

DBCCONTROL

Reference Manual

Relative to version 600.01

Contents

| | |
|--|-----------|
| Recent Changes | 3 |
| May 2015 580.01:MCP 58 | 3 |
| Nov 2016 580.02:Allow monitoring of bigger databases | 3 |
| Introduction..... | 4 |
| Installing..... | 6 |
| Controlling which databases should be monitored | 7 |
| DBC_RESTRICTED | 7 |
| COLLECTOR Mode option..... | 7 |
| DBC_COLLECTOR | 7 |
| Controlling the display of DBCONTROL mix entries | 8 |
| DBC_SUPPRESS | 8 |
| Deciding on data file locations | 8 |
| DBC_FAMILY | 8 |
| Controlling the number of DBCONTROL tasks..... | 10 |
| DBC_MAXTASKS | 10 |
| Task CHARGECODE assignment | 10 |
| DBC_USECCS | 10 |
| Set a default CUMIN value | 11 |
| DBC_CUMIN..... | 11 |
| Set a default CUMAX value | 11 |
| DBC_CUMAX | 11 |
| Using DBControl..... | 12 |
| DBCONTROL codefiles | 12 |
| METALOGIC/DBCONTROL | 13 |
| METALOGIC/MAGUS..... | 13 |
| METALOGIC/DBCONTROL/SCREENLIB | 13 |
| METALOGIC/DBCONTROL/MONITOR..... | 13 |
| Running DBCONTROL | 13 |
| DBCONTROL in operation..... | 14 |
| Changing ALLOWEDCORE and BUFFERS..... | 15 |
| Transient databases..... | 15 |
| Database terminations | 16 |
| Recovery and Reorganizations..... | 16 |
| Terminating the DBCONTROL library | 17 |
| Terminating a DBCONTROL sub-task..... | 17 |
| Excluding databases | 17 |
| Monitor most databases..... | 18 |
| Exclude most databases..... | 18 |
| Precise interaction of DBC_RESTRICTED and marker files | 18 |
| US date formats | 18 |
| Future actions | 18 |
| Troubleshooting | 19 |
| Operator Interfaces | 20 |
| ACONLY [DBCONTROL sub-task only] | 20 |
| COLLECTOR [DBCONTROL sub-task only] | 21 |
| CUMAX [DBCONTROL sub-task only] | 22 |

| | |
|---|-----------|
| CUMIN [DBCONTROL sub-task only]..... | 23 |
| INTERVAL [Main DBCONTROL library]..... | 23 |
| INTERVAL [DBCONTROL sub-task only] | 24 |
| NORESTORE [DBCONTROL sub-task only] | 25 |
| PRIORITY [DBCONTROL sub-task only] | 25 |
| QUIT [Main DBCONTROL library] | 26 |
| REPORT [DBCONTROL sub-task only] | 27 |
| RESTART [Main DBCONTROL library] | 29 |
| SAMPLES [DBCONTROL sub-task only] | 29 |
| SAVE [DBCONTROL sub-task only] | 30 |
| STATISTICS [DBCONTROL sub-task only] | 31 |
| STATUS [DBCONTROL sub-task only] | 32 |
| TIMEOUT [DBCONTROL sub-task only] | 32 |
| TT Command Interface | 34 |
| DBC EXCL command | 34 |
| DBC HELP command | 35 |
| DBC INFO command | 35 |
| DBC INTERVAL command | 35 |
| DBC QUIT | 36 |
| DBC STATUS..... | 36 |
| DBC RESTART | 36 |
| DBC STOP | 37 |
| Using PARAMETERS files | 38 |
| Monitor-COMS interface | 40 |
| DATABASE PERFORMANCE ANALYSIS screen..... | 41 |
| DATABASE RUNTIME INFORMATION screen..... | 42 |
| DATABASE PERFORMANCE STATISTICS screen | 43 |
| CHANGE DBCONTROL RUNTIME PARAMETERS screen..... | 44 |
| Statistics files | 46 |
| Copyright | 48 |

Recent Changes

May 2015 580.01:MCP 58

This minor change reflects that DBCONTROL is compatible with MCP 58

Nov 2016 580.02:Allow monitoring of bigger databases

Values on the Database Performance Analysis screen which were greater than 99999999 were truncated. Values greater than 999999999 are now divided by 1000000 and displayed with a suffix of M. Values between 1000000 and 999999999 are divided by 1000 and displayed with a suffix of K. Values less than 1000000 are display with a suffix of W.

