FUJITSU Software BS2000 TASKDATE

Version 20.0B
May 2019

Readme file

1 Overview

The TASKDATE tool enables the user to apply a user-defined temporal environment to his/her programs.

Once the desired temporal environment has been set, an offset time (GTIME, GDATE, time queries from high-level language programs, special job variables, predefined functions of SDF-P) is supplied at the user interfaces to enable the current time to be calculated.

All other time values will not be influenced by TASKDATE. Especially all privileged system components are still using the system time, e.g. DMS concerning time stamps in the catalog, JMS concerning starting time of repeat and date jobs. Furthermore, the time values which are supplied by the STCK command are also not affected.

For those situations in which the test includes the aforementioned products, the evaluation of catalog time stamps or the schedule-driven starting of batch jobs, or in which the time is determined directly by way of STCK, it is necessary to use VM2000 to set up a separate VM with a modified system time.

For all batch jobs not scheduled by a calendar, the simulated time can be specified externally- without the batch file having to be modified. Batch jobs which are entered by a batch job using simulated time receive the temporal environment of that job etc. This means that the user can test entire job networks in a modified temporal environment by using TASKDATE.

1.1 Target group

The TASKDATE tool is a general solution for users who wish to test their programs to ascertain whether these programs produce correct results in a preselected temporal environment.

1.2 Formulation of terms

The temporal environment set by the user is also referred here to as simulated time, and may be regarded as an alternative to the system time.

2 User interface

2.1 SET-TASK-CLOCK command

Domain: PROGRAM, PROGRAMMING-SUPPORT, JOB
Privileges: STD-PROCESSING

The user sets the simulated time by means of the /SET-TASK-CLOCK command.

The INTERFACES parameter serves to determine the user interfaces (GTIME, GDATE, time queries from high-level language programs, special job variables, predefined functions of SDF-P) at which the simulated time is to be made available. (The associated CTIME interface itself allows the processing of time stamps with a simulated time, and does not therefore need to be listed here). Both the local time and also the UTC are offset by the same amount, i.e. a given difference between local time and UTC is retained.

The SCOPE parameter is used to select whether the simulated time is to be applicable to the user's own task, or to batch jobs - which are no Calendar jobs - entered in the future, or to both. For batch jobs therefore, in addition to the facility to execute the command in the batch file, it is also possible to specify a simulated time from an external situation. In this case, SCOPE allows the user to specify that the batch job is to run in the same temporal environment as the entering task, or that the batch job is to start precisely with the specified simulated time.

When batch jobs are created in interleaved fashion, as soon as the batch jobs have been started they run with the same simulated time as their respective job-originating task.

For reasons of consistency, the command will not be executed while any program remains loaded.

2.1.1 Format

SET-TASK-CLOCK
CLOCK = *SYSTEM-TIME / *PARAMETERS(...)
*PARAMETERS(...)
| DATE = <date 10..10>
|,TIME = <time 8..8>
|,INTERFACES = list-poss(3): **<u>*PROGRAM</u>** / *SDF-P / *SPECIAL-JV
|,SCOPE = list-poss(2): **<u>*OWN-TASK</u>** / *ENTER-JOBS

2.1.2 Description of operands

CLOCK =
The CLOCK parameter serves to define which temporal environment is to apply: the system time or a simulated time.

CLOCK = *SYSTEM-TIME

This parameter value corresponds to resetting the temporal environment to the system time. The system time is supplied once again to all the possible user interfaces specified under INTERFACES, both in the case of the user's own task and also in the case of all batch jobs to be entered in the future. Any batch jobs which have already been created remain unaffected by this.

CLOCK = *PARAMETERS(...)
The DATE and TIME entries are used to specify the desired simulated time. The simulated time is supplied to all the interfaces which are specified in the INTERFACES parameter. The SCOPE parameter indicates the range of validity for the simulated time.

Any batch jobs which have already been created remain unaffected by the (new) simulated time.

DATE = <date 10..10>
Date specification in the format YYYY-MM-DD. The range of permitted values extends from 1950-01-01 to 2041-12-31. The simulated time may not leave the area of the TODR epoch which is defined in the system - cf. manual "System Administration, chapter TODR epochs”. The currently fixed TODR epoch can be determined using the command SHOW-SYSTEM-INFORMATION INFORMATION = *SYSTEM-TIME-PARAMETER.

