Tell commands

D2000 Tell - command syntax

D2000 Server
D2000 Archiv
D2000 Kom
D2000 Event
D2000 Calc
D2000 DBManager
D2000 Alarm
D2000 Topology
D2000 GateWay
D2000 WorkBook

Tell commands can be run:

- interactively from the D2000 System Console process
- interactively from the D2000 Application Manager process
- from the command line using the D2000 Tell process
- from an ESL script using the COMMAND action

D2000 SERVER (KERNEL)

AUDIT_INFO [REINIT [USER|RES GROUP]]	The command without parameters writes out the status of monitoring or storing of auditing information (see also Object Group - monitoring changes in configuration). Parameter REINIT USERS causes the writing of the current configuration of all objects of *User* type into the monitoring database. Parameter REINIT RES_GROUP causes the writing of the current configuration of necessary objects of *Object group* type into the monitoring database. In any object type is not stated in the command, both types of objects will be written into the monitoring database.
BACKUP SYSCFG path	Creates a configuration database backup in a directory specified by the *path* parameter (Sybase SQL Anywhere and PostgreSQL platforms). **Note:** The backup can be influenced by the setting of ConfigBackUpTimeout parameter.
BACKUP LOGFILE path	Creates a log database backup in the directory specified by the *path* parameter (Sybase SQL Anywhere and PostgreSQL platforms).
CFGSYNCHROAUTO ON/OFF	Enables/disables the automatic synchronization of configuration databases after a new SBS (standby server) is connected. **Note 1:** The command can be applied to the HOT server only. **Note 2**: The synchronization can be influenced by the setting of the parameter ConfigSynchroTimeout.
CHECK_ARCHIVE	Checks a value type consistency between the primary archive object and the archive one and shows the errors (writes it into the log). The error is if the value type of the primary archive object is different from the value type of historical value.
DEPLOY_JAVA_SHARED	If some files - the external and user libraries for Java, placed in directory *%lt;appldirgt;/java/shared* are changed, the D2000 Server will send these files to all running clients.
DI ON/OFF **DI** /E+dbginfo **DI** /E-dbginfo	Debug Info - shows (DI ON)/ hides (DI OFF) internal debug information of D2000 Server (in the process window and in the process D2000 Sysconsole). The command allows show/hide viewing debug information by category as the SysConsole user interface allows. Example 1: Show debug information DI /E+DBG.CFG_RQ Example 2: Hide debug information DI /E-DBG.CFG_RQ

<table>
<tr><td>DP ON/OFF</td><td>Debug Pipe - enables/disables the displaying of the communication of the D2000 Server with client processes (only if the D2000 Server's window is displayed on the desktop).</td></tr>
<tr><td>ENABLEAUTOSTARTPROCESSES</td><td>Since version v23.0

If the kernel was started with the /N parameter, i.e. so that it does not start processes, by calling this tell command, startup is turned on and kernel processes start.</td></tr>
<tr><td>INSTANCE SET|GET ACTIVE|PREFERRED| USABLE ...</td><td>It is used to determine and set the active instance, preferred instance, and instance mode in a load-balancing mode.
<table>
<tr><td>INSTANCE GET PREFERRED process_name</td><td>Shows the number of the preferred process instance.</td></tr>
<tr><td>INSTANCE SET PREFERRED process_name instance_number</td><td>Sets a preferred process instance.
It is used to set the preferred archive if the archives run as an instance (for more information see the chapter Redundant archiving). By default, the instance with a minimal number is active.
This parameter can be used also for setting up the D2000 KOM process (for more information see the chapter Redundancy of communication process KOM).

The command writes the number of the preferred instance into the system registers:

HKEY_LOCAL_MACHINE\Software\Ipesoft\D2000V70\cfg_Meno_Aplikacie\Kernel\SELF.ARC_PreferredActiveInstance

Warning: If the preferred value is 0, the process instances are not switched actively while the terminated active instance is started. I.e. when some active instance is finished, another existing instance will be a new active instance. If there are more process instances, the active one will be that with the minimum instance number. This process instance remains active even if another one is started (it had to be active before termination).

Incorrect termination of the active and preferred instances may change the preferred process instance. If the active instance crashes while another instance is operating, it becomes a new preferred active instance (with the minimum number).</td></tr>
<tr><td>INSTANCE GET ACTIVE process_name</td><td>Enables to display of the number of the active process instance.</td></tr>
<tr><td>INSTANCE SET ACTIVE process_name instance_number</td><td>Sets the active instance of a given process.
The parameter is used for setting up the active instance of the archive process if all the instances are running as instances (for more information see the chapter Redundant archiving). By default, the active instance is the one that connects to the D2000 Server process as first.
This parameter can be used also for setting up the D2000 KOM process (for more information see the chapter Redundancy of communication process KOM).

The command switches the active instance of the process.</td></tr>
<tr><td>INSTANCE GET USABLE process_name instance_number</td><td>Enables display '1' if a process of a given instance is active in load-balancing mode. If not, then '0'.</td></tr>
<tr><td>INSTANCE SET USABLE process_name instance_number 1|0</td><td>Sets the instance of the process as active or passive in Load balancing mode (see the chapter Redundant archiving - Load balancing).</td></tr>
</table></td></tr>
<tr><td>MESSAGE Message_Text process_name</td><td>Broadcasts (sends) a text message to client processes. The process name can be entered as a mask (e.g. *.HIP - for all processes D2000 HI). If the return code is Success, it contains the number of clients, and the message from the D2000 Server was sent.</td></tr>
<tr><td>LIC_PostponeLicWrnToUsers 0 až 240 [hod], default 24</td><td>Delay of license warnings for common users. The administrators of D2000 will get the license warning always and immediately.</td></tr>
</table>

LIC_DoNotAllowConsoleOverLimit 0|1, default 0	When setting 1, the system will not allow connecting consoles (HI) over the scope of the license.
LIC_DoNotAllowSysProcOverLimit 0|1, default 0	When setting 1, the system will not allow the system processes (CALC, EVENT, KOM,...) over the scope of the license.
RD_SET_HOT serverName	Sets the HOT server in the redundant system. The parameter *serverName* is the name of the server within the redundant group.
REFRESH_LICENCE	On-line refresh of the information about the license range (from file *LicenceRun.code* or *LicenceDev.code*) and information about the network clients (from file *ConsolesInfo.txt*). It enables changing the license range without any need to stop the system - as well as re-reading the list of network clients after it has been manually edited. **Note 1:** In redundant systems, it is necessary to replace the files *LicenceRun.code* or *LicenceDev.code* on all redundant servers. **Note 2:** In redundant systems, it is necessary to modify the file *ConsolesInfo.txt* only on the active (HOT) server. The changes will be transferred to all SBS servers. **Note 3:** The information about the license is sent to the client processes when connecting to D2000 Server. For that reason, if the change of license affects even other processes than D2000 Server, they must be restarted. For example, when a user adds a new communication protocol, it influences the D2000 CNF process. To enable this protocol in D2000 CNF, it must be restarted. Also, when adding a new communication station and I/O tags with the given protocol to the D2000 KOM process, it must be restarted.
RELOAD_PARAMS	Reloads all parameters for D2000 Server (except the parameters for saving the monitoring database to the depository) from the registry (Windows) or from the configuration file (Linux) without restarting the D2000 system.
RELOAD_SECURITY	Sets the new configuration of Security Policy without restarting the D2000 System.
REPAIR_ARCHIVE	Checks the consistency of value type between the object being archived and the historical value. If some error occurs, it will be repaired immediately and information will be written into the log. The error is if the value type of the object being archived is different from the value type of historical value.
REPOSITORY INIT [path]	Initialize the path to the repository. If the directory does not exist, a new one will be created. If there is a repository on a given path, this one will be used for storing object history. History capturing must be disabled.
REPOSITORY DISCONNECT	Remove the setup path to the repository. History capturing must be disabled.
REPOSITORY ON|OFF	Enable/disable history capturing. A path to the repository must be set.
RESTART process_name	Stops the given process and then starts it correctly. **Note:** If the process is stopped (Stop or Crash status), the command just starts it.
SET_LOAD_BAL_MODE process_name mode	Activates/deactivates Load balancing mode for D2000 Archiv (see the chapter Redundant archiving - Load balancing) or D2000 Kom (see the chapter Redundancy of communication process KOM) processes. **Example:** *SET_LOAD_BAL_MODE SELF.ARC 1* - activates Load balancing mode for SELF.ARC process. *SET_LOAD_BAL_MODE SELF.KOM 1* - activates Load balancing mode for SELF.KOM process.
SHOW_ACCESS_LIST	The command writes the list of the names registered by the GETACCESS action within the application (with the keyword **PUBLIC**) into the text console and to the log file (*kernel.log*) of the D2000 Server process. The list contains the owner of the registered name as follows: *computer_name*;*process_name*;*event_name/activepicture_name* **Example:** ACCESS LIST BEGIN GETACCESS : "RegisteredName" is locked by : ComputerName;SELF.EVH;E.EventName GETACCESS : "RegisteredName2" is locked by : ComputerName;ComputerName.HIP;S.PictureName ACCESS LIST END
SHOW_ARC_STAT process_name	Shows the statistical information about the running instances of the archive - the maximum number and sum of the read requests that are being processed.

