECTORY	${BACKUP_BASE_DIRECTORY}/data	Directory into which the data files of the backup will be written
⦸	BACKUP_CONFIG_DIRECTORY	${BACKUP_BASE_DIRECTORY}/config_files/${WEEKDAY}	Directory into which the backup copy of the configuration files will be written. If a specific suffix has been provided on the command line (parameter --suffix), this suffix will be filled into variable ${WEEKDAY}
⦸	BACKUP_FILE_NAME	backup_${SID}_${WEEKDAY}	The file name of the backup files (without path name). If a specific suffix has been provided on the command line (parameter --suffix), this suffix will be filled into variable ${WEEKDAY}
☒	BACKUP_FILE_FULL_NAME	${BACKUP_DATA_DIRECTORY}/${B	The fully qualified file name of the backup files, i.e. file name including full path

		ACKUP_FILE_NAME}	

hdbsql and hdbuserstore information

hdbsql is a command line program which comes with any installation of SAP HANA Database and allows running SQL statements in the database from the command line. Hdbsql knows two ways of authenticating against SAP HANA Database: by user key from the hdbuserstore (preferred and recommended); or via classical credentials, i.e. user name and password.

hdbuserstore is a secure store for logon data of an SAP HANA Database that comes with an installation of the hdbclient package. Entries in the SAP HANA user store are identified by a user store key and contain all information required for logging on to the database server. For more details regarding hdbuserstore see section “Using hdbuserstore”

The following parameters are available in the backup script to configure hdbsql and its authentication against SAP HANA Database:

Change	Name	Default	Description
☒	HDBSQL_EXE	${INSTPATH}/exe/hdbsql	File name of the executable for hdbsql
⦸	USE_HDBUSERSTORE	TRUE	Set to TRUE if you intend to use hdbuserstore. Capitalization matters. Any other value will be interpreted as “FALSE”. If you do not use hdbuserstore, you must specify connection information in file ${BACKUP_SQL}. For this purpose, modify the section of the backup script in which file ${BACKUP_SQL} is generated.
☑	USERSTORE_KEY	---	The key in hdbuserstore that shall be used by hdbsql to authenticate against SAP HANA Database

Log and Statistics Output into log files

The backup script will write execution log information to a dedicated log file. In addition, any output of the actual backup execution via hdbsql will be written into another log file. And finally, the backup script collects statistical information about backup size and time needed to create the backup in a csv file.

Change	Name	Default	Description
☑	LOG_DIRECTORY	${INSTPATH}/${HOSTNAME}/trace	Directory in which we create log files of the script
⦸	SCRIPT_LOG	${LOG_DIRECTORY}/script_backup_${WEEKDAY}.txt	Log file of script execution
⦸	BACKUP_LOG	${LOG_DIRECTORY}/backup_log_${WEEKDAY}.txt	Log file of actual data backup execution via hdbsql (output of hdbsql)
⦸	TIMING_MEASUREMENTS	${LOG_DIRECTORY}/timing_measurements.csv	Statistics collection: csv file with columns for time stamp of start of backup, backup size, and time needed to generate backup

Log and Statistics Gathering in Database Tables

The backup script can optionally also write log and statistics information into dedicated tables on SAP HANA Database. The table layout is defined in the script, target schema, tables names etc can be configured.

Change	Name	Default	Description
☑	WRITE_STATS_TO_TABLE	TRUE	Switch: set to TRUE if you want to write logs and statistics to tables. Recommended value: TRUE
☑	STATS_SCHEMA	---	Enter database schema to create the log and statistics table in. The user defined via the userstore key below must have CREATE ANY and INSERT privileges for that schema.
⦸	STATS_TABLE	Z_SCRIPT_BACKUP_STATISTICS	Name of table in which we gather size and runtime statistics of the data backup execution. If the table does not exist in schema STATS_SCHEMA, it will be created by the script.
⦸	LOG_TABLE	Z_SCRIPT_BACKUP_RUNS	Name of table in which we gather log information of the script execution. If the table does not exist in table ${STATS_SCHEMA}, it will be created by the script.
⦸	CONFIG_TABLE	Z_SCRIPT_CONFIG_CUSTOMIZING	Name of table in which we record all customized entries of the .ini files – that is, all database configuration changes with respect to HANA's default values.
⦸	USERSTORE_KEY_STATS	${USERSTORE_KEY}	Key in hdbuserstore for database user that shall write the statistics entries. The default is to use the same user that is running the data backup (so we re-use ${USERSTORE_KEY} as value here). You may use a different entry if you really want to.

Program Control

The backup script can be configured to wait for a given amount of time before actually starting the backup, giving the user a chance to cancel backup when running the script interactively.
While waiting, information about the database system and the statistics record of the last backup is displayed. Elapsed waiting time is shown using a simple progress bar.

The splash screen can be suppressed by supplying command line parameter –q

A useful configuration is to generally enable the wait time, and using the command line switch –q to suppress the wait time when backups are scheduled non-interactively (e.g. via cron daemon).