TIME = <time 8..8>
Time specification in the format HH:MM:SS. The time specification relates to the local time, i.e. not to the UTC. The UTC is offset by the same amount as the local time.

INTERFACES =
The INTERFACES parameter serves to specify the user interfaces at which the simulated time is to be made available.

INTERFACES = *PROGRAM
Default value.
The simulated time is made available at the program interfaces.
These are:

- GTIME
- GDATE
- Time queries from high-level languages

INTERFACES = *SDF-P
The predefined SDF-P functions for the output of date and time supply the values for the simulated time. The SDF-P functions concerned are as follows:

- DATE( )
- DAY( )
- MONTH( )
- TIME( )
- DATE-VALUE( )
- ELAPSED-DAYS()

INTERFACES = *SPECIAL-JV
When the /SHOW-JV command is called for a special job variable having time-related contents, the simulated time is supplied. The special job variables concerned are as follows:

- $SYSJV.DATE - $SYSJV.DATUM
- $SYSJV.DAY - $SYSJV.TAG
- $SYSJV.MONTH - $SYSJV.MONAT
- $SYSJV.TIME - $SYSJV.ZEIT
- $SYSJV.DATE-ISO4
- $SYSJV.DATE-TIME-LONG
- $SYSJV.DATE-TIME-SHORT
- $SYSJV.DATE4

- $SYSJV.TIME-ZONE
- $SYSJV.TIMESTAMP
- $SYSJV.TOMORROW
- $SYSJV.YESTERDAY

SCOPE =
The SCOPE parameter specifies the range of validity for the specified simulated time.
SCOPE = *OWN-TASK
Default value.
The specified time applies to the user's own task from the point the command is issued, i.e. thereafter the task receives the time of an altered clock.

SCOPE = *ENTER-JOBS
The new simulated time applies to batch jobs which are entered after the command - and whose run is not scheduled by a Calendar. There is no inheritance of Calendar jobs.

It is necessary to differentiate between two situations here:

1) SCOPE=*ENTER-JOBS:
An implicit dependency is assumed amongst the batch jobs which are entered subsequently. The specified time begins running when the first such batch job actually starts its execution. The remaining batch jobs start with a simulated time which has the same time offset relative to the system time as also applies to execution of the first batch job. The result of this is that if these batch jobs simultaneously query the time, they will also receive the same simulated time. In this manner it is possible to test job networks with a precisely defined simulated start time (example 3, see below). In the special situation where there is only one batch job, this of course begins exactly with the specified time.
2) SCOPE=(*OWN-TASK,*ENTER-JOBS):
The simulated time is immediately applicable to the user's own task. It is also "bequeathed" to batch jobs which are started in the future. This means that these batch jobs have the same simulated time and therefore have the same offset relative to the system time (example 2, see below).

The characteristic behavior in these two situations may also be illustrated as follows: Let us assume that there are different virtual clocks. One execution unit (batch job or task) can be assigned to one of these virtual clocks, in which case time queries are then answered using this virtual clock time. The clocks are set by means of the /SET-TASK-CLOCK command, and begin to run as soon as one of the assigned execution units is running or is started.

In the former case (SCOPE=*ENTER-JOBS), the first batch job starting causes the clock to start running. In the latter case (SCOPE=(*OWN-TASK,*ENTER-JOBS)), however, the task which the command calls is already running, i.e. the virtual clock starts running immediately.

Additional note: When batch jobs are called in interleaved fashion, the job-originating batch job is already always running with the simulated time, i.e. the virtual clock is already running in the "inheritance situation".

2.2 SHOW-TASK-CLOCK command

Domain:	PROGRAM PROGRAMMING-SUPPORT JOB
Privileges:	STD-PROCESSING

The user can use the /SHOW-TASK-CLOCK command to obtain information whether a simulated time has been defined for a job or a task. If this is the case, then the time difference with respect to the system time and its simulated time are output.