SHOW_CONFIG	Shows configuration information of the process D2000 Server. This information covers: • start parameters of process D2000 Server, • configuration parameters of D2000 system installation common for all applications, • parameters of process D2000 Server specified in Windows registry (specific for running application), • in redundant systems parameters of redundancy specified in Windows registry (specific for running application), • parameters specific for ODBC and OCI versions of process D2000 Server. • detailed information on the number and structure of the tags **For Structured Variables:** Object Name; the number of tags; the number of tags according to the methodology into the version D2000 V11 ;D2RECORD;SV.DaE_Export_Head; 0; 11 **For Structured Eval Tags:** Object Name; the number of tags; ;CLC_VAL Struct;P.TASK_SUM_Pending; 13 **Summary information on the contribution to the total number of tags for each object types:** Number of tags for structured variables from V11 and to V11 ;D2RECORD;Summary V11+; 9126 ;D2RECORD;Summary OLD; 49044 Number of tags for structured eval tags ;CLC_VAL Struct;Summary; 314 Number of tags for eval variables ;CLC_VAL;; 98 Number of tags for I/O tags ;POINT;; 25 Number of tags for user variables ;USER_VAR;; 2095 Number of tags for remote objects ;REM_OBJ;; 2 The total number of application tags ;TOTAL TAGS;; 11510 **Note:** By comparing the information, obtained by this TELL command, it is possible to find out whether D2000 Server processes in a redundant group are configured identically, resp. if any parameter was omitted during the migration of the application server.
SHOW_DYN_INFO object_name or HOBJ	Shows dynamic information on the specified object (list of objects that dynamically or statically use the object) on the text console of the D2000 Server process. **Note:** The command is intended to be used mainly for the D2000 system diagnostic purposes.
SHOW_DYNO_LIST idFirst idLast	The command dumps the list of current dynamic objects (ID and object name) to the text console of the process based on the defined interval. **Note:** The command is mainly intended for diagnostic purposes.
SHOW_TRANS_LIST	Lists internal transactions of the D2000 system. **Note:** The command is intended to be used mainly for the D2000 System diagnostic purposes.
SHUTDOWN D2000	Stops the D2000 system.
SHUTDOWN RESTART_D2000	Restarts the D2000 system.
SHUTDOWN RESTART	Stops the D2000 system and restarts the operating system.
SHUTDOWN SHUTDOWN	Stop the D2000 system and shutdowns the operating system.
SHUTDOWN WINNT	Stops D2000 System and restarts the operating system.
START process_name	Starts the specified process.

STDOUT output	The command redirects the standard output to a new output specified by *output*. **Example:** D2000 on OpenVMS platform: *STDOUT "_TNA2:"* (output is redirected to the terminal *_TNA2:*) D2000 on Windows platform: *STDOUT C:\output.txt* (output is redirected to the file *output.txt*) **Note:** If a D2000 process is run on the Windows platform as a service, the console output is disabled.
STOP process_name [FORCE]	It stops a process. When using the optional parameter FORCE, the termination of the process will be forced.
***SUSPEND** * SLEEP [WAKEUP yyyy-mm-ddThh:mi:ssZ] |HIBERNATE	It causes the operating system to switch into sleep mode (SLEEP) or hibernation (HIBERNATE). If the OS switches into sleep mode, the user may define a time of its automatic awakening by the parameter WAKEUP yyyy-mm-ddThh:mi:ssZ, where yyyy-mm-ddThh:mi:ssZ defines UTC time of awakening. The string must be in ISO 8601 format (http://sk.wikipedia.org/wiki/ISO_8601). **Example:** *SUSPEND SLEEP WAKEUP 2015-02-11T12:00Z* - causes putting the computer to sleep and automatic awakening at noon, February 11th, 2015 (UTC).
TCP_NO_DELAY process_name|process_mask [FALSE|TRUE|QUERY]	It enables, disables, or queries the delay of data transfer between the D2000 Server process and the client connected via TCP/IP protocol. Setting the parameter TCP_NO_DELAY to TRUE disables TCP delay before sending TCP packets (setsockopt function with parameter TCP_NODELAY - see Nagle's algorithm). The delay improves the efficiency of TCP/IP networks by reducing the number of packets that need to be sent over the network. On the other hand, if the communicating partners wait for each other's response (e.g. remote procedure calls between two EVENT processes or between HI and EVENT), this delay can significantly slow down the execution of scripts and other interactions. By default, TCP delay is on (TCP_NO_DELAY = FALSE). This TELL command is intended only for tuning and debugging purposes. **Note 1:** When registering a TCP/IP client, the process D2000 Server queries the value of parameter TCP_NO_DELAY, and, depending on this value, it does or does not change the parameter TCP_NO_DELAY for client connection. This value is then sent to the client, which also changes the TCP_NODELAY parameter of its TCP connection to the server. **Warning:** In the current implementation, the TCP_NO_DELAY command does not cause any change in the setting on the client's side! **Note 2:** Besides the TCP_NO_DELAY parameter, there is also another parameter that influences sending TCP delayed acknowledgements (it can be set on OpenVMS and *nix system by calling *sysconfig -r inet tcpnodelack*). This parameter induces a 200 ms delay before sending TCP acknowledgements and it can interact with the TCP_NODELAY parameter.
XML_EXPORT_TABLE	The configuration parts exported to the XML files, which is not part of the object configuration. The parameters: [Path, FileName, TableName]. • **Path** - defines the path for saving the configuration. • **FileName** - defines the name of the output file. E.g. for the "systemtxt" file, a file with the name "systemtxt.xml" is created. • **TableName -** defines the table of the database, from which the configuration is to be exported. TableName can acquire the following values: SYS_COLORS, SYSTEM_TEXT, TRANS_MASKA, DICTIONARY, LANGUAGES, LOGDEF, NAME_RULES. **Warning**: Extension ".**xml**" is added automatically, do not explicitly specify it.
XML_IMPORT	**Warning:** This command is not supported for OpenVMS. XML file(s) import of a specific file or of all XML files from specified directory, depending on *path_to_dir* / *path_to_file* parameter: *path_to_dir* - directory name with XML files (eg D:\temp). Note - the directory is searched for xml files recursively. *path_to_file* - absolute path to the XML file (eg D:\temp\H.Sec.xml) For this TELL command to be executed, a configuration key must be present!. The path for **the file import on VMS** should be entered in **Unix format**. **Correct syntax:** /DKB0/d2000/d2000-app/RIS_ROVE/XML/ **Incorrect syntax:** DKB0:[d2000.d2000-app.RIS_ROVE.XML] - this syntax is not supported, import of directory containing XML files ends with an error The import of one file supports both records: **Correct syntax**: /DKB0/d2000/d2000-app/RIS_ROVE/XML/file.xml **Correct syntax**: DKB0:[d2000.d2000-app.RIS_ROVE.XML]file.xml **Note:** The command belongs to the "security" TELL commands. I.e., it can be activated only from interactive processes with configuration key or from process D2000 Event Handler.

XML_IMPORT_TABLE	Import of the configuration parts that are not part of the object configuration from XML files. The parameter [Path, FileName, TableName[, CleanTable]] • **Path** - defines the path from which the configuration will be read. • **FileName** - defines the name of the input file without XML extension. For example, "systemtxt" for the file with the name "systemtxt.xml". • **TableName** - defines the table of the database to which the configuration is to be imported. TableName can acquire the following values: SYS_COLORS, SYSTEM_TEXT, TRANS_MASKA, DICTIONARY, LANGUAGES, LOGDEF, NAME_RULES. • **CleanTable** - defines whether the table in the database will be cleaned before import. This parameter is **optional**, the default value is *FALSE*. The parameter can acquire two values: *TRUE* - clean table, *FALSE* - the table is updated. The initialization of the TELL command is conditioned by using the configuration key. **Note:** The command belongs to the "security" TELL commands. I.e., it can be activated only from interactive processes with configuration key or from process D2000 Event Handler.
XML_Update	Partial update of the objects by XML from the directory, which is defined by parameter *Path.* The parameter: [path_to_dir / path_to_file] **Note:** The command belongs to the "security" TELL commands. I.e., it can be activated only from interactive processes with configuration key or from process D2000 Event Handler.

D2000 ARCHIV

BACKUP [path]	Creates an archive database backup. If the parameter *path* is not specified, the backup will be performed into the directory, which is set in the archive configuration. **Note:** this command is supported for Sybase SQL Anywhere only. To back up Oracle/PostgreSQL/MsSql databases, use their respective utilities.
CALC_OLD_REQUESTS [path+filename]	Executes recalcs of old values, which were redirected to the auxiliary calc task (see the RecalcImmediateDepth archive parameter), and then instead of being performed, they were stored to the file (see the DropOldRequests archive parameter). If *path+filename* is not specified, the file *name_ARCHIV_DROPPED_CALCS.DAT* will be processed by *name.ARC* and then renamed to *name_ARCHIV_DROPPED_CALCS.DONE* (if such a file exists, it will be deleted). **Note:** This command fails if the archive parameter DropOldRequests has value 1 because in this case the recalcs would be again stored to the file after being read.
CLEANUP_SLICES	A command that must be run after the change of DataTableSlices parameter from value 1 to value 2 (i.e. after changing the time slices mode to time slices for structured archives only). The archive will transfer the archived values from time slices of simple archives to the original data tables and these time slices will be deleted. **Note:** for simple archives, from the start of the archive till the transfer of archived values to the original data tables, the archive will provide only data which are located in original data tables. Therefore this conversion is recommended only for systems with redundant archives and the archive performing conversion should be set as a passive instance.
CRC path\filename mask from to [step] [PO]	Parameter checks the identity of data in instance archives. Creating the file with a CRC (checking sum) for archive objects that correspond to the mask for the entered time period. The optional parameter *step* (in hours) allows dividing the calculation for time period into intervals. The *PO* parameter ensures the calculation will be executed only for primary archive data. **Note:** Values entering the CRC calculation can be "truncated" - in a 64-bit floating-point value complying with the IEEE 754 standard, the two lowest bytes will be zeroed, if a debug category *DBG.ARCHIV.CRC.Z2LB* is enabled. The truncation can be used to ignore minor differences (16. and higher decimal point), which can occur in floating-point calculations in archives running on different processors.
CRC_TREE path\filename name row col from to [step]	Parameter checks the identity of data in instance archives. Creating the file with CRC (checking sum) for archive objects that correspond to the mask for the entered time period. The calculation will be realized for the object that has been defined by parameter "name row col" and for all objects that influence its value. The optional parameter *step* (in hours) allows dividing the calculation for the time period into intervals. See the note to the CRC command.
DEL mask [before]	Deletes data from the archive database as follows: • the parameter *before* is not defined - for individual archive objects, there will be deleted all data that are older than the history depth (the parameter History depth) defined in the configuration of the objects, • the parameter *before* is defined - there will be deleted all data with the timestamps older than defined by the parameter *before*.