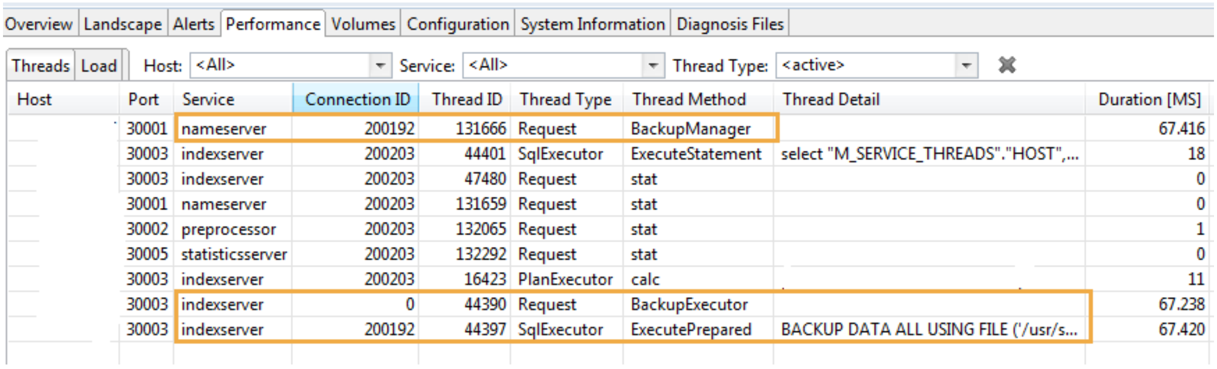
Change	Name	Default	Description
⦸	WAIT_AND_WARN	TRUE	If set to TRUE (capitalization matters), the backup script will wait for a given amount of time before actually starting the backup. Any other value is interpreted as FALSE
⦸	WAIT_TIME	20	Time in seconds before actual backup starts.

Change	Name	Default	Description
☑	CATALOG_DELETE_FROM_DISK	TRUE	When deleting log backups using command-line option -cd (that is, when the script will invoke the BACKUP CATALOG DELETE command), you can choose to only delete log backup entries from the backup catalog (value = FALSE) or also to delete the log backup files from disk (TRUE). Technically, when this parameter is true, we invoke the BACKUP CATALOG DELETE command including the “WITH FILE” clause.

Monitoring and Statistics Gathering

Monitoring backup execution

While data backups are running in SAP HANA Database, one can find corresponding threads in the Administration Console of SAP HANA Studio, tab performance. There is one Thread with method “BackupManager” of the nameserver, and threads with methods “BackupExecutor” and “ExecutePrepared” of the indexserver.

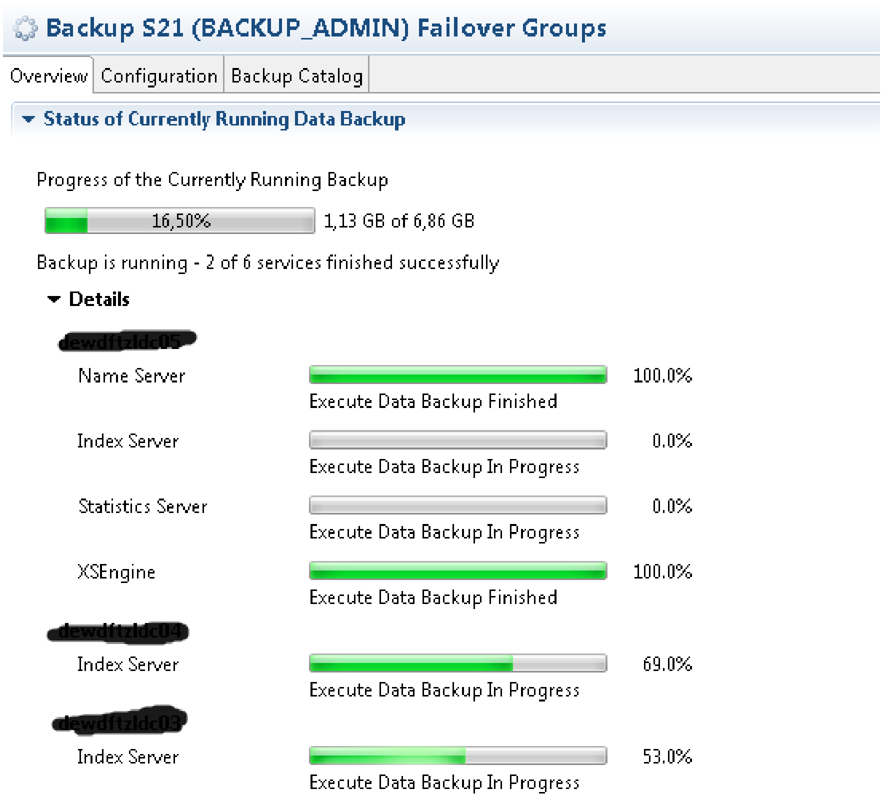


Host: <All> Service: <All> Thread Type: <active>

Host	Port	Service	Connection ID	Thread ID	Thread Type	Thread Method	Thread Detail	Duration [MS]
	30001	nameserver	200192	131666	Request	BackupManager		67.416
	30003	indexserver	200203	44401	SqlExecutor	ExecuteStatement	select "M_SERVICE_THREADS"."HOST",...	18
	30003	indexserver	200203	47480	Request	stat		0
	30001	nameserver	200203	131659	Request	stat		0
	30002	preprocessor	200203	132065	Request	stat		1
	30005	statisticsserver	200203	132292	Request	stat		0
	30003	indexserver	200203	16423	PlanExecutor	calc		11
	30003	indexserver	0	44390	Request	BackupExecutor		67.238
	30003	indexserver	200192	44397	SqlExecutor	ExecutePrepared	BACKUP DATA ALL USING FILE ('/usr/s...	67.420

Once backup is finished, entries are added to the database’s monitoring tables PUBLIC.M_BACKUP_CATALOG and PUBLIC.M_BACKUP_CATALOG_FILES.