2.2.1 Format

```
SHOW-TASK-CLOCK
--------------------------------------------------------------
JOB-IDENTIFICATION = *TSN(...) / *MONJV(...)

     *TSN(...)
        | TSN = *OWN / *ALL / <alphanum-name 1..4>
     *MONJV(...)
        | MONJV = <filename 1..54 without-gen>
```

2.2.2 Description of operands

JOB-IDENTIFICATION =
Indicates how the job or the task is specified, for which information relating to the simulated time is to be output.

JOB-IDENTIFICATION = *TSN(...)
Default value. The job or the task is specified by way of the TSN.

TSN = *OWN
The information about the simulated time is to be output for the user's own task. This is the default value.

TSN = *ALL
The information about the simulated time is to be output for all tasks under the user's own user ID (under $TSOS for all tasks on the system)

TSN = <alphanum-name 1..4>
The information about the simulated time is to be output for the task or the job having the specified TSN.

JOB-IDENTIFICATION = *MONJV(...)
The job or the task is specified by way of the monitoring job variable.

MONJV = <filename 1..54 without-gen>
The information about the simulated time is to be output for the job having the specified MONJV

2.2.3 Output format

TSN	TASK-CLOCK	STATE	CLOCK FOR ENTER JOBS	SET BY	INTERFACES
<tsn>	<date+time>	RUNNING	<date+time>	<tsn> *SAME	P /JV / SDF
	*SYSTEM	INACTIVE	*SAME *NONE	(<tsn>)	

TSN : TSN of the task/job for which the output is taking place

TASK-CLOCK : time with which the current task is running or with which the job will run.
<date+time> : Date and time of a simulated time
*SYSTEM : Task/job is running with system time

SET BY : TSN of the task which specified the simulated time for the current task
<tsn> : The specifying task is similarly still working with this simulated time
(<tsn>) : The task which specified the time for the current task no longer exists or is in the meantime using a different task time

STATE : Supplementary information for a simulated task time
RUNNING : The simulated time has already started running
INACTIVE : The simulated time has not yet started running

CLOCK FOR ENTER JOBS : Time specification for ENTER jobs if this is used to start the current task
<date+time> : Date and time of the simulated time for ENTER jobs
*SAME : ENTER jobs run with the same simulated time as the current task/job
*SYSTEM : ENTER jobs run with system time

INTERFACES : Interfaces for which a simulated time applies
P / JV / SDF : By analogy with the INTERFACES operand of the /SET-TASK-CLOCK command (where P corresponds to *PROGRAM, JV to *SPECIAL-JV, and SDF to *SDF-P)

2.3 Examples

2.3.1 Example 1: Task with simulated time (*OWN-TASK)

System time	Time applying to task	Commands and output	
1996-11-28 11:00:00	1996-11-28 11:00:00	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 11:00:00	(1)

1996-11-28
11:05:00
	1996-11-28
	11:05:00
		/SET-TASK-CLOCK CLOCK=*PARAMETERS(
		DATE=1997-02-01,TIME=03:00:00,
		INTERFACES=(*PROGRAM,*SPECIAL-JV))
			(2)

1996-11-28 11:07:00	1997-02-01 03:02:00	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 03:02:00	
1996-11-28 11:08:05	1997-02-01 03:03:05	/COPY-FILE X,Y	(3)
1996-11-28 11:08:30	1997-02-01 03:03:30	/SHOW-FILE-ATTR Y, INFORMATION=*PARAM(HISTORY=*YES) % ----- HISTORY ------ % CRE-DATE = 1996-11-28 ... % CRE-TIME = 11:08:05 ... %	(4)
1996-11-28 11:10:04	1997-02-01 03:05:04	/ENTER-JOB BATCHDATEI.1, JOB-NAME=BATCH1 % JMS0066 JOB 'BATCH1' ACCEPTED ON 96-11-28 AT 11:10, TSN = 1TSN	(5)
1996-11-28 11:15:00	1997-02-01 03:10:00	/SET-TASK-CLOCK CLOCK=*SYSTEM-TIME	(6)
1996-11-28 11:15:01	1996-11-28 11:15:01	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 11:15:01	

.

.