The default values of CUMAX and CUMIN may now be set via config variables DBC_CUMAX and DBC_CUMIN. These will be added to the Install program at a later date but may be set via Supervisor at present.

Eg TT / (\$DBC_CUMAX.CONFIG:="123456789") TT / (\$DBC_CUMIN.CONFIG:="54321")

Previously CUMAX and CUMIN could only be set to 10 digit numbers, whether via AX to the Monitor program. 12 digit numbers are now accepted.

Introduction

The control of resources allocated to a DMSII database can be a very complex operation. In particular, memory resource is dictated primarily by the amount of assigned ALLOWEDCORE, which can be critical for both batch and on-line applications. If the ALLOWEDCORE is too low, then throughput and response times may be unacceptably degraded; too high, and although those problems may then disappear, other applications running alongside the database may be degraded because of lack of memory. It should be remembered that database ALLOWEDCORE memory is always SAVE (i.e. non overlayable). Further, the demand on individual databases will fluctuate with time, so setting a high static ALLOWEDCORE for a database which may have a high transaction rate for short periods is wasting valuable memory resource.

The optimisation of database memory is further complicated by the MCP and DMSII handling of their own particular variants of SAVE memory. MCP will handle its own overlayable memory using standard virtual memory techniques; SAVE memory cannot be overlayed. Most database memory buffers are SAVE but DMSII does have the capability to overlay this memory, paging buffers in and out, upon demand. This results in two separate overlay rates for systems running DMSII: an MCP overlay rate for overlayable memory and a DMSII overlay rate for its own save memory.

This "juggling" of memory resource between the demands of one or more databases and other system processes can be volatile, requiring constant attention especially where available memory shortage is acute. In such situations, Metalogic's DBCONTROL is designed to dynamically control the memory parameters of a database to balance system and database response time and reduce overall memory consumption for database buffers. Research has shown that the optimal balance for database memory handling is achieved when DBCONTROL has matched the overlay rates for both the database and system.

DBCONTROL will continuously monitor individual database memory usage and how it is affected by the overall system demand for memory, i.e. both DMSII and MCP overlays rates are continuously checked. If DMSII's overlay rate is lower than MCP's, this indicates that DMSII has too much memory and DBCONTROL will lower the database ALLOWEDCORE. However, if DMSII's overlay rate is higher than the system, DMSII has too little memory and its ALLOWEDCORE is increased.

This process is done automatically without intervention but may be easily controlled and refined through a variety of runtime parameters, which provide the user with considerable flexibility. For each database that DBCONTROL is permitted to control, a separate task is invoked allowing this control to be readily performed at a local level.

If desired, individual databases may be configured to use site-selected minimum and maximum ALLOWEDCORE thresholds which DBCONTROL will not transgress. These values, called CUMIN and CUMAX respectively, are dynamically controllable through several user interfaces into the main DBCONTROL program.

Also, DBCONTROL has in-built statistics gathering capabilities which allow the collection of performance data to flat files on the A-series system. This data can be readily uploaded to a PC for import into any spreadsheet for analysis, charting etc. The COLLECTOR option in DBCONTROL can be used to monitor databases and collect performance data without DBCONTROL changing ALLOWEDCORE settings. This allows "before-and-after" effects of DBCONTROL on a production system using databases with their normal ALLOWEDCORE settings as a benchmark.

In summary, DBCONTROL can considerably improve overall system performance by continuously adjusting individual database memory parameters on a more regular and consistent basis than any human.

Installing

This section discusses the configuration aspects that need to be considered during the installation of DBCONTROL.

The **Metalogic Software Installation** reference manual includes the basic instructions for installing all Metalogic software and includes a short section on some of the questions that are asked by the INSTALL utility, specifically about DBCONTROL's operational environment.

All Metalogic software is installed using the INSTALL utility. The software is provided in Unisys wrapped container format and INSTALL provides a menu-driven interface for selective software installation.