More comfortably, you can see the backup progress also in the backup editor of the SAP HANA studio. Even if you trigger the backup using the SQL interface (as this script does), the backup editor can show backup progress in a beautiful way:



Backup S21 (BACKUP_ADMIN) Failover Groups

Status of Currently Running Data Backup

Progress of the Currently Running Backup

16,50% 1,13 GB of 6,86 GB

Backup is running - 2 of 6 services finished successfully

Details

[redacted]

```
Name Server          100.0%
  Execute Data Backup Finished
Index Server           0.0%
  Execute Data Backup In Progress
Statistics Server      0.0%
  Execute Data Backup In Progress
XSEngine             100.0%
  Execute Data Backup Finished
```

[redacted]

```
Index Server          69.0%
  Execute Data Backup In Progress
```

[redacted]

```
Index Server          53.0%
  Execute Data Backup In Progress
```

Text-file based log and statistics output of the backup script

The backup script will always generates three output files during script execution the names and locations of these files can be configured. The default location is the directory containing the diagnosis/trace files of the database, so that the output files of the backup script are also accessible via SAP HANA Studio → administration console → tab "diagnosis files".

Log output

During script execution, the backup script writes logging entries into a file whose name is defined in script parameter ${SCRIPT_LOG}. This text file contains

- Information on the planned backup run as is displayed in the "splash screen", i.e. system information, size and execution time of previous backup, free space on backup device etc.
- Optionally full parameterization of the backup script for this particular execution, including command-line parameters and parameters set within the script
- Log messages during backup execution, time stamps of start and end of data backup
- In case of severe errors occurring during script execution, information on these errors.

In addition, the actual data backup execution via hdbsql will write output into a second text file whose name is given in script parameter ${BACKUP_LOG}. If the backup completes without issues, this text file will be empty. If errors occur during backup execution, the file will contain error messages.

Statistics output

If a database backup was executed successfully in the course of script execution, statistics information concerning this backup is written to a text file whose name is defined in parameter ${TIMING_MEASUREMENTS}.

This text file is a comma-separated file (field delimiter is the comma) with fields

- TIME_STAMP: time stamp of start of backup
- BACKUP_SIZE: size of the full data backup on disc in kB
- REAL_TIME: run time of backup execution (measured using linux “time” command
- USER_TIME: time waited for user input during backup execution (again from “time”)
- SYSTEM_TIME: system CPU time during backup execution

Table based log and statistics output of the backup script

If enabled, the backup script creates and populates two tables in SAP HANA Database, one containing log information of the backup script execution; the other one containing information of the actual database backup if it happened; reasons for the data backup not to happen can be usage of parameter -c (only create backup of configuration files) or severe errors encountered in script execution prior to executing the actual data backup.

All of these tables reside in the schema specified via parameter ${STATS_SCHEMA}. The script only writes to these tables if parameter ${WRITE_STATS_TO_TABLES} == TRUE.

Log output

The log output is written to a table for every execution of the backup script. We create and populate a table with the following fields:

- TIME_STAMP, VARCHAR(20): time stamp of beginning of backup execution
- DATE, VARCHAR(20): the date portion of the above time stamp
- TIME, VARCHAR(20): the time portion of the above time stamp
- FILE_SUFFIX, VARCHAR(128): file suffix used for script execution, i.e. either day of week or suffix passed via command line parameter --suffix.
- RETURN_CODE, VARCHAR(4): Return code of the backup script
- RETURN_CODE_TEXT, VARCHAR(256): Meaning of return code

In default parameterization, the table name is Z_SCRIPT_BACKUP_RUNS. The actual name is specified via parameter ${LOG_TABLE}.

Statistics output

The statistics output is written to a table only if the data backup could be completed successfully. We create and populate a table with the following fields:

- BACKUP_ID, BIGINT: the BACKUP ID of the data backup as used in SAP HANA Database’s own monitoring table PUBLIC.M_BACKUP_CATALOG
- TIME_STAMP, VARCHAR(20): the time stamp of beginning of backup execution
- DATE, VARCHAR(20): the date portion of the above time stamp
- TIME, VARCHAR(20): the time portion of the above time stamp
- BACKUP_SIZE, BIGINT: the size of the full data backup in kB
- REAL_TIME, INT: the overall time needed for backup execution, measured using Linux command “time”
- USER_TIME, INT: the user CPU time needed for backup execution (from “time”)
- SYSTEM_TIME, INT: the system CPU time needed for backup execution (from “time”)

It should be noted that the time stamp in the statistics output table is not identical to any time stamp in SAP HANA Database's backup monitoring tables M_BACKUP_CATALOG or M_BACKUP_CATALOG_FILES. It's a time stamp recorded by the backup script.

In default parameterization, the table name is Z_SCRIPT_BACKUP_STATISTICS. The actual name is specified via parameter ${STATS_TABLE}.

Ini-file customizing

After a database recovery, you might need to be able to restore configuration file changes that have been implemented on the source system of the backup. This is particularly interesting in the case of a system copy.