System time	Time applying to task	Commands and output	
1996-11-28 16:44:23	1996-11-28 16:44:23	/SET-LOGON-PARAMETERS % JMS0154 ‘TEST’ LOGGED ON FOR ‘NC’. JOB-NAME ‘BATCH1’. CALLER ‘TSN 2TSN’. TID 11223344	(7)
1996-11-28 16:44:24	1996-11-28 16:44:24	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 16:44:24	

(1) The task starts with the system time as the valid time.

(2) The time applicable to the task immediately after the command is called is 03:00:00 hours on February 1st.1997. The default value *OWN-TASK applies to the SCOPE parameter. The clock at the user program interfaces and for interrogation of the special job variables subsequently runs with an appropriately offset time for the task.

(3) A new file Y is created by copying the existing file X. the catalog entries for creation date and time are written using the system time …

(4) ... and also displayed using the system time by means of the /SHOW-FILE-ATTR command.

(5) A batch job is entered, its TSN is 1TSN.The message JMS0066 is issued using the system time.

(6) The time applicable to the task after the command was called is once again the system time. As far as the task is concerned, the time behaves in the same way as before the first /SET-TASK-CLOCK command.

(7) The batch job starts. Its clock indicates the system time because only *OWN-TASK was specified for the SCOPE operand in the /SET-TASK-CLOCK command.

2.3.2 Example 2: Task and batch jobs with the same simulated time (*OWN-TASK, *ENTER-JOBS)

System time	Time applying to task	Commands and output (interactive task)	
1996-11-29 10:16:10	1996-11-29 10:16:10	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 10:16:10	(1)
1996-11-29 10:16:11	1996-11-29 10:16:11	/SET-TASK-CLOCK CLOCK=*PARAMETERS(DATE=2000-01-01,TIME=16, INTERFACES=(*PROGRAM,*SPECIAL-JV), SCOPE=(*OWN-TASK,*ENTER-JOBS))	(2)
1996-11-29 10:16:12	2000-01-01 16:00:01	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 16:00:01	
1996-11-29 12:05:24	2000-01-01 17:49:13	/ENTER-JOB BATCHDATEI.1, JOB-NAME=BATCH2 % JMS0066 JOB 'BATCH2' ACCEPTED ON 96-11-29 AT 12:05, TSN = 2TSN	(3)
1996-11-29 12:05:55	2000-01-01 17:49:44	/ENTER-JOB BATCHDATEI.2, JOB-NAME=BATCH3 % JMS0066 JOB 'BATCH3' ACCEPTED ON 96-11-29 AT 12:05, TSN = 3TSN	(4)

System time	Time applying to task	Commands and output (Batch job with TSN 2TSN)	
1996-11-29 16:44:23	2000-01-01 22:28:12	/SET-LOGON-PARAMETERS % JMS0154 ‘TEST’ LOGGED ON FOR ‘NC’. JOB-NAME ‘BATCH2’. CALLER ‘TSN 1TSN’. TID 44332211	(5)
1996-11-29 16:44:24	2000-01-01 22:28:13	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 22:28:13	

System time	Time applying to task	Commands and output (Batch job with TSN 3TSN)	
1996-11-29 16:45:58	2000-01-01 22:29:47	/SET-LOGON-PARAMETERS % JMS0154 ‘TEST’ LOGGED ON FOR ‘NC’. JOB-NAME ‘BATCH3’. CALLER ‘TSN 1TSN’. TID 44332211	(6)
1996-11-29 16:45:59	2000-01-01 22:29:48	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 22:29:48	

(1) The task starts with the system time as the valid time.

(2) The time applicable to the task immediately after the command is called is 16:00:00 hours on January 1st 2000. The clock runs with a correspondingly offset time both for the task and also for batch jobs which are still to be created.

(3) The first batch job is entered, its TSN is 2TSN. The message JMS0066 is issued using the system time.

(4) The second batch job is entered, its TSN is 3TSN.The message JMS0066 is issued using the system time

(5) The first batch job is started. Its simulated time is 22:28:12 hours on January 1st 2000. The difference from the system time is identical to the difference which applied in the case of the associated /ENTER-JOB. The simulated time has apparently continued to run.

(6) The second batch job is started. Its simulated time is 22:29:47 hours on January 1st 2000, in other words also the simulated time, which has continued to run correspondingly, for the task which entered the job. Within the session, the reference to the simulated time of a job-originating task is retained regardless of when the batch job starts and when the job-originating task terminates.