DELETE_DATA mask from [to]	Deletes data from the archive database without activation of dependent statistics evaluation (similar to the action DELETEARCHDATA). Time format is *dd-mm-rrrr hh:mi:ss.mss*.
DI ON/OFF	Debug Info - shows/hides internal debug information of the D2000 Archiv process (in the process window and in the D2000 Sysconsole process).
DI ON/OFF [HOBJ/mask [ROW [COL]]]	Debug Info for values - enables/disables displaying information about the evaluation of historical values with specified HOBJ or with the name matching the given mask. For structured historical values, you can specify the required structure item - column (if ROW=0) or row (if COL=0). **Example:** *DP ON H.Test1 3 4* - enables debug information for the item of structured historical value *H.Test1* on row 3 and column 4 *DP ON H.Test2* - enables debug information for historical value *H.Test2* (for all items if the object is a structure) *DP OFF 1234 0 4* - disables debug information for the whole column 4 in the structured historical value with HOBJ=1234 By default, debug information shows just the time and the value of historical value. Enabling the debug category *DBG.ARCHIV.DATA* (in the process D2000 System Console or using the start parameter /E+DBG.ARCHIV.DATA when starting the process D2000 Archiv), allows to show the complete stored data designed for the D2000 System developers: *Time* (archive value time) *Value* (value) *Status* (a True/False array containing Valid, ProcAlarm, NoAckPAlarm, PrAlSilent, Weak, NoAckValue, Transient, Default, Manual, AlCrit, Unknown values) *Limits* (one of the value of InLimit, VL_Limit, L_Limit, H_Limit, VH_Limit, LimitsProblem) *ArchFlags* (a True/False array containing Start, Stop, Block, UnBlock, Deleted, UserModify, OldVal, ProcesModify, LoadData, MONO_TIME, K, L, M, N, O, Ver1 values)
DI ONREC /OFFREC HOBJ/mask [ROW [COL]]	Debug Info for values - recursive. The command extends the functionality of the previous commands with the possibility of showing / hiding the debug information for historical values that depend on specified historical values (e.g. minute averages).
DISMOUNT_TREZOR path+trezor_name	Sybase platform: The parameter *path+trezor_name* specifies the path and the name of the depository database to dismount. The * and ? characters for mask definition can be used in the depository database name. If more files match to entered mask, all these ones will be dismounted. For more information see the topic On-line access to data from depository databases.
DISMOUNT_TREZOR Id [SEGMENT seg] **DISMOUNT_TREZOR** trezor_name [SEGMENT seg] **DISMOUNT_TREZOR** IdFrom IdTo [SEGMENT seg]	Oracle platform: The parameter *Id* or *trezor_name* defines the number (name) of the depository database to dismount. The name of the depository database is the name of the depository tablespace (e.g. MYAPP_TS_TREZOR15 or MYAPP_TS_TREZOR15_02) or the name of the tablespace's datafile (e.g. MYAPP_TS_TREZOR0015_S02_20040801.ORA). You can also dismount all depository databases (except the ones that are currently being filled) by using the parameter *all*. The parameters *IdFrom* and *IdTo* allow dismounting a sequence of depository databases. The parameter *SEGMENT seg* (where *seg* is the number of depository database segment) allows dismounting just the segment. *Seg* is the number between 0 and TrezorCountSegments. If the **DISMOUNT_TREZOR** command is used without the *SEGMENT Seg* parameter for depository database with segments enabled, the process D2000 Archiv attempts to dismount all segments of specified depository database/s. For more information see the topic On-line access to data from depository databases.
DISMOUNT_TREZOR Id [SEGMENT seg] **DISMOUNT_TREZOR** IdFrom IdTo [SEGMENT seg]	PostgreSQL platform: The parameter *Id* defines the number of depository databases to dismount. You can also dismount all depository databases (except the ones that are currently being filled) by using the parameter *all*. The parameters *IdFrom* and *IdTo* allow dismounting a sequence of depository databases. The parameter *SEGMENT seg* (where *seg* is the number of depository database segment) allows dismounting just the segment. *Seg* is the number between 0 and TrezorCountSegments. If the command **DISMOUNT_TREZOR** is used without the parameter *SEGMENT Seg* for depository database segments enabled, the process D2000 Archiv attempts to dismount all segments of specified depository database/s. For more information see the topic On-line access to data from depository databases.
DP ON/OFF	Debug Pipe - shows/hides the course of communication with D2000 Server (only if the process window is displayed on the desktop).
DROP_LIST path+filename	Creates the file (defined by the parameter *path+filename*) containing SQL commands to delete tables, which are not used by the archive configuration. If the time slices are enabled, the file will contain also the commands to delete respective rows of the table ARC_SLICE, which contain information about the time slices corresponding to tables that are to be deleted.
FILL_CACHE	Manual loading of archive cache (only available for isochronous cache mode) by reading values from the archive database. Archive cache loading can be also automatic after startup - for more information see the description of parameter IsoCacheAutoFill.
FORCE_CLEAR HOBJ /mask	For the selected object(s), their ClearTime is set to a value of several hours in the past, which means that a periodic delete of object's data should occur shortly - within a minute (unless it is disabled for some reason, e.g. if the time slices are enabled, it can depend on parameters DeleteInSlice0 and DeleteInSlices).

FREEZE freeze_seconds [report_seconds] **UNFREEZE**	FREEZE command causes the writing and calculating tasks of the archive stop processing the requests for a period *freeze_seconds*. Within this time, another task (deleting) will list the number of requests (primary and calculated) in the queues. After the elapsing of the *freeze_seconds time,* the processing of values will be restored. If the *report_seconds* parameter is entered, the deleting task continues to list the number of requests in the queues for this period. FREEZE command is primarily used for performance tests (testing the maximum speed for processing the requests under load). Before its use, you should realize that it does not influence reading from the archive, but during its execution, the writing to the archive (so the newest values are not available when reading) and the calculations are not executed for the *freeze_seconds* period. In redundant systems with 2 and more archives, we recommend executing the FREEZE command on a passive instance, so that the functionality of the archiving subsystem will not be affected. UNFREEZE command (without parameters) is used for aborting inactivity that was caused by the FREEZE command.
LIST_TREZOR [all] **LIST_TREZOR** id	The parameter allows showing the list of depository databases used by the process D2000 Archiv. The contents of the list depend on the used platform - Sybase, PostgreSQL, or Oracle. Parameter *id*: • for Sybase - name of a depository database • for Oracle - number of depository tablespace • for PostgreSQL - number of a depository database
LOAD_TREZOR path+filename [mask] [from] [to]	On-line depository database import (Sybase only)
MOUNT_TREZOR path+trezor_name	Sybase platform: The *path+depository_name* parameter defines the path and name of the depository database to be mounted. The name may contain "*" and "?" characters to define a mask. If more files match the specified mask, all these will be mounted. For more information see the topic On-line access to data from depository databases.
MOUNT_TREZOR Id *[SEGMENT seg] [WRITE]* **MOUNT_TREZOR** trezor_name *[SEGMENT seg] [WRITE]* **MOUNT_TREZOR** IdFrom IdTo *[SEGMENT seg] [WRITE]*	Oracle platform: The parameter *Id* (*trezor_name*) defines the number (name) of a depository database to be mounted. The name of the depository database is the name of depository tablespace (e.g. MYAPP_TS_TREZOR15 or MYAPP_TS_TREZOR15_02) or the name of tablespace's datafile (e.g. MYAPP_TS_TREZOR0015_S02_20040801.ORA). The *IdFrom* and *IdTo* parameters allow mounting a sequence of depository databases. The *SEGMENT Seg* parameter (where *Seg* is the number of the depository database segment) allows mounting a single depository database segment. If the **MOUNT_TREZOR** command is used without the *SEGMENT Seg* parameter for depository database with segments enabled, the process D2000 Archiv attempts to mount on all segments of specified depository database/s. The parameter *Write* can be used to mount the depository database (s) for writing (for depository database synchronization through the Arcsynchro tool). For more information see the topic On-line access to data from depository databases.
MOUNT_TREZOR Id *[SEGMENT seg] [WRITE]* **MOUNT_TREZOR** IdFrom IdTo *[SEGMENT seg] [WRITE]*	PostgreSQL platform: The parameter *Id* specifies the number of a depository database to be mounted. The parameter *SEGMENT Seg* (where *Seg* is the number of the depository database segment) allows mounting a single depository database segment. If the **MOUNT_TREZOR** command is used without the *SEGMENT Seg* parameter for depository database with segments enabled, the process D2000 Archiv attempts to mount on all segments of specified depository database/s. The parameter *Write* can be used to mount the depository database (s) for writing (for depository database synchronization through the Arcsynchro tool). For more information see the topic On-line access to data from depository databases.
OPTIMIZE_QUEUE	Reduction of the number of calc requests in the archive queue. If for some reason, the number of requests in the archive queue has increased (for example, due to the arrival of old values from the communication), this TELL command can reduce duplicate requests.
PENDING_REQUESTS path+filename	Creates a file containing the descriptions of pending requests.
RECALC Mask [Row [Col]] From [To] [BACKGROUND]	Recalculates the values in the statistical archive. Mask - defines the list of objects, which are to be recalculated. If the specific row and/or column of the structured archive is recalculated, it is possible to specify parameters *Row* and *Col* (if unspecified, their default value is 0, meaning "whole row/column"). The parameters *<From, To>* define a time interval to be recalculated. If the end time is not entered, the current time will be used. Time format is *dd-mm-yyyy [hh[:mi[:ss[.mss]]]]*. Optional parameter *BACKGROUND* puts the recalculation to the auxiliary calc task (see the archive parameter RecalcImmediateDepth). The parameter is recommended for long-lasting recalculation of historical values that are supposed to be performed in the background while the archive still continues archiving real-time data. **Note:** If the parameter RecalcTimeIntervalLimit has a non-zero value and the time interval <From, To> is longer than RecalcTimeIntervalLimit hours, the parameter *BACKGROUND* is mandatory, to ensure that the recalculation will be performed by auxiliary calc. This behavior is implemented as a protection against overloading the main task by recalculations.
RELEASE_TREZOR	Premature disconnection of the current depository database.