After a backup execution, the script will log all configuration file values that deviate from the default value into a table of the following layout:

- BACKUP_ID, BIGINT: the backup ID of the data backup we have just created
- FILE_NAME, VARCHAR(256): the file name of the .ini file
- LAYER_NAME, VARCHAR(16): the layer of the configuration value (DEFAULT, SYSTEM, HOST)
- TENANT_NAME, VARCHAR(256): the name of the tenant DB
- HOST, VARCHAR(64): the name of the host for the given configuration file
- SECTION, VARCHAR(128): Configuration section within the file
- KEY, VARCHAR(128): The parameter name
- VALUE, VARCHAR(5000): the parameter value

We log all entries for which LAYER_NAME != 'DEFAULT'.

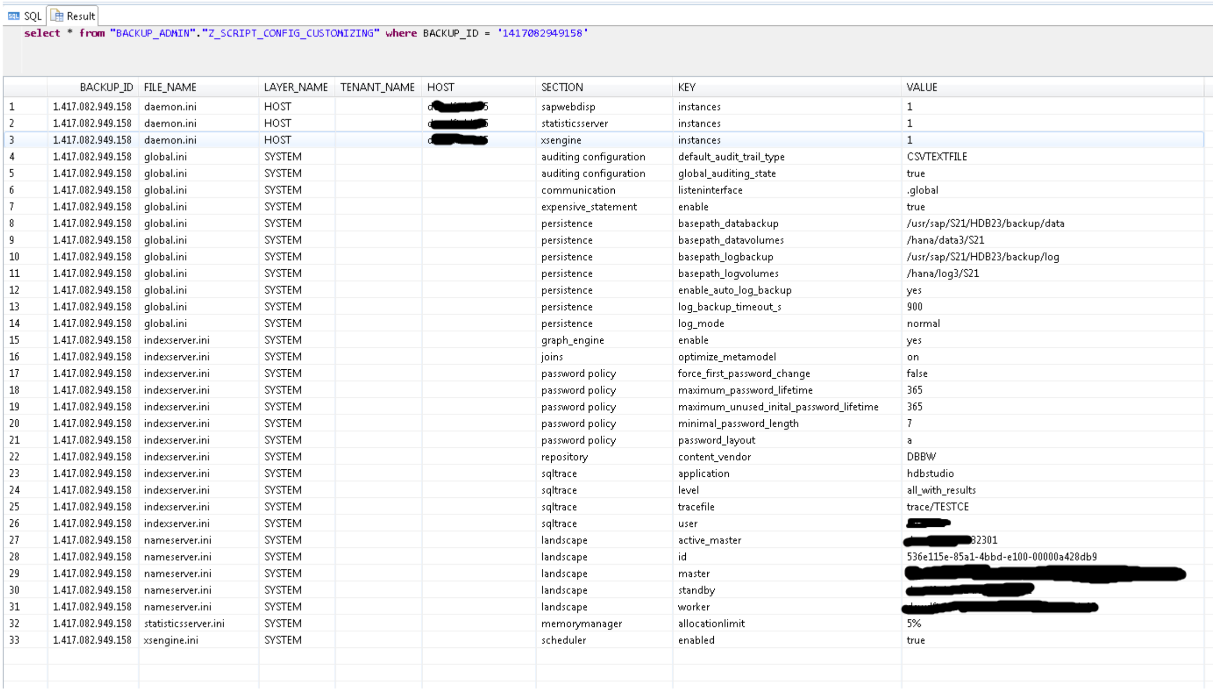
You can retrieve the customizing values for a given data backup with a given BACKUP_ID by running a simple query such as (for BACKUP_ID=1417082949158 – see screenshot below for output):

```
select * from "BACKUP_ADMIN"."Z_SCRIPT_CONFIG_CUSTOMIZING" where BACKUP_ID =
'1417082949158'
```

In the script implementation, we record the configuration changes into the table before the backup starts. At this point in time, we do not yet know the backup ID of the data backup we will create. For this reason, we initially write all entries with BACKUP_ID = 0. Once the backup has finished successfully, we update the BACKUP_ID to the actual backup ID.

This mechanism ensures that the data backup contains all configuration customizations at the time of creating the backup (to be on the safe side: between us recording the configuration customization and the actual start of the data backup, there may be a few milliseconds in which it is technically possible that the DB configuration is changed). In a recovered data backup, you can thus see all configuration customizations of the source system by searching our logging table for entries with BACKUP_ID = 0.

We also list the configuration customizing in the main script log file, so that they can be viewed independently from a running database. Note that we only log the customizing into the script log if logging into database tables is enabled. The log file does not contain a tabular listing of these customizings, but rather the SQL commands you might run in order to implement these changes in an SAP HANA system.



```
select * from "BACKUP_ADMIN"."Z_SCRIPT_CONFIG_CUSTOMIZING" where BACKUP_ID = '1417082949158'
```