During the first installation of the software, the INSTALL utility will present the screen shown below:

```

Metalogic S.a.R.L.
Install Utility
60.600.02

DBControl Configuration

These are config settings applicable to DBCONTROL only.

1. DBControl will be restricted to run .... FALSE Range True/False
   for user-specified databases?
2. DBControl will run in Collector mode ... TRUE Range True/False
3. DBControl will suppress all its mix .... TRUE Range True/False
   entries?
4. Family used for RESTORE, PARAMS and .... DEV
   STATS files? Leave blank for defaults
5. Maximum number of DBCONTROL tasks ..... 255 Range 1<10<256
   allowed to run simultaneously?
6. Charge codes will be assumed ..... FALSE Range True/False
7. Default CUMIN ..... 987654 Max 549755813887
8. Default CUMAX ..... 123456789 Max 549755813887

Action: OK
        OK / Quit / Log
```

These settings can also be maintained by running META/INSTALL with a parameter of CONFIG.

```

Metalogic S.a R.L
Install Utility
60.600.02

Interrogation of Product Variables

1. Supervisor          6. Full FLEX
2. TRIM                 7. Cachepack
3. Waitwatcher          8. System
4. DBcontrol            9. Install
5. FLEX Inquiry

Action: 4
        Quit / Spcfy or Enter Number / Prev or Spcfy in Action / Log
```

and then entering the DBCONTROL menu.

```
Screen 1 of 1                               Metalogic S.a R.L
                                           Install Utility
                                           60.600.02
                                           DBcontrol Configuration Variables

1. DBC_RESTRICTED .. FALSE
2. DBC_COLLECTOR ... TRUE
3. DBC_SUPPRESS .... TRUE
4. DBC_FAMILY ..... DEV
5. DBC_MAXTASKS .... 255
6. DBC_USECCS ..... No value defined
7. DBC_CUMIN ..... 987654
8. DBC_CUMAX ..... 123456789

Action ▶ ◀
Quit / Spcfy on entry or Enter Number / Prev or Spcfy in Action / Log
```

The questions presented on these screens are discussed in the rest of this section.

Controlling which databases should be monitored

DBC_RESTRICTED

Q: DBControl will be restricted to run for user-specified databases?

A: TRUE/FALSE

Answering 'FALSE' is typical and allows DBCONTROL to run against all databases. It is possible to exclude a specific database.

e.g. (USER)MYDBNAME, by creating a file called:

***METALOGIC/DBCONTROL/USER/MYDBNAME**

For exclusion to operate correctly, this file must have a FILEKIND other than DCALGOLCODE and be loaded onto the same family as the DBCONTROL codefile location.

Answering 'TRUE' ensures that DBCONTROL will only run against a database where the above file is resident and has a FILEKIND DCALGOLCODE (see the section on "[Excluding Databases](#)" later in this section.

COLLECTOR Mode option

DBC_COLLECTOR

Q: DBControl will run in Collector mode only (i.e. no changes to ALLOWEDCORE)?

A: TRUE/FALSE

DBCONTROL has the capability to run in a special mode called 'COLLECTOR'. This allows new or trial sites to collect database performance information and store it into statistics files for future analysis. When this option is active, DBCONTROL

does NOT change ALLOWEDCORE.

A "TRUE" to the above question will force DBCONTROL to run in a special information collection mode without making dynamic changes to ALLOWEDCORE. By default, the setting of this global option will apply to all databases that are controlled by DBCONTROL though, it is possible to override this setting by changing local options for the DBCONTROL son through the monitor or AX interfaces (see [Operator Interfaces](#) and [MONITOR COMS interface](#) for more information).

The STATISTICS option is automatically set if collector mode is active, though this may also be overridden at the local level for an individual database.

Controlling the display of DBCONTROL mix entries

DBC_SUPPRESS

Q: DBCONTROL will suppress all of its mix entries?

A: TRUE/FALSE

Answering 'TRUE' to this question means that each DBCONTROL monitor task will only show in the response to an "A ALL" ODT command. In this case, DBCONTROL will automatically issue a SUPPRESS mix command to each of his sons during their individual initialisations.

The answer 'FALSE' means all of these sub-tasks will appear as normal active entries (i.e. visible with the "A" command only).

It should be noted that the main DBCONTROL stack never appears in the active entries since it is a freeze controlled library. Hence, it will only be visible using the "LIBS" ODT command.

Alternatively but less preferred, the mix entries for DBCONTROL and its sub-tasks can be automatically suppressed by equating the title of the logical file SUPPRESS to TRUE by using the WFL MODIFY command:

```
MODIFY METALOGIC/DBCONTROL; FILE SUPPRESS=TRUE
```

Deciding on data file locations

DBC_FAMILY

Q. Family used for RESTORE, PARAMS and ..DBCONTROL uses several data file types to hold information about an individual STATS files? Leave blank for defaults database.