<table>
<tr><td>RELOAD_VALUES mask SCRIPT|PRIMARY|STATISTIC|EVAL|ALL</td><td>The command is used to force the reading of the last values of the specified objects from the archive database. Objects can be specified by name/mask and by type: SCRIPT - script PRIMARY - primary archive objects STATISTIC - statistical archive objects EVAL - calculated archive objects ALL - all types of archive objects If e.g. in the test environment the inserting of values of script-filled archives is not possible, but these are inserted into the archive database by the arcsynchro utility, it is possible to execute this command after the insertion is completed (e.g. RELOAD_VALUES * STATISTIC). Subsequently, a read (e.g. of all script-filled archive objects) is performed and recalculations of archive objects that depend on the read objects can be initiated.</td></tr>
<tr><td>REM_EMPTY_SLICES HOBJ|mask</td><td>If the time slices are activated, the command causes the archive to review all time slices of simple archive objects according to the defined HOBJ or mask and to delete the time slices without data.
The command is used to clean up the archive database from empty time slices that were created due to the error of the archive. This error was repaired in the supported versions on June 8, 2015.</td></tr>
<tr><td>REORGANIZE</td><td>On-line reorganization of archive database. The command compacts all archive tables (same as the Oracle command ALTER TABLE MOVE or the Sybase command REORGANIZE TABLE INDEX or the PostgreSQL command VACUUM - see the PG_ReorganizeSlice parameter).

Note: For Oracle, PostgreSQL, and Sybase 8.0 (and above) platforms only.</td></tr>
<tr><td>REORGANIZE HOBJ|mask [SLICE slice] [tablespace]</td><td>On-line reorganization of archive table of an object with specified HOBJ or with a name matching the specified mask. If the parameter tablespace is also defined, the given archive table will be moved to the specified tablespace. If time slices are on, all slices of one archive object will be reorganized/moved unless a slice number slice is specified.

Note 1: The parameter tablespace can be used on Oracle and PostgreSQL platforms only.
Note 2: The slice number, except for ordinary values, supports special values:
value -1 means "all slices"
value -2 means "slice for current time"
value -3 means "previous time slice"
value -4 means "all slices older than current time slice"
value -5 means "only those archive objects that do not have time slices" (if time slices are configured only for structured archives, then only simple ones will be reorganized)</td></tr>
<tr><td>REORGANIZE ON</td><td>Enables the reorganization of the archive database (same as running the process D2000 Archiv with the /DBCY parameter).</td></tr>
<tr><td>REORGANIZE OFF</td><td>Enables the reorganization of the archive database (same as running the process D2000 Archiv without the /DBCY parameter).</td></tr>
<tr><td>REORGANIZE ACTIVE</td><td>Enables the reorganization of the archive database in the active mode only (same as running the process D2000 Archiv with the /DBCA parameter).</td></tr>
<tr><td>REORGANIZE PASSIVE</td><td>Enables the reorganization of the archive database in the passive mode only (same as running the process D2000 Archiv with the /DBCP parameter).</td></tr>
<tr><td>REORGANIZE SHRINK</td><td>Enables the alternate mode of the reorganization of the archive database - only for Oracle 10g and above (dame as running the process D2000 Archiv with the /DBCS parameter).
Note: To enable the original mode, use the REORGANIZE MOVE command.</td></tr>
<tr><td>REORGANIZE TableRowLimit rowLimit</td><td>On-line change of the parameter ReorganizeTableRowLimit is used for the automatic reorganization of the archive database.</td></tr>
<tr><td>REORGANIZE TableTimeLimit timeLimit</td><td>On-line change of the parameter ReorganizeTableTimeLimit is used for the automatic reorganization of the archive database.</td></tr>
<tr><td>SET_CACHE size [MB]</td><td>The command allows the user to change the dynamic cache size to speed up the calculation of statistical historical values while the process D2000 Archiv is still running.

Note 1: The initial cache size can be specified by the parameter MaxCacheSize in the process D2000 Application Manager.
Note 2: The command SET_CACHE also changes the value of the parameter MaxCacheSize in the registry.</td></tr>
<tr><td>SET_OPTION

name_parameter value_parameter</td><td>The SET_OPTION command controls the following configuration and start-up parameters of the process D2000 Archiv:<table><tr><td>name_parameter</td><td>value_parameter</td><td>description</td></tr></table></td></tr>
</table>

AlmostZero ALMOST_ZERO	positive real number (e. g. 1e-10)	Setting the tolerance for comparing two numbers in the archive (default is 1e-10). See the description of the parameter AlmostZero.
ChainedReadPrio CHAINED_READ_PRIO	string	Set the priorities of the statistical functions used in chained reading from the archive. See the description of the ChainedReadPrio parameter.
CommitCountActive COMMIT_COUNT_ACTIVE	positive number	The number of values inserted into the archive database by an active **D2000 Archiv** after which a COMMIT is triggered (default is 1000). See the description of the parameter CommitCountActive.
CommitCountPassive COMMIT_COUNT_PASSIVE	positive number	The number of values inserted into the archive database by a passive **D2000 Archiv** after which a COMMIT is triggered (default is 1000). See the description of the parameter CommitCountPassive.
CommitTimeActive COMMIT_TIME_ACTIVE	positive number	Interval after which a periodic COMMIT is triggered by an active **D2000 Archiv** process (default is 60 s). See the description of the parameter CommitTimeActive.
CommitTimePassive COMMIT_TIME_PASSIVE	positive number	Interval after which a periodic COMMIT is triggered by a passive **D2000 Archiv** process (default is 60 s). See the description of the parameter CommitTimePassive.
DBPO	ON/OFF	Only primary data processing. Calculations of statistics and evaluated archives will not be performed.
RX	ON/OFF	No delete. Periodic deletes (for removal of data beyond configured history depth) will not be performed.
RO	ON/OFF	Read-only.
RS	ON/OFF>	Backup on the Sybase platform will be performed by copying the database file.
READ_ARCHIVE_DEPTH	"dd-mm-yyyy hh:mi:ss" OFF	Limitation of the reading depth from the archive. If the depositories are ON, the data older than the specified date will be read from the depository databases. This parameter can be used for checking if the depositories truly contain all data (e.g. after the patching by arcsynchro). The value OFF deactivates the limitation. When the limitation is active, every reading from the archive (in which the limitation is applied) produces the following note in the archive log file: *Start time for reading values from archive is before READ_ARCHIVE_DEPTH, limiting*
DiskUsageWarning DISK_USAGE_WARNING	ON/OFF	Calling with parameter OFF disables the archive warning about low disk free space. See the description of the parameter DiskUsageWarning.
DropOldRequests DROP_OLD_REQUESTS	ON/OFF	Requests for recalc of data older than RecalcImmediateDepth will be discarded. See the description of the parameter DropOldRequests.
IsochronousCache ISOCHRONOUS_CACHE	ON/OFF	Activation of the IsochronousCache. See the description of the parameter IsochronousCache.
IsoCacheFullDepth ISOCACHE_FULL_DEPTH	ON/OFF	Setting the depth of the IsochronousCache for objects with a period greater than the current cache depth. See the description of the parameter IsoCacheFullDepth.
IsoCacheDepth ISOCACHE_DEPTH	a positive number (seconds)	Manual setting of the depth of the IsochronousCache. See the description of the parameter IsoCacheDepth.
OldValIgnoreAge	a positive number (hours)	Values older than a specified number of hours will be discarded. See the description of the parameter OldValIgnoreAge.

OldValOnAuxTask OLD_VAL_ON_AUX_TASK	ON/OFF	Old values from the communication (having the OldVal flag set) will be processed by the auxiliary calc task. See the description of the parameter OldValOnAuxTask.
OneThreadForGraph ONE_THREAD_FOR_GRAPH	ON/OFF	Read requests for all objects in a single Graph will be serialized on one read task. See the description of the parameter OneThreadForGraph.
PG_ReorganizeSlice PG_REORGANIZE_SLICE	0-2	The parameter determines how time slices on the PostgreSQL platform will be reorganized (VACUUM/VACUUM FULL/CLUSTER). See the description of the parameter PG_ReorganizeSlice.
PG_ReorgSliceTime PG_REORGSLICE_TIME	seconds	The parameter defines a time interval after which previous time slices will be reorganized on the PostgreSQL platform. See the description of the parameter PG_ReorgSliceTime.
PG_TrezorFileMulti PG_TREZORFILEMULTI	number	The parameter defines a multiplicator for the CommitCount parameter if also the PG_TrezorFilePath parameter is specified. See the description of the parameter PG_TrezorFileMulti.
RecalcImmediateDepth RECALC_IMMEDIATE_DEPTH	seconds /OFF	Time depth of the recalcs of computed historical values, which are performed by main archive tasks. See the description of the parameter RecalcImmediateDepth.
RecalcParallelInterval RECALC_PARALLEL_INTERVAL	seconds /OFF	The parameter defines a minimum size of recalculated time interval intended for parallelization. See the description of the parameter RecalcParallelInterval.
RecalcTimeIntervalLimit RECALC_TIME_INTERVAL_LIMIT	hours/OFF	The parameter defines a limit for the time interval for recalculation of statistics and evaluated archives. See the description of the parameter RecalcTimeIntervalLimit.
RecalcUseTrezor RECALC_USE_TREZOR	ON/OFF	The parameter defines whether the archive will read also data from the depository databases during recalculations. See the description of the parameter RecalcUseTrezor.
ReadTimeBeforeStart READ_TIME_BEFORE_START	seconds /OFF	The parameter shifts the start time of reading from the archive by the specified number of seconds so that the last value before the start of the specified interval can be found within one reading. See the description of the parameter ReadTimeBeforeStart.
ReorganizeOffset REORGANIZE_OFFSET	hh:mi	An offset of the periodic reorganization starts in the given period. See the description of the parameter ReorganizeOffset.
ReorganizePeriode REORGANIZE_PERIODE	hours	Periodic reorganization period is given in hours. See the description of the parameter ReorganizePeriode.
ReorganizeTableRowLimit REORGANIZE_TABLE_ROW_LIMIT	rows	The parameter defines the number of deleted rows after which an automatic reorganization of an archive table is performed. See the description of the parameter ReorganizeTableRowLimit.
ReorganizeTableTimeLimit REORGANIZE_TABLE_TIME_LIMIT	hours	The parameter defines the minimum time interval between two automatic reorganizations of an archive table. See the description of the parameter ReorganizeTableTimeLimit.