	BACKUP_ID	FILE_NAME	LAYER_NAME	TENANT_NAME	HOST	SECTION	KEY	VALUE
1	1.417.082.949.158	daemon.ini	HOST		[redacted]	sapwebdisp	instances	1
2	1.417.082.949.158	daemon.ini	HOST		[redacted]	statisticsserver	instances	1
3	1.417.082.949.158	daemon.ini	HOST		[redacted]	xsengine	instances	1
4	1.417.082.949.158	global.ini	SYSTEM			auditing configuration	default_audit_trail_type	CSVTEXTFILE
5	1.417.082.949.158	global.ini	SYSTEM			auditing configuration	global_auditing_state	true
6	1.417.082.949.158	global.ini	SYSTEM			communication	listeninterface	.global
7	1.417.082.949.158	global.ini	SYSTEM			expensive_statement	enable	true
8	1.417.082.949.158	global.ini	SYSTEM			persistence	basepath_databackup	/usr/sap/S21/HDB23/backup/data
9	1.417.082.949.158	global.ini	SYSTEM			persistence	basepath_datavolumes	/hana/data3/S21
10	1.417.082.949.158	global.ini	SYSTEM			persistence	basepath_logbackup	/usr/sap/S21/HDB23/backup/log
11	1.417.082.949.158	global.ini	SYSTEM			persistence	basepath_logvolumes	/hana/log3/S21
12	1.417.082.949.158	global.ini	SYSTEM			persistence	enable_auto_log_backup	yes
13	1.417.082.949.158	global.ini	SYSTEM			persistence	log_backup_timeout_s	900
14	1.417.082.949.158	global.ini	SYSTEM			persistence	log_mode	normal
15	1.417.082.949.158	indexserver.ini	SYSTEM			graph_engine	enable	yes
16	1.417.082.949.158	indexserver.ini	SYSTEM			joins	optimize_metamodel	on
17	1.417.082.949.158	indexserver.ini	SYSTEM			password policy	force_first_password_change	false
18	1.417.082.949.158	indexserver.ini	SYSTEM			password policy	maximum_password_lifetime	365
19	1.417.082.949.158	indexserver.ini	SYSTEM			password policy	maximum_unused_inital_password_lifetime	365
20	1.417.082.949.158	indexserver.ini	SYSTEM			password policy	minimal_password_length	7
21	1.417.082.949.158	indexserver.ini	SYSTEM			password policy	password_layout	a
22	1.417.082.949.158	indexserver.ini	SYSTEM			repository	content_vendor	DBBW
23	1.417.082.949.158	indexserver.ini	SYSTEM			sqltrace	application	hdbstudio
24	1.417.082.949.158	indexserver.ini	SYSTEM			sqltrace	level	all_with_results
25	1.417.082.949.158	indexserver.ini	SYSTEM			sqltrace	tracefile	trace/TESTCE
26	1.417.082.949.158	indexserver.ini	SYSTEM			sqltrace	user	[redacted]
27	1.417.082.949.158	nameserver.ini	SYSTEM			landscape	active_master	[redacted]
28	1.417.082.949.158	nameserver.ini	SYSTEM			landscape	id	536e115e-85a1-4bbd-e100-00000a428db9
29	1.417.082.949.158	nameserver.ini	SYSTEM			landscape	master	[redacted]
30	1.417.082.949.158	nameserver.ini	SYSTEM			landscape	standby	[redacted]
31	1.417.082.949.158	nameserver.ini	SYSTEM			landscape	worker	[redacted]
32	1.417.082.949.158	statisticsserver.ini	SYSTEM			memorymanager	allocationlimit	5%
33	1.417.082.949.158	xsengine.ini	SYSTEM			scheduler	enabled	true

In default parameterization, the table name is Z_SCRIPT_CONFIG_CUSTOMIZING. The actual name is specified via parameter ${CONFIG_TABLE}.

Monitoring query

The log table (default name “Z_SCRIPT_BACKUP_RUNS”) and the backup information table (default name “Z_SCRIPT_BACKUP_STATISTICS) can be joined via field TIME_STAMP in both tables. Table Z_SCRIPT_BACKUP_STATISTICS can be joined to the standard backup monitoring tables in SAP HANA Database (PUBLIC.M_BACKUP_CATALOG and PUBLIC.M_BACKUP_CATALOG_FILES) on field BACKUP_ID in all three tables.

Two example queries joining all four tables are given below. Keep in mind that each backup execution creates several backup files (indexserver, statisticsserver, …), so M_BACKUP_CATALOG_FILES has more entries than the other tables.

Both queries do not specify the database schema for the tables created by the backup script, i.e. they assume that the tables are located in the user schema of the user running the query.

The first query uses the execution of the backup script as reference (table Z_SCRIPT_BACKUP_RUNS) and adds data backup information from the other tables via left outer joins. If the data backup was not performed successfully, the fields coming from the other tables will be filled with NULL values.

```
select * from
  ( select * from
       (
       select * from
       Z_SCRIPT_BACKUP_RUNS as a
       left outer join
       Z_SCRIPT_BACKUP_STATISTICS as b
       on a.TIME_STAMP = b.TIME_STAMP
       ) as c
       left outer join PUBLIC.M_BACKUP_CATALOG as d
       on c.BACKUP_ID = d.backup_ID
  ) as e
  left outer join PUBLIC.M_BACKUP_CATALOG_FILES as f
  on e.BACKUP_ID = f.BACKUP_ID;
```