2.3.3 Example 3: Task with simulated time for batch jobs (*ENTER-JOBS)

System time	Time applying to task	Commands and output (interactive task)	
1996-11-30 13:36:40	1996-11-30 13:36:40	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 13:36:40	(1)
1996-11-30 13:36:41	1996-11-30 13:36:41	/SET-TASK-CLOCK CLOCK=*PARAMETERS(DATE=2005-06-17,TIME=00:00:00, INTERFACES=(*PROGRAM,*SPECIAL-JV), SCOPE=*ENTER-JOBS)	(2)
1996-11-30 13:36:42	1996-11-30 13:36:42	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 13:36:42	
1996-11-30 15:17:54	1996-11-30 15:17:54	/ENTER-JOB BATCHDATEI.1, JOB-NAME=BATCH4 % JMS0066 JOB 'BATCH4' ACCEPTED ON 96-11-30 AT 15:17, TSN = 4TSN	(3)
1996-11-30 15:17:59	1996-11-30 15:17:59	/ENTER-JOB BATCHDATEI.2, JOB-NAME=BATCH5 % JMS0066 JOB 'BATCH5' ACCEPTED ON 96-11-30 AT 15:17, TSN = 5TSN	(4)

System time	Time applying to task	Commands and output (Batch job with TSN 4TSN)	
1996-11-30 18:34:23	2005-06-17 00:00:00	/SET-LOGON-PARAMETERS % JMS0154 ‘TEST’ LOGGED ON FOR ‘NC’. JOB-NAME ‘BATCH4’. CALLER ‘TSN 1TSN’. TID 44332211	(5)
1996-11-30 18:34:24	2005-06-17 00:00:01	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 00:00:01 /LOGOFF % EXC0419 /LOGOFF AT '18:34' ON 96-11-30, TSN = 4TSN	

System time	Time applying to task	Commands and output (Batch job with TSN 5TSN)	
1996-11-30 18:45:58	2005-06-17 00:11:35	/SET-LOGON-PARAMETERS % JMS0154 ‘TEST’ LOGGED ON FOR ‘NC’. JOB-NAME ‘BATCH5’. CALLER ‘TSN 1TSN’. TID 44332211	(6)
1996-11-30 18:45:59	2005-06-17 00:11:36	/SHOW-JV JV-CONTENTS=$SYSJV.TIME % 00:11:36	

(1) The task starts with the system time as the valid time.

(2) The system time retains its validity for the task. The simulated time specified applies only to batch jobs entered in the future, and it applies in such a way that the first of them starts with precisely the specified simulated time and the start of the others is then relative to the first, with the same difference from the system time.

(3) The first batch job is entered, its TSN is 4TSN.The message JMS0066 is issued using the system time.

(4) The second batch job is entered, its TSN is 5TSN. The message JMS0066 is issued using the system time. It is assumed by implication that the jobs having the TSNs 4TSN and 5TSN are intended to have the same simulated time.

(5) The first batch job is started. Its simulated time is 00:00:00 hours on June 17th 2005. The difference from the system time is calculated at the start time.

(6) The second batch job is started. Its simulated time is 00:11:35 hours on June 17th 2005 because it is started 11 minutes and 35 seconds later than the first batch job. Both the batch jobs run with the same simulated time.

2.3.4 Example 4: Simulated time differs from the current time only in respect of the year number (month, day and time of day are identical)

System time	Time applying to task	Commands and output (interactive task)	
1997-11-29 10:16:10	1997-11-29 10:16:10	/VDATE=SUBSTRING(DATE(),6,5) /SHOW-VAR VDATE % 11-29	(1)
1997-11-29 10:18:10	1997-11-29 10:18:10	/SET-TASK-CLOCK *P(DATE= 2000-&(VDATE),T=&(TIME()),	(2)

		I=(*P,*SDF,*SP),SC=(*O,*E))	
1997-11-29 10:19:11	2000-11-29 10:19:11	/SHOW-TASK-CLOCK	(3)

(1) Store only the month and the day in the SDF variable "VDATE" (displacement 6 from the beginning of a normal date variable and with a length of 5 => result: 11-29)