ReportLongRecalc REPORT_LONG_RECALC	seconds /OFF	The parameter activates the logging of recalcs of intervals greater than the specified number of seconds. See the description of the parameter ReportLongRecalc.
SystemLogging SYSTEM_LOGGING	ON/OFF	The parameter activates the logging of information on completed recalcs in the log database. See the description of the parameter SystemLogging.
Tablespace_Name TABLESPACE_NAME	name_of_tablespace	The parameter defines the name of the archive tablespace (only on the Oracle platform). See the description of the parameter Tablespace_Name.
TrezorCompress TREZOR_COMPRESS	ON/OFF	The parameter activates depository data compression. See the description of the parameter TrezorCompress.
TrezorCompressKeep TREZOR_COMPRESS_KEEP	ON/OFF	The parameter is used to test depository data compression. See the description of the parameter TrezorCompressKeep.
TrezorCompressReorg TREZOR_COMPRESS_REORG	ON/OFF	The parameter activates the reorganization of the depository data before depository data compression See the description of the parameter TrezorCompressReorg.
TrezorCompressCmt TREZOR_COMPRESS_CMT	positive integer number	The parameter specifies the number of archive objects after the compression of which a COMMIT will be executed. See the description of the parameter TrezorCompressCmt.
TrezorCompressOrder TREZOR_COMPRESS_ORDER	ON/OFF	The parameter indicates whether time sorting is required when reading data from a compressed depository. See the description of the parameter TrezorCompressOrder.
TrezorReadSegment0 TREZOR_READ_SEGMENT0	ON/OFF	The parameter causes D2000 Archiv to read also from depository segment 0 when reading from any other depository segment. See the description of the parameter TrezorReadSegment0.
TrezorReadSinceCreate TREZOR_READ_SINCE_CREATE	ON/OFF	The parameter causes the depositories that are older than Create Time of historical value not to be read. See the description of the parameter TrezorReadSinceCreate.
WATCH_OLD_VALS	"dd-mm-yyyy hh-mi-ss" OFF	Activation of monitoring of primary values that are older than the specified parameter. The following are monitored: • object values • values that come as old values (OLDVAL) • values entered or changed in HI • values inserted through INSERTARCHARR and UPDATEARCHVAL actions Monitoring is deactivated by the OFF parameter.
WorkingHoursStart WORKING_HOURS_START	non-negative integer number	The parameter defines an hour marking the beginning of working hours, during which time-consuming archive operations are minimized. See the description of the parameter WorkingHoursStart.
WorkingHoursEnd WORKING_HOURS_END	non-negative integer number	The parameter defines an hour marking the end of working hours, during which time-consuming archive operations are minimized. See the description of the parameter WorkingHoursEnd.

Note: For parameters that are stored in the Registry database the SET_OPTION command will change the value of configuration parameters not only in the archive but also in the Registry database.

SHOW_DYN_INFO HOBJ /name [ROW [COL]]	Shows dynamic information on the specified object (current value, last sent value, time of next calculation for periodically archived objects, time of periodic deleting, parameters, and content of cache) on the text console of the D2000 Archiv process. **Note:** The command is mainly intended for D2000 system diagnostic purposes.
SHOW_CACHE count	If the archive cache is enabled, the result is a list of top *count* historical values having the most values in the archive cache. For every value, the following properties are displayed: HOBJ, name, number of items, and time interval of values in the archive cache.
SHOW_CONFIG	Shows configuration information of the process D2000 Archiv and archive database. This information covers: • start parameters of the process specified in the configuration of the archive process (if the archive process is run by process D2000 Server), or archive service (if the archive process is run by its own watchdog), or command-line parameters (if the archive process is run manually), • parameters for archive specified in Windows registry, • parameters stored directly in the archive database in the LOG_DATA table (see the parameter /CLD of arcsynchro utility), • specific parameters of the database (currently only for the Oracle platform). **Note:** By comparing information, obtained by this TELL command, it is possible to find out whether the redundant archives are configured identically or if any parameter was omitted during the migration of archive, database, etc.
SHOW_INFO	Shows various information about the D2000 Archiv process and the archive database.
STATISTICS path+filename hours	Creates the file *path+filename* with statistical information, i.e. number of values for individual historical values from the last n-hours. The *path+filename* file will be created by the D2000 Archiv process, so its naming is OS-dependent (e.g. D:\stat.txt on Windows, /tmp /stat.txt on Linux).
STDOUT output	The command redirects the standard output to a new output. **Example:** D2000 on OpenVMS platform: *STDOUT "_TNA2:"* (output is redirected to the terminal *_TNA2:*) D2000 on Windows platform: *STDOUT C:\output.txt* (output is redirected to the file *output.txt*) **Note:** If a D2000 process is running on the Windows platform as a service, the console output is disabled. Therefore if this functionality is required, it is necessary for a D2000 process to be started by D2000 Server without /X parameter.
STOP_IMPORT_DATA	Deactivates a feature of the process D2000 - import of external data.
STOP [NOQUEUE]	Stops the process D2000 Archiv. All unsaved values from the request queue are to be automatically stored on the disk and during the next start of the process, the values will be read from the disk. The current values of historical values are to be stored in the table *SAVED_LAST_ROWS* to quicken the next start of the process. If the **NOQUEUE** parameter is used, the unsaved values from the request queue will not be stored in the file (it speeds up stopping the process with a large number of unprocessed values).
TIMESTAT START **TIMESTAT** RESTART **TIMESTAT** SHOW **TIMESTAT** SHOWLONG **TIMESTAT** SHOWSTART **TIMESTAT** STOP	Starts (START), stops (STOP), restarts (RESTART), shows (SHOW), shows and restarts (SHOWSTART) or shows the time statistics of reading task(s) in long time format, including the days (SHOWLONG). After starting the statistics, the database reading task(s) will start the measuring time, spent in various parts of the reading procedure. These values can be used for tuning by system specialists. Time statistics are displayed per reading task and if ReadThreadsCount > 1, also totals for all read tasks are shown. **Note:** When time statistics are on, reading from the archive may be several per cent slower due to time measuring.
TREZOR COMPRESS Id [SEGMENT seg]	PostgreSQL platform: The command is used to compress the data of existing depository databases. See the Depository Data Compression chapter for more information. The *Id* parameter specifies the number of a depository database whose data is to be compressed. The *SEGMENT seg* parameter, where *seg* is the number of the depository database segment (between 0 and TrezorCountSegments), allows you to compress a specific depository segment if depository segment creation is enabled. Note: The compression of a depository is possible if it is mounted for writing (see MOUNT_TREZOR command with WRITE parameter).
TREZOR DEMPRESS Id [SEGMENT *seg*]	PostgreSQL platform: The command is used to decompress the data of existing depository databases. See the Depository Data Compression chapter for more information. The *Id* parameter specifies the number of a depository database whose data is to be decompressed. The *SEGMENT seg* parameter, where *seg* is the number of the depository database segment (between 0 and TrezorCountSegments), allows you to compress a specific depository segment if depository segment creation is enabled. Note: The decompression of a depository is possible if it is mounted for writing (see MOUNT_TREZOR command with WRITE parameter).