The second query only looks at full data backups that have been completed successfully, and then only at those that have been triggered via the backup script. The reference table is Z_SCRIPT_BACKUP_STATISTICS, information from the backup catalog tables is added via inner joins. Finally, script execution information is also added via an inner join.

```
select * from
 ( select * from
   ( select * from
          Z_SCRIPT_BACKUP_STATISTICS as a
          inner join
          PUBLIC.M_BACKUP_CATALOG as b
          on a.BACKUP_ID = b.BACKUP_ID
      )
      as c
      inner join
      PUBLIC.M_BACKUP_CATALOG_FILES as d
      on c.BACKUP_ID = d.BACKUP_ID
  )
  as e
  inner join
  Z_SCRIPT_BACKUP_RUNS as f
  on e.TIME_STAMP = f.TIME_STAMP;
```

Running the Script

Command Line Options

The backup script offers the following command line options

Name	Description
-h / --help	Display usage information and exit (regardless of any other command line parameters given)
--config-file	Name of configuration file with script parameterizations to be used for this backup run.
Backup modes	
-w	Run backup with unique backup file name for each day of week (i.e. overwrite old backups weekly) Suitable for scheduling backups.
-o	Run backup with retention time = 2 days, i.e. overwriting backups every other day. Suitable for scheduling backups
--retention=<value>	Keep backups for <value> days, i.e. overwrite old backups every <value> days. Generates backup files named backup_<SID>_COUNT_<i> where i <= 0 < value Suitable for scheduling backups Note: there must not be any white space on either side of the '=' sign.
--suffix=<value>	Create backup files with file name backup_<SID>_<value> Intended for manual backups, e.g. preceding revision upgrade Note: there must not be any white space on either side of the '=' sign.
Further Options for running data backups	
-t	Test mode: do not create or delete backup files, i.e. do not create data backup, do not create configuration file backup, … Does write log messages into file ${SCRIPT_LOG}
-q	Suppress wait time and information output (recommended if operated in batch mode)
-d	Only create data backup, do not back up configuration files
-c	Only back up configuration files, do not run database backup
-p	Add script parameterization and command line switches into script log file
Housekeeping for backup files	
-ld	List data backups that have been completed successfully (we only list the backup file for the indexserver to keep the list of output files small) If option -cl is given in addition, only list those data backups for which the backup file still exists in its original location: - If several backups of the same name have been created, only list the latest one - And in addition check that the corresponding backup file still exists If option -cl is not given, we list all data backups that are listed as "successful" in the database's backup catalog.

Name	Description
-ll	List log backups that do not contain log entries more recent than the redo_log_position of a given data backup. The backup ID of that data backup has to be given with command line option --backup-id or switch -od has to be used to force automatic determination of the backup ID of the oldest data backup. The preferred data backup ID can be determined in a run of the script using option -ld. If option -cl is given in addition, only those log backup files will be listed that still exist in the location stored in the backup catalog (clean list) This run is useful to determine those log backups that can be safely removed if the oldest backup that can be restored is that of the given backup ID New with version 03 of the script: files of the backup catalog are also included in the query, see SAP Notes 1812980 and 1852242.
-cd	Delete backup entries from backup catalog and data/log backups from disk that are older than a given backup ID. Can be combined with --backup-id -od (and then optimally also with -cl). The combination -cd -od -cl will delete all log backups from disk and all entries from the backup catalog that are not needed to recover the oldest successful data backup whose backup files are still available in their original file system location. It uses the SQL syntax BACKUP CATALOG DELETE ALL BEFORE BACKUP_ID <ID> WITH FILE
-cl	Produce cleaned list of data or log backups in combination with options -ll or -ld.
-od	When creating list of deletable log backups, determine automatically the backup ID of the oldest data backup. Most useful in conjunction with option –cl, because then we'll determine the backup ID of the oldest data backup that still exists on disk.
--backup-id=<value>	ID of data backup for which all "older" log backup files shall be listed with option -ll.
--output_file=<file>	Per default, options -ld or -ll write their output to std out. This output can be explicitly redirected to the named output file.

If no command line parameters are given, the backup script will run in "normal" mode, i.e. it will

- create a database backup and a backup of the configuration files;
- create backup files for current weekday;
- Use wait time and display information during wait as specified in the parameterization within the script.

Where it makes sense, command line options can be combined. The posix-style of combining command line options (e.g. -tqd instead of -t -q -d) is not supported.

Examples for Backup Creation

Scheduling a daily backup with a retention time of three days

Scheduling a data backup with a retention time of three days works via option --retention=3

When scheduling a data backup, e.g. via cron, the splash screen does not need to be displayed. So we use the switch -q to suppress the wait time and splash-screen display. At the same time, one typically wants to print all script parameterization into the script log file, so we use switch -p as well. The result is the following:

```
backup.sh --config-file=<config_file.cfg> -q -p --retention=3
```

Running a backup interactively from the command line

When running an interactive backup, one typically wants to specify a unique backup file name. For this we can use the option --suffix=<name>. This will create backup files named backup_<SID>_<name>_databackup_*.

Splash screen display is recommended in interactive mode, and do take the default 20 seconds to read the displayed information, e.g. available disk space in the backup location (so we do not use switch -q). Printing of script parameterization and command line options into the log file may be switched on (-p).

```
backup.sh --config-file=<config_file.cfg> -p --suffix=<name>
```

Examples for displaying the backup catalog

Showing all existing data backup files

Switch -ld is used to display data backup files from the catalog. Without further options, this would simply list all data backups that have been executed in the past (assuming nobody deletes from the backup catalog tables).