(2) Make the following settings by means of the /SET-TASK-CLOCK command:
- Date: year 2000 and current month and current day
- Time: current time of day
- Interfaces: All
- Scope: *OWN and *ENTER

(3) Use the /SHOW-TASK-CLOCK command to determine that the simulated time is exactly 3 years in advance of the current time

3 Scope of delivery

The scope of delivery for the product encompasses the following components:

Components	Standard name
OML of subsystem	SYSLNK.TASKDATE.200
OML of subsystem (hardware dependent)	SKMLNK.TASKDATE.200
SDF syntax file	SYSSDF.TASKDATE.200
MSG file for subsystem	SYSMES.TASKDATE.200
Subsystem declaration	SYSSSC.TASKDATE.200
Subsystem information file	SYSSII.TASKDATE.200
REP file for subsystem	SYSREP.TASKDATE.200
REP container	SYSRMS.TASKDATE.200
Product documentation (German)	SYSRME.TASKDATE.200.D
Product documentation (English)	SYSRME.TASKDATE.200.E
Release notice (German)	SYSFGM.TASKDATE.200.D
Release notice (English)	SYSFGM.TASKDATE.200.E
Macro library	SIPLIB.TASKDATE.200

4 Installation OF TASKDATE

4.1 Starting TASKDATE

Installation of product TASKDATE has to be done using the installation monitor IMON. Relevant information for this procedure is situated in the Release Note, the Delivery Note and the Readme file of TASKDATE. Either all necessary input parameters as the functional process are described in IMON manual.

After successful installation with IMON please consider the change dates between summer and winter time have also to cover all the simulated times which will be chosen. Any missing change dates have to be completed before starting TASKDATE – cf . chapter 4.3

Explicit starting of the subsystem is done via :
/START-SUBSYSTEM SUBSYSTEM-NAME=TASKDATE

4.2 Termination TASKDATE

Termination of the subsystem is effected by means of the command: /STOP-SUBSYSTEM SUBSYSTEM-NAME=TASKDATE. All those tasks and jobs for which a simulated time has been defined are compulsorily terminated by this means. (The reason is to avoid the situation whereby applications are initially supplied with simulated times and then, after the subsystem has been terminated, with the system time. Since this can result in inconsistencies in the data, all the subsystem users are terminated.) The system administrator can obtain information about those currently using the subsystem before termination of the subsystem (by means of the command /SHOW-TASK-CLOCK *TSN(*ALL)) and delay the termination until later if necessary.

The command /SHOW-SUBSYSTEM-STATUS SUBSYSTEM-NAME=TASKDATE indicates which tasks are currently running with a simulated temporal environment. So, in other words:

- Jobs for which a simulated time has been defined but which have not yet started are not displayed by this command.
- Tasks which have defined a simulated time only for ENTER jobs that are to be started (/SET-TASK-CLOCK ...,SCOPE=*ENTER) are similarly not displayed.

In error cases (task crashes) it may be necessary to terminate the subsystem via /STOP-SUBSYSTEM ..,FORCE=YES.

4.3 Adding time changeover information

An unambiguous time specification also includes information relating to whether summer or winter time is applicable. The time change data set for the system are also valid for the simulated time used by TASKDATE. Missing time change data have to be set by command /ADD-CHANGE-DATE and/or included to BS2000 parameter file in section GTIME if necessary.

Without this additional time change data, the correlation between local time and UTC time is not correct for the simulated time.

5 Further notes and restrictions

5.1 Testing database systems with TASKDATE

If TASKDATE is to be used as a vehicle for testing database user programs or databases, the following restrictions must be observed:

- testing should never be carried out on a database that is in productive use.

Reason: While a database session using a time set in the future is actually possible, problems arise when returning to the present. These problems should be avoided for various reasons, e.g.:

- the database will contain dates from the future, and these could result in the formation of negative values when calculating differences to the current date
- recovery activities will no longer be possible or may lead to inconsistencies (UDS/SQL, LEASY).

Incidentally, similar problems also arise, regardless of whether or not TASKDATE is used, if the IPL option is used first to put the system time forward and then to reset it at a subsequent system start!!!