DI ON/OFF HOBJ/mask	Debug Info for values - enables/disables displaying information about the evaluation of: • I/O tag with specified HOBJ, • all I/O tags on the station with specified HOBJ, • I/O tags with the name matching the given mask, • all I/O tags on the stations with the names matching the given mask.
DP ON/OFF	Debug Pipe - shows/hides the course of communication with the D2000 KOM (only if the process window is displayed on the desktop).
GETKOMARCDEPTH	Gets the timestamp of the oldest data stored in the KOM Archive.
GETOLDVAL StationName [M:PointMask] [NORECALC] "BeginTime" ["EndTime"]	Reads historical values from a device - *StationName* station. The values from the specified interval will be read. The parameter *EndTime* is optional and if is not entered, the current time will be used. Time format ("BeginTime", "EndTime") is "dd-mm-yyyy hh:mi:ss". This feature is supported only by some communication protocols (Datalogger ESC8800, ESC8816, UNIP_TS ...). **Warning:** This is not a function of KOM Archiv but it is the reading values directly from a device! NORECALC parameter causes the archive to not execute the calculation of calculated archive objects, the values of which are calculated from the values acquired by this calling of a TELL command. For some protocols (OPC DA, OPC HDA, and IEC 870-5-104), a parameter mask "M:" can be used, which is used to read the archive values only for selected I/O tags which match the mask, from the *StationName* station.
LNALTERPATH LineName ON|OFF|PRIMARY|BACKUP|QUERY [1|2|3|4]	The command allows switching between the primary (OFF or PRIMARY) and backup (ON or BACKUP) devices for lines of categories "SerialOverUDP Device Redundant", "SerialOverUDP Line Redundant", and "SerialOverUDP System&Line Redundant", or discovering the active device (QUERY). The active device is used to write UDP packets and packets received from it are processed. Data received from the inactive device is ignored. Similar functionality is also supported for lines of the "TCP/IP-TCP System Redundant" category - the TCP connection is connected to a primary or backup device. The third parameter (numbers 1-4) specifies the subline. For a line of category "SerialOverUDP Device Redundant", it need not be specified or must be equal to 1. For a line of category "SerialOverUDP Line Redundant", it can be equal to 1 (the device on the Primary line is set) or 2 (the device on the Secondary line is set). For a "SerialOverUDP System&Line Redundant" or "TCP/IP-TCP System Redundant" category line, it can be equal to 1 - 4 (the device on the Primary/Secondary line of the system A/B is set). If an active device is detected (QUERY) and the third parameter is not entered, all active devices are listed (1, 2, or 4 depending on the line type). The command is also applicable on the OPC UA Client line in the OPC Unified Architecture Data Access Client protocol for Client Type =Default. Using the command, it is possible to find out which of several specified hosts the D2000 KOM process is connected to (QUERY) or to force a connection to a specific one (ON, third parameter set to 1 .. N). The command is also applicable on RFC2217 Client, TCP/IP-TCP and TCP/IP-TCP Redundant lines. With the help of the command, it is possible to find out which of several specified hosts the D2000 KOM process is connected to (QUERY, or QUERY PRIMARY /BACKUP) or to force a connection to a specific one (PRIMARY/BACKUP, third parameter set to 1 .. N). Examples of use: *LNALTERPATH L.MyLine ON 3* *LNALTERPATH L.MyLine BACKUP 3* *LNALTERPATH L.MyLine QUERY*
LNSTAT OPEN/CLOSE LineName	The command can be used only for the lines of Serial (asynchronous serial lines) and RFC2217 Client categories. The command stops the communication (CLOSE) and disconnects the communication port. Then the port can be used by other programs, e.g. diagnostic or configuration tools. Stations on the line are to be in the Communication error state. The communication will be recovered either using the command LNSTAT OPEN or after restarting the whole communication process.
LNTRACE DEC/HEX	Enables the decimal/hexadecimal format of line listening. The default format is hexadecimal.
READKOMARC ["BeginTime"] ["EndTime"]	Reads the values from KOM Archiv. There are transferred all old values from the specified interval. The parameter *BeginTime* is optional and if it is not entered, the data from all history depth of KOM Archiv will be read there. The parameter *EndTime* is optional and if it is not entered, the current time will be used. If the parameters *BeginTime* and *EndTime* are not defined, there will be sent just the values, which have not been sent. If at least the parameter *BeginTime* is defined, there will be sent all values within the specified interval. Return codes: • **SUCCESS** - transaction successful, data transfer to the system finished, • **IGNORED** - the process D2000 KOM is not running in the mode of KOM Archiv, • **ERROR** - invalid parameters (time format: *dd-mm-yyyy hh:mi:ss*), wrong sequence of entered times - BeginTime > EndTime.

RELOAD StationName | StationHOBJ | LineName | LineHOBJ	The command de-configures the object (line or station) and its descendants (stations, I/O tags) and reloads them. The command can be used to apply configuration changes that would otherwise require restarting the KOM process. The command is currently implemented for protocols: • IEC 60870-6 ICCP/TASE.2 • IEC 870-5-101 (client, server, balanced mode) • IEC 870-5-104, IEC 870-5-104 Server • Modbus Client, Modbus Server • OPC UA Client
RESETPERF LineName /StationName	Resets the statistics of given line / station - the system structured variable *SV._System_LinePerformance* / *SV._System_LinePerformance*. It is executed immediately but it will be visible after the process D2000 KOM sends new statistics to the process D2000 Server (periodically every 10 seconds).
SETPTADDR name "address1" "address2"	Dynamic change of the I/O tag address. The parameter *name* specifies the I/O tag. It can be entered as the object name (e.g. "M. CoolantTemperature") or the object HOBJ (the syntax requires the character '$' before HOBJ, e.g. "$278"). The parameters *address1* and *address2* define the new I/O tag address. Change of address will be performed only in internal data structures of the KOM process. It is not sent, in a centralized way, to the server, i.e. changed address is valid until the restart of the KOM process. After this, the value from the initial configuration will be used again. Return codes: • **SUCCESS** - changed address (the value of the I/O tag is invalid until it is read from the new address). • **IGNORED** - D2000 KOM process does not support dynamic address change for the specified protocol. • **ERROR** 1. duplicate address 2. point not found 3. bad address format Dynamic change of the I/O tag address is implemented only for a limited group of communication protocols: • AMiT ATOUCH32 DB-Net • BACnet • General Electric SRTP protocol • Honeywell C-Bus • L&G TOCCATA • L&G TOCCATA via UNIP2 • MODBUS Client • OPC Data Access 2.05 & 3.0 Client • Siemens SAPHIR • SNMP
SETSTADDR StationName| StationHOBJ "address"	Dynamic change of station address. The first parameter specifies the station. It can be entered as the object name *StationName* (e.g. "B.Station") or station ID *StationHOBJ*. The *"address"* parameter defines its new address. Change of station address will be performed only in internal data structures of the KOM process. It is not sent, in a centralized way, to the server, i.e. changed address is valid until the restart of the KOM process. After this, the value from the initial configuration will be used again. Return codes: • **SUCCESS** - changed address, • **IGNORED** - D2000 KOM does not support a dynamic change of address for the specified protocol, • **ERROR** 1. object not found 2. invalid number of parameters Dynamic change of address is implemented only for a limited group of communication protocols: • ALYA Lubrikacie • MODBUS Client • MODBUS Server • MODBUS Telemecanique TSX • MODBUS Valmet Damatic RTU • MODBUS RTU Quad2000 • MODBUS SCT PPU
SHOW TAG TagName /TagHOBJ **SHOW** STATION StationName| StationHOBJ **SHOW NAN**	Enables debug information for the specified I/O tag/station. The command requires either the name or HOBJ of the I/O tag/station. The debug information is usable for D2000 System developers. The *SHOW NAN* command displays all I/O tags whose value (current, last telemetry, or last valid) is NaN (unspecified type).

<table>
<tr><td>SHUTDOWN WINNT</td><td>Restarts the computer with the D2000 KOM process (the process must be running).</td></tr>
<tr><td>STALTERPATH StationName ON|OFF</td><td>For the Microtel 700 protocol, it explicitly activates (ON) or deactivates (OFF) an alternative communication route for a station named StationName (e.g., "B.Station").</td></tr>
<tr><td>START_RECORD dir_name_only [begin_time] end_time</td><td>Starts a data recording into a subdirectory (application directory) with the name <i>dir_name_only</i>. If the directory does not exist, it will be created.
If this TELL command is run from the D2000 Server, firstly, the file <i>dodm_values.dat</i> containing the initialization values of DODM objects is recorded and then the D2000 Server will send this command to clients.
Data are recorded within the time period <<i>begin_time, end_time</i>>. If <<i>begin_time</i>> is not set, data are recorded immediately.

Example:
Data files, recorded by the TELL command:
START_RECORD DirName "01-12-2008 08:22:27" "01-12-2008 11:02:07"

These four data files (it relates to four hourly intervals) are the result of recording:
KOM-SELF_2008_12_01_08.dat < 01-12-2008 08:22:27 ; 01-12-2008 09:00:00 >
KOM-SELF_2008_12_01_09.dat < 01-12-2008 09:00:00 ; 01-12-2008 10:00:00 >
KOM-SELF_2008_12_01_10.dat < 01-12-2008 10:00:00 ; 01-12-2008 11:00:00 >
KOM-SELF_2008_12_01_11.dat < 01-12-2008 11:00:00 ; 01-12-2008 11:02:07 >

Note: The TELL command uses <i>"dd-mm-rrrr hh:mm:ss"</i> format for the parameters of <i>DateTime</i> type.
It is possible to initialize this TELL command from the D2000 Server, as well as directly from the clients (currently only from the D2000 KOM and Gateway Client processes).
Data are recorded into the specified subdirectory in the application directory, where a separate data file is created for every hourly time period.</td></tr>
<tr><td>STOP_RECORD</td><td>Stops a data recording immediately, it is not necessary to wait for end_time. If the recording is not in progress, an error will be returned.

Note: It is possible to start up this TELL command from the D2000 Server, as well as directly from the clients (currently only from the D2000 KOM and Gateway Client processes).</td></tr>
<tr><td>START_REPLAY dir_name_only [begin_time [/NOW]]
or
START_REPLAY dir_name_only /LOOP</td><td>Starts a data replaying from a subdirectory (of application directory) with the name dir_name_only. If the directory does not exist, an error will be returned.
If the command is run from the D2000 Server - at first, the dodm_values.dat file containing the initialization values of DODM objects will be recorded by the D2000 Server, and then the D2000 Server will send this command to the clients with /NOW parameter.
It replays all data (in data files) with the time stamp >= begin_time. The timestamp of data represents a particular time from the recorded time period. The time of data replaying must be from the range of the recorded time period.