In order to show only existing data backup files, add option -cl. The following will print this list to std output, typically to the command line.

```
backup.sh --config-file=<config_file.cfg> -ld -cl
```

The example can be modified to write to a given text file by adding option --output-file=<file_name>. this option can be added to all examples of listing data or log backup files.

```
backup.sh --config-file=<config_file.cfg> -ld -cl --output-file=<file_name>
```

Showing all existing log backup files

Log backup files can be listed using switch -ll. This will create a list of all log backup files in the backup catalog – most of these files will not exist anymore.

By adding command line switch -cl we can make sure to list only those log backup files that still exist on disk in the log backup destination.

```
backup.sh --config-file=<config_file.cfg> -ld -cl
```

Showing all existing log backup files that are not required to restore a given data backup

The above example lists all existing log backup files. For housekeeping purposes, we cannot simply move or delete all existing log backup files but only those files that cannot be used anymore to restore the oldest data backup that shall be restorable at the point in time of the housekeeping activity.

We first have to determine the backup ID of the oldest data backup we still want to be able to restore (e.g. using the backup script with switch -ld).

Once we have this ID, we can run the script to display all existing log backup files (switches -ll and -cl) plus adding option --backup-id=<DATA_BACKUP_ID>.

So the list of obsolete log backup files can be created via

```
backup.sh --config-file=<config_file.cfg> -ll -cl --backup-id=<DATA_BACKUP_ID> --
output-file=<file_name>
```

Showing all existing log backup files that are obsolete given the existing data backup files

In the above example we still had to manually determine the backup ID of the oldest data backup we want to be able to restore. In backup schemas with a fixed retention time, we overwrite data backups every <n> days. Thus we have automatic housekeeping of data backup files on the backup device (unless we also created backups manually). In the case that we want to be able to restore all data backups that still exist on the backup device, the backup script can automatically determine the backup ID of the oldest existing data backup and use that backup ID to display the list of all obsolete log backups. This can be achieved by adding switch -od.

```
backup.sh --config-file=<config_file.cfg> -ll -cl -od --output-file=<file_name>
```

Moving obsolete log backups to a long-term backup store

Once we have determined the list of obsolete log backup files (written to a file named <file_name>), we can move the log backup files to the long-term backup store. The following is an example of such a move operation (where in fact we are deleting the files. We write a log of the deletion into a time-stamped log file):

```
while read FILE; do
    if [ -e "${FILE}" ]; then
      rm -v "${FILE}";
    else
      echo "(EE) Log Segment ${FILE} does not exist";     fi;
  done < <file_name> &> log_backup_deletion_log_$(date +%F_%T).txt
```

New: Deleting log backups files older than a given data backup ID from catalog and file system

With version 04 of the script, we also adapt HANA's SQL command to delete from the backup catalog: BACKUP CATALOG DELETE (see the SAP HANA administration guide or the SAP Press book on SAP HANA administration for more details). The statement can be invoked to only delete from the backup catalog; or you may specify the "WITH FILE" clause to also remove files from disk (if they still exist in the original file system location). Whether or not files shall be deleted from disk is governed in this script by the configuration parameter CATALOG_DELETE_FROM_DISK.

Deleting from the backup catalog (and potentially from disk) is invoked using the command line switch -cd (for catalog delete).

You can combine option -cd with the command line parameter --backup-id=<DATA_BACKUP_ID>:

```
./backup.sh --config-file=<my_config_file.cfg> -cd --backup-id=<DATA_BACKUP_ID>
```

In this case, the script will invoke the following SQL command in the HANA system:

BACKUP CATALOG DELETE ALL BEFORE BACKUP_ID <DATA_BACKUP_ID> [WITH FILE]

New: Deleting all log backup files and catalog entries that are obsolete given the existing data backup files

The new command line switch -cd (for deleting log backups via the BACKUP CATALOG DELETE command) can also be combined with the switches -od -cl. In this case, the system will determine the oldest full data backup that still resides in its original location on disk. The backup ID of this data backup will be used in the BACKUP CATALOG DELETE command.

```
./backup.sh --config-file=<my_config_file.cfg> -cd -od -cl
```

Return Codes

The backup script will exit with return code zero, if no errors are encountered. In case of errors, the type of error can be determined from the return code.

Return Code	Meaning
0	Regular program termination. Including program modes in which no backup will be created such as -h (help) and –t (test)
11	Invalid command line option was given
12	Attempt to specify command line options -c and -d at the same time which is not possible
13	Attempt to specify command line options -w, -o, --retention= or --suffix= at the same time which is not possible
14	No value specified for option --suffix=<value>
15	No value or non-numeric value given for command line option --retention=<value>
16	Options -ld and -ll have been combined which is not possible
17	Options -cd and -ll or -ld have been combined, which is not possible
18	No configuration file given or configuration file not found
1	Backup base directory did not exist and could not be created
2	Data backup directory did not exist and could not be created
3	Config backup directory did not exist and could not be created
4	Running data backup via hdbsql failed. Note: hdbsql will exit with an error code other than zero if the connection fails (invalid credentials or wrong host name/port); or if the execution of individual SQL statements fail. We always exit the script with error code 4; the hdbsql error code is written to the output file ${SCRIPT_LOG} and error messages from hdbsql may be found in output file ${BACKUP_LOG}.
20	Plausibility checks for in-script parameterization failed. We currently check: - Length of SID is 3 characters - Instance is two-digit numeric value - Hostname has a length greater than zero and host name can be pinged - Instance path /usr/sap/<sid>/<instance> exists - If the writing of statistics to DB tables is enabled, a target schema for these tables has to be given If any of the above criteria is violated, the script will exit with error code 20. Details of the error will be logged into the script log file and (if possible) into the backup run logging table.
30	When listing log backups (option -ll) or deleting log backups via catalog (option -cd), either no backup ID was given, or the system failed to determine an appropriate backup ID (e.g. because no data backup exists)
31	When listing log backups (option -ll) or deleting log backups via catalog (option -cd): For the given or determined BACKUP ID, the system could not find a successful data backup. That could mean that the BACKUP ID is invalid, or it belongs to a data backup that failed.
99	You did not acknowledge one of the three required statements that tell people that this script is really no SAP-supported software