A. Alphanumeric text - maximum of 17 characters

In all the following cases, the <USERCODE> and <DATABASE> fields in each filename are self-explanatory:

- **RESTORE** file holds the original allowed core of the database. This file is not modifiable, except by DBCONTROL, and is usually named:

- ***METALOGIC/DBCONTROL/<USERCODE>/<DATABASE>/RESTORE**
- **PARAMETERS** file which holds any run-time DBCONTROL tuning parameters for the specified database. This text file is easily modifiable by the user and is named:
- ***METALOGIC/DBCONTROL/<USERCODE>/<DATABASE>/ PARAMETERS**
- **STATISTICS** file which is used to hold daily performance data for the specified database in both collector and non-collector modes. The data can be uploaded to a PC for performance analysis. Files are generated on a daily basis for each individual database and are called:

***DBC/<USERCODE>/<DATABASE>/STATISTICS/<DAYNAME>**

where <DAYNAME> is SUNDAY, MONDAY etc.

Unless otherwise specified, these files will by default be created and maintained on disk families where, usually, each DMSUPPORT library is found, though this may not always be the case. In effect, the maintenance of all these files, in particular statistics files, becomes difficult because each database has its own family where these files are held. Further, over time, it is possible for these locations to change due to DASDL updates or changes enforced using DMCONTROL.

Therefore, DBCONTROL has an option to centralise all such data files in one disk family location, which will be asked during the first installation:

Q: Currently, DBCONTROL does NOT place the RESTORE, PARAMETERS and STATISTICS files on a specific family. Do you wish to change this assignment? (Enter YES or NO)

A: TRUE/FALSE

Answering "TRUE" will permit the user to subsequently select a suitable family name, which will be the repository for all DBCONTROL data files.

Alternatively, if a family has been previously specified in an earlier installation and the user has elected to change the existing parameters, INSTALL will ask the following.

Q: Currently, DBCONTROL expects to locate all RESTORE PARAMETERS and STATISTICS files on <family name>. Do you wish to change this assignment? (Enter YES or NO)

A: YES/NO

If "YES" is answered in both cases, INSTALL will prompt for a user-selected family name:

Q: Family for DBCONTROL's statistics and resource files

A: <pack name>

Enter the name of the family you wish to use to hold these files. The main DBCONTROL library must be quit and restarted to reflect this change.

Configuration variable: DBC_FAMILY (17-CHARACTER STRING)
Example: U META/INSTALL DBC_FAMILY=PACK

Controlling the number of DBCONTROL tasks

DBC_MAXTASKS

Q: Maximum number of simultaneous DBCONTROL users? (1<10<256)

A: Number between 1 and 256

This gives the maximum number of database monitor tasks that DBCONTROL can handle simultaneously. By default, this value is 10. It should be noted that each additional controlled database would add about 250 words of memory to the working set of the program.

DBCONTROL will issue a warning message to any attempt to monitor a database, which will take the current limit beyond the maximum number of tasks.

Configuration variable: DBC_MAXTASKS (Integer 1eq 256)

Example: U META/INSTALL DBC_MAXTASKS=255

Task CHARGECODE assignment

DBC_USECCS

DBCONTROL has the ability to set a default charge code automatically to the various usercoded processes that it spawns. There are two ways that DBCONTROL can enforce this behaviour. The first is by using the MAKEUSER utility to set the CHARGERREQ attribute to those usercodes that regularly run databases controlled by DBCONTROL. It is important that each usercode possesses at least one charge code in its respective CHARGECODELIST. If both these usercode attributes are assigned, then DBCONTROL will automatically assign the first charge code found in this list to any usercoded task that is processed.

An alternative method without enforcing CHARGERREQ on each usercode, is to consider the following question asked by INSTALL:

**Q: Should DBCONTROL force Charge Code assumption
for USERCODED databases? (Enter YES or NO)**

A: YES/NO

A "YES" answer to the above question will cause DBCONTROL to behave as if CHARGERREQ had been set for all usercodes and the first CHARGECODE from the relevant usercode's CHARGECODELIST, if it is present.