The test process in detail:

Testing must include the following work steps:

- Prepare a separate test database for the test and align the records to the future time or introduce appropriate new records.
- Reset the time (using TASKDATE) and then start the database and application tasks (using the same task time).
- Carry out the actual test.
- Delete the test database once the test has been completed. This is to ensure that the test status of the database is not used once you have returned to the current time.

If these steps are properly carried out, the productive database will not be affected by the test.

Notwithstanding the security measures, testing is subject to the following restrictions:

- SESAM: Saving to tape is not possible; backup tapes produced in this way cannot be reused.
- PRISMA: User programs used for tape backups do not work. Printouts contain the system time.
- GOLEM: Testing of the execution of periodic search profiles (SDI profiles) is restricted. In particular, it is not possible to insert the date of the last execution. In addition, the execution date cannot be reset.
- ORACLE: The effects of changing the time have not yet been investigated. It is thought that it may lead to inconsistencies in the database system.
- DRIVE: No restrictions. However, DRIVE generally works on the basis of database systems such as UDS/SQL or SESAM, in which case the restrictions affecting these database systems come into effect

5.2 LMS

a) (old) LMS (started by /EXEC $LMS):

New or modified library elements of type "S", "X" and "P" get the simulated time as USER-DATE; elements of type "R" and "L" on the other hand get the system time.

Reason:
The attribute USER-DATE of any element from PLAM libraries gets the system time if the USER-DATE is set in TPR; it gets the simulated time if USER-DATE is set in TU.

b) (new) LMS with SDF-functions (started via /LMS or /START-LMS):

The USER-DATE can be set explicitly when adding or modifying the library elements. If no USER-DATE is set by the user, the element -independent from the element type- gets the simulated time as USER-DATE. CRE-DATE and MOD-DATE contain always the system time.

5.3 MAREN

a) explicit reservation of tapes

The simulated date is set as RESERVATION-DATE in the MAREN catalogue if the reserving task is working with a simulated time. As well the simulated date is used as base for calculating the FREE-DATE if a time limit in days is set (instead of an explicit date).

All other dates in the MAREN catalogue (CREATION-DATE resp. -TIME, LAST-ACCESS-DATE resp. -TIME, EXPIRATION-DATE and LAST-CLOSE-DATE resp. -TIME) contain the system time, because all these date values were ascertained in TPR

b) implicit reservation of tapes (via SCRATCH request)

The RESERVATION-DATE in the MAREN catalogue contains the simulated date for tape reservations of any task, if the MARENUCP batchjob is running with a simulated time. The FREE-DATE also gets a simulated time, if a time limit of days was fixed (instead of an explicit date) by the global MARENADM parameters.

c) exporting of tapes

The simulated time also appears as EXPORT-DATE in the MAREN catalogue.

This aspects have to be considered for tests of MAREN user programs.

5.4 ADABAS

ADABAS uses internally the STCK instruction in order to get the actual time. Therefore, TASKDATE can only be used for testing ADABAS user programs.

5.5 Privileges

It is possible to define privileges for the commands /SET-TASK-CLOCK and /SHOW-TASK-CLOCK in order to restrict the usage to some specific user-ID's for testing.

So named "customer privileges" can be defined - if SDF-A and SECOS V2.1 are available: The syntax file with the 2 commands has to be linked then (via SDF-A) to the customer privilege (defined via SRPM).

5.6 Relation of time values

5.6.1 Local time - UTC

Local time:
The local time is the time which is legally applicable in the user's environment. This means, for example, that the local time contains the time changeovers between summer and winter time.

UTC:
The UTC (Universal Time Coordinated) is an internationally defined time. It corresponds to GMT (Greenwich Mean Time), it continues running monotonously, and recognizes no changes in time such as that between summer and winter time. The difference between the local time and UTC in Germany is one hour in the winter period, and two hours in the summer period.

5.6.2 System time - simulated time

System time:
Time which is used for running the operating system. The system time is defined at start-up time and normally corresponds to the same time in local time. The system time subsequently proceeds to run continuously. If the system time is manipulated for test purposes, this changes both the local time for the system and also the UTC.

Simulated time:
This is a newly introduced time with TASKDATE. The new function enables any task to set its own system time. This means that local time and UTC are altered by the same amount.