Scheduling backups via cron

In order to schedule regular backups via cron, add an appropriate line to the <sid>adm's crontab (<sid> being the SAP System ID of the SAP HANA Database instance). For example, in order to run a daily database backup at 10.15 pm, enter the following crontab entry (adding the -q option to avoid wait time and to suppress "splash screen" output):

```
15 22 * * * /<path_to_backup_script>/backup.sh –q
```

Further Information

Creating a database account for running backups

If you would like to create a dedicated database account for running backups of your SAP HANA Database, you can create a database user that has the following two privileges:

- System privilege "BACKUP ADMIN"
- System privilege "CATALOG READ"

Typically, you will want to exempt this user from the password life time rules (meaning: you will neither have to change the initial user password when you log on for the first time with this user nor will you have to change the password of this user in regular intervals).

You may create a user to run database backups by pasting the following SQL statements into a SQL editor for database user SYSTEM (or for another database user with sufficient privileges). Please note that this is only a technically working example that should give enough information for you to create an appropriate user in accordance with your company's security guidelines.

```
/*** create a role containing the necessary privileges */
/*   for creating and restoring backups ***/
Create role BACKUP_ADMIN_ROLE;
/* System privilege required to create/restore backups */
grant BACKUP ADMIN to BACKUP_ADMIN_ROLE;
/* System privilege that allows reading all catalog metadata */
grant CATALOG READ to BACKUP_ADMIN_ROLE;
/*** Now create the user ***/
create user BACKUP_ADMIN password <password_of_your_choice>;
/* grant the role to this user */
grant BACKUP_ADMIN_ROLE to BACKUP_ADMIN;
/* and exempt the user from the password lifetime rules */
alter user BACKUP_ADMIN disable password lifetime;
```

Using hdbuserstore

hdbuserstore is a secure store for logon data of an SAP HANA Database that comes with an installation of the hdbclient package. In order to use hdbuserstore locally on the SAP HANA Database server, the hdbclient package in the correct flavor (typically Linux x86 64 bit) has to be installed on the database server. Entries in the SAP HANA user store are identified by a user store key. Entries contain all information required for logging in to the database server.

In order to create an entry in hdbuserstore for a given database user of a given database instance, run the following command (assuming hdbuserstore is installed in /usr/sap/hdbclient/):

```
/usr/sap/hdbclient/hdbuserstore SET <KEY> <ENVIRONMENT> <USER> <PASSWORD>
```

Where:

- <KEY> is the identifier for the entry; can be freely chosen
- <ENVIRONMENT> defines the SAP HANA Database system in the form <hostname>:<port> where <port> is the SQL port of the indexserver process, i.e. 3<instance>15, <instance> being the two-digit instance number of the SAP HANA Database
- <USER> is the name of a SAP HANA Database user
- <PASSWORD> is that user's password in SAP HANA Database

Note that per default only the Operating System user who created a key in hdbuserstore will be able to use that particular key to identify against a SAP HANA Database.

For reasons of security, the preferred way of entering keys into the user store is to use the -i option (for interactive completion of the entry) for the password:

```
hdbuserstore -i SET <KEY> <ENVIRONMENT> <USER>
```

With this invocation, hdbuserstore will prompt you for the password, so that you do not need to specify it on the command line.

To see the details of a given key, run:

```
/usr/sap/hdbclient/hdbuserstore LIST <KEY>
```

To see further options of hdbuserstore, run:

```
/usr/sap/hdbclient/hdbuserstore –h
```

More information on hdbuserstore is included in the "SAP HANA Overall Security Guide" which is available in the SAP Library at http://help.sap.com/hana_appliance.

Other means to run backups

Backups in SAP HANA Database can be executed by many other means, e.g.

- Using SAP HANA Studio
- Using dbacockpit of a sufficiently recent NetWeaver-based system – the scheduling calendar of dbacockpit can then be used to schedule regular backups. Dbacockpit for SAP HANA Database SPS 03 is available starting with:
  - SAP NetWeaver BASIS 7.02 SP 10
  - SAP NetWeaver Business Warehouse 7.30 SPS 5
  - SAP Solution Manager 7.1 SP 4
- Third-party backup managers if certified to work with HANA, especially using the BACKINT interface. Contact your HANA hardware vendor for details of their particular solution.

We generally recommend scheduling database backups via the scheduling functionality of dbacockpit and only using other means if this is really required (e.g. if there is no SAP system available in your landscape with a sufficiently recent NetWeaver SP).

Contact

Author: Richard Bremer, Product Management SAP HANA
I will not respond to e-mails sent to me regarding this script.