Answering "NO" implies that the CHARGECODELIST should only be used if the USECHARGERREQ attribute has been set for the specified usercode in the USERDATAFILE.

In the event that CHARGECODE assignment is enforced but no CHARGECODELIST is present for an individual usercode, DBCONTROL will issue warnings and continue normally.

Configuration variable: DBC_USECCS (TRUE/FALSE)

Example: U META/INSTALL DBC_USECCS=TRUE

Set a default CUMIN value

DBC_CUMIN

The default value of CUMIN may be set. This value will be used to set the CUMIN value if no [Parameters file](#) is resident. [CUMIN](#) will be set to this value

Configuration variable: DBC_CUMIN (Integer leq 549755813887)

Example: U META/INSTALL DBC_CUMIN=1234

Set a default CUMAX value

DBC_CUMAX

The default value of CUMAX may be set. This value will be used to set the CUMIN value if no [Parameters file](#) is resident. [CUMAX](#) will be set to the maximum of this value and the original allowed core of the database.

Configuration variable: DBC_CUMAX (Integer leq 549755813887)

Example: U META/INSTALL DBC_CUMAX= 549755813887

Using DBControl

Before running the DBCONTROL in earnest, it is important to have decided upon the run-time parameters assigned during the installation process discussed in [Installing DBControl](#). In particular, whether to run in collector mode or not depends on whether you want to benchmark DBCONTROL's effects on your databases in a "before" and "after" scenario, using the "before" scenario as a control (i.e. no dynamic ALLOWEDCORE changes).

Dbcontrol will run with the ACONLY option set for any new database. This means that buffers settings and overlay goal will not be changed.

It is strongly suggested that overlay goal be set to a non zero value, even if very small.

Jim Stewart makes strong suggestions on buffer settings in the [MCP BLOG](#)

His suggestions are:

The random buffer setting can be set to 1, and the serial buffers set to 2 to enable readahead. The system buffers should be set to a high value, for example 2000 or more, to ensure that buffer allocation does not constrain database performance as ALLOWEDCORE is adjusted..”

Remember also that using collector mode also automatically sets DBCONTROL's statistics option, so each database that DBCONTROL is permitted to run against will be collecting data in its own set of statistics files.

It is also practical to allow DBCONTROL to monitor all databases by default (i.e. **DBC_RESTRICTED=FALSE**), especially if you are running in collector mode, and then decide if there is a need to exclude any specific databases. Typically, DBCONTROL can fail to improve some databases because of special requirements or DMSII parameter settings. The [exclusion mechanism](#), which checks for resident files of a specific file kind, can be used to achieve this.

DBCONTROL supports Unisys mainframe environments of up to 256 databases and runs effectively on systems with the MORETASKS system option set (supporting mix numbers up to 65535).

DBCONTROL codefiles

During the installation process, various critical codefiles will be loaded by the INSTALL utility:

```
*METALOGIC/DBCONTROL
*METALOGIC/DBCONTROL/SCREENLIB
*METALOGIC/DBCONTROL/MONITOR
*METALOGIC/MAGUS
```

These files are discussed below.

METALOGIC/DBCONTROL

This codefile is the primary DBCONTROL program, which monitors all database activities, including BOJ notices. When run for the first time, DBCONTROL will make itself an MCS in the network DATACOMINFO file because it requires certain MCP privileges available only to an MCS.

DBCONTROL will process a separate task for each database that it is required to run against; each task will inherit the usercode, accesscode and optionally, chargecode from the database stack and also the name of the task will be constructed from the original database name. Each DBCONTROL process will have a library linkage to the relevant DMSUPPORT library process.

METALOGIC/MAGUS

This special codefile is critical to the successful operation of the DBCONTROL program. As a library, it is loaded and SL-ed by the INSTALL program and is responsible for handling software key licensing and much of the intelligence used to monitor and control system and database memory allocations.

METALOGIC/DBCONTROL/SCREENLIB

Library to support the Monitor screens.

METALOGIC/DBCONTROL/MONITOR

The SCREENLIB library and MONITOR programs support a remote user interface into the DBCONTROL program, allowing runtime control of individual database-DBCONTROL relationships and viewing of on-line statistics information. This interface is COMS-window based and requires some simple set-up using COMS UTILITY, as discussed in [MONITOR COMS interface](#).

All DBCONTROL run-time parameters may be changed through a series of formed screens rather than using the less preferred ACCEPT mechanism.