Example of data replaying:
Time Stamp (TS) - blue color, current time (CT) - red color

<table>
<tr><th>begin_time</th><th>/NOW</th><th>Description</th></tr>
<tr><td>no</td><td>no</td><td>• data are replayed immediately
• if CT = 02-12-2008 14:11:02, data are replayed from TS >= 01-12-2008 09:11:02
• if CT = 02-12-2008 14:33:55, data are replayed from TS >= 01-12-2008 08:33:55</td></tr>
<tr><td>yes</td><td>no</td><td>• if begin_time = 01-12-2008 08:27:27, data are replayed from time so that this time corresponds with a shift from hour
• if CT = 02-12-2008 14:11:02, data are replayed from TS >= 01-12-2008 08:27:27, when CT = 02-12-2008 14:27:27
• if CT = 02-12-2008 14:33:55, data are replayed from TS >= 01-12-2008 08:27:27, when CT = 02-12-2008 15:27:27</td></tr>
<tr><td>yes</td><td>yes</td><td>• if begin_time = 01-12-2008 08:27:27, the command finds the nearest time (time stamp) so that data could be replayed immediately
• if CT = 02-12-2008 14:11:02, data are replayed from TS >= 01-12-2008 09:11:02
• if CT = 02-12-2008 14:33:55, data are replayed from TS >= 01-12-2008 08:33:55

In this situation, also data, recorded before TS, are replayed. This data are replayed at the beginning.</td></tr>
</table>

If /LOOP is used, data replaying will be again started automatically. It can be stopped by the STOP_REPLAY command.

Note: This TELL command uses the "dd-mm-rrrr hh:mm:ss" format for the parameters of DateTime type.
It is possible to start this TELL command from the D2000 Server, as well as directly from the clients (currently only from KOM client).
The D2000 KOM process can be started up in replay mode only with the start parameter /Replay. It causes the process D2000KOM not to communicate with devices.</td></tr>
</table>

STOP_REPLAY	Stops a data replaying immediately. If the replaying does not work, the error will occur. **Note:** There is possible to start up this TELL command from the D2000 Server, as well as directly from the clients (currently only from KOM client).
STDOUT output	The command redirects the standard output to a new one. **Example:** D2000 on OpenVMS platform: *STDOUT "_TNA2:"* (output is redirected to the terminal *_TNA2:*) D2000 on Windows platform: *STDOUT C:\output.txt* (output is redirected to the file *output.txt*) **Note:** See a note for STDOUT command for D2000 Archiv.
STOP	Stops the D2000 KOM process.
STSTAT START/STOP StationName	Enables the communication (the parameter *START*) or disables the communication (parameter *STOP*) with the *StationName* station. The station will get the StON (START) or StOFF (STOP) value.
STWATCH StationName ["Number"]	**For protocols based on periodic polling of I/O tags (request-response protocols):** the command generates one or several requests for priority reading of all I/O tags on a specified station. The *Number* parameter defines the number of requests, possible value is within 1...5. If the parameter is not specified, a single request will be generated. **For protocols based on sending of changes (change-based protocols):** the command generates one or several requests for the reading of all data. The command is implemented for a selected set of communication protocols: • BACnet (sends ReadProperty, ReadPropertyMultiple, and Subscribe messages depending on the configuration of I/O tags) • DNP3 (sends requests for reading of all configured Poll Classes specified in the configuration of station parameters and for reading of values of all objects with "Explicit read" enabled) • IEC 60870-6 ICCP/TASE.2 (sends requests for reading values of all I/O tags) • IEC 61850 (sends requests for reading values of all I/O tags) • IEC 870-5-101 (sends Interrogation Command/Counter Interrogation Command depending on station protocol parameters) • IEC 870-5-104 (sends Interrogation Command/Counter Interrogation Command depending on station protocol parameters) • IEC 870-5-104 Server (sends Interrogation Command/Counter Interrogation Command depending on station protocol parameters) • IEC 870-5-104 Sinaut (sends Interrogation Command/Counter Interrogation Command depending on station protocol parameters) • OPC Data Access (sends requests for synchronous reading values of all I/O tags) • OPC UA (sends requests for reading values of all I/O tags)

D2000 EVENT

DYNAMIC_INFO	TELL command for the processes with the name *.EVH and *.HIS. This TELL command does not have any parameters. After receiving the command, the process writes, into its log file, information about: - list of all running instances of ESL scripts Format: **ESL;ESL name** ESL - text identifying the next information ESL name - script identification[1] - registered text strings (GETACCESS action) Format: **ACCESS;ESL name;String;time of GetAccess;bGlobal;userComment;status** ACCESS - text identifying the next information ESL name - script identification[1] String - registered text string time of GetAccess - time of string registration bGlobal - flag defining the string to be global userComment - comment status - text "IN PROGRESS", if the global string has been registered, otherwise empty text "" - list of all open XML documents (%XML_CreateDocument, %XML_OpenDocument) Format: **XML;ESL name;handle** XML - text identifying the next information ESL name - script identification[1] handle - unique numerical identifier of XML document; it is identical to handle which is used by functions %XML_* - list of objects that have been opened by function %OpenRefToObject Format: **REFOBJ;Esl Name;Synchro;Obj HOBJ;Obj Name;Open In Progress** REFOBJ - text identifying the next information ESL name - script identification[1] Synchro - value of parameter *_bSynchro* at calling of the function %OpenRefToObject Obj HOBJ - unique identifier of opened object Obj Name - unique name of opened object Open In Progress - flag defining if the object is just being opened - list of active database connections (by the DB_TRANS_OPEN, DB_CONNECT, PG_CONNECT, and SQL_CONNECT actions) Format: **DBCONNECT;ESL name;DB TRANS HANDLE;SUB CONNECT HANDLE;TYPE;OBJNAME;Prepared SQL Command;Last SQL Prepare; Comment** DBCONNECT - text identifying the next information ESL name - script identification[1] DB TRANS HANDLE - identifier of transaction connections (it is created by calling the DB_TRANS_OPEN action), or 0, if the connection is not transactional SUB CONNECT HANDLE - identifier of connection (it is created by calling the DB_CONNECT, PG_CONNECT, or SQL_CONNECT actions) TYPE - type of connection (SUB CONNECT HANDLE) OBJNAME - object name, on which the connection is joining Prepared SQL Command - flag of the existence of prepared SQL command Last SQL Prepare - format of the last SQL command which has been prepared by calling the SQL_PREPARE action Comment - the position where the SQL_PREPARE action was called - list of files which have been opened by functions %FIO_* Format: **FIO;ESL name;file name** FIO - text identifying the next information ESL name - script identification[1] file name - filename - list of created data containers (CNT_CREATE action) Format: **CNT;ESL name;handle;NR;valTyp;is array** CNT - text identifying the next information ESL name - script identification[1] handle - unique numerical identifier of data container NR - number of elements valTyp - a type of key which identifies the items in the data container uniquely is array - flag determining whether the action CNT_CNVTOARRAY is applied on the respective data container [1] - script identification: text string uniquely identifying the running ESL scrip. It contains the object name (HOBJ)[instance number] unique numerical identifier. Example: E.Script(728)[105]1872 (%GetSelfInstanceId, %GetSelfHBJ)

SHOW_TRANS_LIST idInstance	If *idInstance* = 0, the command shows both the list of all instances of events that are pending to finish the transaction and the list of transactions. If *idInstance* <> 0, the command shows the information about the particular instance (you can find the list of instances in the dialog window ESL Diagnostic Pack or by the **TASK_STATUS** command).
STATISTICS path+file	Writes data about the processor time consumption according to individual events into a file.
STDOUT output	The command redirects the standard output to a new one. ***Example:*** D2000 on OpenVMS platform: *STDOUT "_TNA2:"* (output is redirected to the terminal *_TNA2:*) D2000 on Windows platform: *STDOUT C:\output.txt* (output is redirected to the file *output.txt*) **Note:** See a note for STDOUT command for D2000 Archiv.
TASK_STATUS path+filename	Writes currently executed lines of all the script running in a specified process into a file (*.EVH or *.HIS). The command also supplies the information about whether the ESL script is currently executing the external function. If yes, the file will contain the name of the function and the line number of the script calling the function.
ON_CHANGE_INFO [idInstance]	The command writes all active ESL ON CHANGE actions within the process to the log file. If the optional idInstance parameter is specified, only the actions of the given instance of the ESL script (idInstance described in the ESL Diagnostic Pack tab Event Instances) will be listed.

D2000 CALC

DP ON/OFF	Debug Pipe - shows/hides the information about communication with D2000 Calc (only if the process window is displayed on the desktop).
RESET_STATISTICS	Resets the counters for evaluating the statistical data.
STATISTICS path+file	Writes statistical data about individual I/O tags into a file. The file contains the following information: I/O tag name, number of recalculations and number of new values broadcasted into the system. The counters are reset while starting the process D2000 Calc or using the command **RESET_STATISTICS**.
STDOUT output	The command redirects the standard output to new one. **Example:** D2000 on OpenVMS platform: *STDOUT "_TNA2:"* (output is redirected to the terminal *_TNA2:*). D2000 on Windows platform: *STDOUT C:\output.txt* (output is redirected to the file *output.txt*). **Note:** See a note for STDOUT command for D2000 Archiv.

CALCINFO ON|OFF clcName [row]	The command is used to determine the reason for calculating the eval tag that has the *Calculation method* parameter set to *At change* or *Trigger* value.
	
	The command ensures (**CALCINFO ON**) that into the LOG file of the process will be continually written information about the reason for the calculation of the eval tag. The **CALCINFO OFF** command terminates the continuous log to the LOG file. It is used for the eval tags that have the *Calculation Method* parameter set to *At change* or *Trigger* value.
	
	Record in the LOG file contains:
	
	• the identification of the object that caused the conversion and its value
	• the output value of the eval tag that was sent to the server
	
	Parameters:
	
	clcName - the name of eval tag
	
	row - specifies a row number for a structured eval tag. If not specified, the replacement value is 0 and all rows will be tracked.
	
	Example (the content of LOG file):
	
	TELL command: CALCINFO ON P.OnSec
	
	In the LOG file, there is a record that the information is about the calculation of the calculated **P.OnSec** object. The value of the object changed from HOBJ=20 (VALUE IN $**20**). Next, the individual attributes of the input value are displayed. Output value attributes are displayed after a row containing **VALUE OUT** content.
	
	`CALCINFO: P.OnSec[ 0]`
	`VALUE IN $20`
	`GValTyp : INT (Integer)`
	`Status : VALID`
	`LimitStatus : INLIMIT`
	`ProcAlarmStatus : NOALARM`
	`ValTyp : INT`
	`Flags : F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F`
	`ValTime : 20.11.2017 16:39:59.000`
	`AlarmTime : (null)`
	`Value : 59`
	`VALUE OUT`
	`GValTyp : INT (Integer)`
	`Status : VALID, NOACKVALUE`
	`LimitStatus : INLIMIT`
	`ProcAlarmStatus : NOALARM`
	`ValTyp : CE`
	`Flags : F,F,F,F,F,F,F,F,F,F,F,F,F,F,F,F`
	`ValTime : 20.11.2017 16:39:59.001`
	`AlarmTime : (null)`
	`Value : 60`

CHECK_DESTID_VALUES [DestId [ColIndex]] [FlagList]	The command enables to compare the values of destination columns (of structured variables), configured in the eval tags, with the values that were calculated by D2000 Calc. When specifying the particular object of *Structured variable* type, for which the comparison should be done, the value of parameter *DestId* , or also *ColIndex* for the particular column, must be a non-zero. It is followed by the optional keywords from the list: *Detail, Debug, Ignore_Time, ReCalc0s, ReCalc1m*. You may find the detailed information in the document Checking values of destination columns. **Note :** This command is mainly used for diagnostic purposes in D2000 Systems.
SHOW_DYN_INFO {Mask|HOBJ} [Row]	It enables displaying a configuration and the current status of a particular eval tag. The eval tag is defined as an input parameter either through its ID (HOBJ) or mask. If several eval tags match the mask, the first 20 tags will be displayed. If *Row* parameter is different than 0, it must be the eval tag, the values of which are written to a destination column of the structured variable. For common eval tags (those that have not a destination column) the *Row* parameter must be 0 or not specified. The log format is identical with the format [Calc], which is stated in the article Checking values of destination columns, Example 3. **Note:** This command is mainly used for diagnostic purposes in D2000 systems.

GET_SOURCE_VALUE HOBJ Row Col	It enables the displaying of the inputs into the calculation providing they are the inputs from the structured variables. The log format is identical with the format [DestVal] Checking values of destination columns, Example 3. **Note:** This command is mainly used for diagnostic purposes in the D2000 systems.

D2000 DBMANAGER

SHOW_HANDLE [table /database/structure_definition HOBJ] or [table mask]	Shows information on open descriptors. For more information see the chapter DBManager - debugging.
SHOW_CONNECT [database HOBJ [connect/transaction ID]] or [database mask [connect ID]] [DETAIL]	Shows information on open connections. For more information see the chapter DBManager - debugging.
SET_WATCHDOG database_mask seconds [NONTRANS]	Switches on the monitoring of connections that perform SQL commands longer than the specified time *seconds*. This information can be used for the profiling of applications and for the detection of slow SQL commands. If a NONTRANS clause is specified, transactional connections are not monitored. After the long-running SQL is finished, detailed info about it is written to a log file. More information can be found in chapter D2000 DBManager - debugging.
SET_WATCHDOG_QUEUE database_mask seconds [NONTRANS]	Switches on the monitoring of database actions that take longer to be processed (including waiting in queues of DBManager) than the specified time *seconds*. After such a database action is finished, detailed info about it is written to a log file. If a NONTRANS clause is specified, transactional connections are not monitored. More information can be found in chapter D2000 DBManager - debugging.
MONITOR_TRANS SHOW {ALL/<id>} [<file_path>] **MONITOR_TRANS** ON <history_depth_sec> **MONITOR_TRANS** OFF **MONITOR_TRANS** RESET	The mechanism for monitoring of the transactions. ON - activates the monitoring. <*history_depth_sec*> means a time-depth history of closed transactions. OFF - deactivates the monitoring RESET - resets the list of transactions (including the opened ones) SHOW - shows either the list of transactions including the history (*ALL*) or the particular transaction (<*id*>) to a console or file <*file_path*> in CSV format separated by ";" Columns in the list: - ID (transaction identifier) - time, task, traceback (time the last operation within the group of identical ones, task, traceback in the internal code) - comment, count (the last comment and the number of consecutive identical operations) (the identical operations has the same task and traceback) **Note:** After starting DBManager, the monitoring is switched off.
MONITOR_CONNECTS SHOW {ALL/<id>} [<file_path>] **MONITOR_CONNECTS** ON <history_depth_sec> **MONITOR_CONNECTS** OFF **MONITOR_CONNECTS** RESET	Monitoring the connections. ON - activates the monitoring. <*history_depth_sec*> represents the time-depth history of closed connections. OFF - deactivates the monitoring RESET - resets the list of connections (including the opened ones) SHOW - shows either the list of connections including the history (*ALL*), or the particular connection (<*id*>) to a console or file <*file_path*> in CSV format separated by ";" Columns in the list: - ID (auxiliary identifier of connection) - logon_time, logon_task, logon_traceback (time of connection, task and traceback in internal code) - logoff_time, logoff_task, logoff_traceback (time of disconnection, task, and traceback in internal code) - comment (the last written comment for the given connection, e.g. traceback in ESL code) **Note:** After starting DBManager, the monitoring is activated with the time-depth history of closed connections which is 3 hours (10 800 seconds).
REFRESH_TABLES seconds	The command is intended for use on the PostgreSQL platform. It is used to close and reopen cursors used in page access to tables so they don't block table vacuuming if these cursors are open for too long. Typically, these are schemes with browsers, which are e.g. open for several days. The parameter *seconds* is the minimum age of the cursor in seconds to be reopened (e.g. 86400).
TIME_STATISTICS database_mask [DETAIL]	Displays statistics of execution of individual database action types per-database or per-table (if a *DETAIL* parameter is specified). For more information see the chapter DBManager - debugging.

D2000 ALARM

DI ON/OFF [HOBJ/mask]	Debug Info for alarm objects - enables/disables displaying the information about the evaluation of alarm objects with specified HOBJ or with the name matching given mask. If neither *HOBJ* nor *mask* is specified, all alarm objects will be affected.
SHOW_DYN_INFO HOBJ / mask	Shows dynamic information on the specified alarm object(s) on the text console of the process D2000 Alarm. **Note:** The command is mainly used for D2000 system diagnostic purposes.

D2000 TOPOLOGY

SHOW_TOPOLOGY HOBJ or topology_name [DETAIL]	Shows the information about topology. For more information - see Topology - Topology debugging chapter.
DEBUG_TOPOLOGY HOBJ or topology_mask ON /OFF	Turns on/off writing out detailed information on topology evaluation. For more information - see the chapter Topology - Topology debugging.
SHOW_ASYMETRIC TERMINALS or TRANSFORMERS [topology_mask]	Lists asymmetric Terminals or Transformers in the topology. For more information - see Topology - Topology debugging chapter.

D2000 GateWay Client

GETOLDVAL remoteTagName "BeginTime" ["EndTime"] ["NORECALC"] **GETOLDVAL** remoteObjectMask "BeginTime" ["EndTime"] ["NORECALC"] **GETOLDVAL** "remoteTagName[row]^item "BeginTime" ["EndTime"] ["NORECALC"]	Reads the values of the remote tag within a specified interval. If the **EndTime** parameter is not specified, the current time will be used as the default value. Time format ("BeginTime", "EndTime") is "dd-mm-yyyy hh:mi:ss". The command may be used for example after a failure or stoppage of the application, which the **D2000 GateWay Client** process belongs to. It allows transferring the values from the archive of the remote application to the archive of the application. The only requirement of the value transfer is that the values of corresponding objects have been stored in the archive of the remote application. The first form of the command reads all values transferred by specified remote tags (as well as structured objects). The second form of the command reads all the values of several objects according to the mask. The third form reads the values of a part of structured object. The parameter *item* specifies a column of the destination structure (see the Destination structure parameter in the configuration of the remote tag). When a single row (column) needs to be read, use the following: • reading row 10 - *GETOLDVAL RemoteTagName[10], …* • reading the column named *Values* - *GETOLDVAL RemoteTagName[0]^Values, …* The TELL command must always be executed in the transaction mode - the COMMAND action (within the D2000 System Console process, the command is automatically executed in the transaction mode). After the command is executed, the values that were transferred will be stored in the archive and the D2000 Archive automatically recalculates corresponding statistics. The *NORECALC* parameter can be set at the end of the GETOLDVAL. In this case, the statistic calculations will not be executed by the D2000 Archiv process.
DI ON/OFF [HOBJ/mask]	Debug Info for remote objects - enables/disables displaying the information about processing the values of objects with specified HOBJ or with the name matching the given mask. **Note:** if the gateway client is run in a transparent gateway mode, use the **TRGTW** command to display debug information.
SHOW_DYN_INFO HOBJ / mask	Shows dynamic information on the specified remote object(s) on the text console of the process D2000 Gateway. **Note:** The command is mainly used for D2000 system diagnostic purposes.
TRGTW	Debug command for transparent gateway mode. The command lists the names of objects and their HOBJ on the local and remote systems.

START_RECORD dir_name_only [begin_time] end_time	Starts recording of values read from the GateWay Server. For details see the command description for the D2000 KOM process. **Note:** Replaying of recorded values by the D2000 KOM process is possible if the GateWay Client was running in a transparent gateway mode during the recording.
STOP_RECORD	Instantly stops recording of values read from the GateWay Server. For details see the command description for the D2000 KOM process.
START_REPLAY dir_name_only [begin_time [/NOW]] or **START_REPLAY** dir_name_only /LOOP	Starts replaying of data recorded by command **START_RECORD**. For details see the command description for the D2000 KOM process.
STOP_REPLAY	Instantly stops the replaying of data. For details see the command description for the D2000 KOM process.

D2000 WorkBook

OBJECTS_INFO	The process writes the list of all objects that are or were used in reports into its *.log* file.

Related pages:

D2000 Tell
Work with D2000 Tell