Running DBCONTROL

DBCONTROL should be run from the ODT using the following command:

```
??RUN METALOGIC/DBCONTROL ON <FAMILY>
```

where <FAMILY> is the disk family where the *SYSTEM/= directory of files typically are resident. DBCONTROL codefiles are automatically loaded to this user-selected location during the installation process.

If it is necessary to ensure that DBCONTROL is automatically invoked by the MCP after a halt-load, then this facility can be conferred onto DBCONTROL by using either of the following commands:

```
CS METALOGIC/DBCONTROL ON <FAMILY>  
AI + METALOGIC/DBCONTROL ON <FAMILY>
```

Note that the CS command is scheduled for de-implementation in later Unisys MCP releases and that the AI command is only available on MCP release 42.0 or later.

If DBCONTROL is invoked by a normal RUN command, DBCONTROL will automatically rerun itself as an independent job to save the overhead of an additional job stack. Alternatively, DBCONTROL can be run through WFL, incurring a small core penalty, which is useful to control restart capabilities. Using the task attribute SW8 set to TRUE within a RUN statement will inhibit the re-run action.

For example:

```
PROCESS RUN *METALOGIC/DBCONTROL ON MYPK;SW8=TRUE
```

will tell DBCONTROL not to initiate itself as a separate JOB stack.

When DBCONTROL is first run, it will check to see if it's codefile has privileged capabilities. If not, the following message will be displayed:

```
CODEFILE MUST BE PP'ED
```

and the program will terminate. If this occurs, the codefile must be given privilege status using the MP command from an ODT:

```
MP METALOGIC/DBCONTROL ON <FAMILY> + PU
```

DBCONTROL can also simulate a <mixnumber>LP for his own mixnumber and any of his sub-tasks to help protect against accidental operator DS.

DBCONTROL in operation

Once running, DBCONTROL scans for all DMSII databases currently active. If DBCONTROL is permitted to 'control;' the database, processes will be invoked to link to each database and begin monitoring ALLOWEDCORE.

If DBCONTROL is advised that a new database has started or an active database has been discovered during initialisation and there are no exclusion restrictions, DBCONTROL will start a new process called:

```
*METALOGIC/DBCONTROL/<DBUSER>/<DBNAME>
```

This process will attempt to link to the database support software. After a successful linkage, the DBCONTROL task will be able to commence monitoring operations immediately for the database. Note that this task will inherit the usercode of the original active database.

Each DBCONTROL process will need to determine the location of the database CONTROL file and the DMSUPPORT library. Whilst a database is being monitored, the DBCONTROL process will remain linked to the appropriate DMSUPPORT library.

The main DBCONTROL stack will appear as a freeze-controlled LIBRARY stack. Its mixnumber may be obtained by using the ODT commands, 'MCS' or 'LIBS NAME=DBCNTROL=' or, even better, using the wildcard AA NAME command.

```
AA NAME=DBC=
---Mix-Pri--CPU Time----- 3 ACTIVE ENTRIES (ALL) =DBC=-----
 2958  50      1:14 Lib *METALOGIC/DBCNTROL
* 5761  50      :00    *METALOGIC/DBCNTROL/MTODB2
* 5751  50      :00    *METALOGIC/DBCNTROL/MTODB1
```

The advantage of the AA command is that the DBCONTROL main stack and all son processes are visible in the same response.

The following screen display shows the typical messages that are seen during DBCONTROL's initialisation phase. Here, mixnumber 4409 is the main library task, whilst mixnumber 4410 is the DBCONTROL monitor task, which is linking to a database called METATAPELIB (seen as part of the message identity) which runs under the usercode TAPELIB..

```
MSG
---Mix-Time----- 21 MESSAGES -----
* 5761 12:21 DBC:MTODB2:COLLECTOR OPTION RESET
* 5761 12:21 DBC:MTODB2:STATISTICS OPTION DISABLED
* 5761 12:21 DBC:MTODB2:REPORTING TO LIBRARY IS INACTIVE
* 5761 12:21 DBC:MTODB2:IDLE TIMEOUT IN MINUTES = 0
* 5761 12:21 DBC:MTODB2:METALOGIC/DBCNTROL/MTODB2/PARAMETERS ON DEV.
* 5761 12:21 DBC:MTODB2>Loading params:
* 2958 12:21 DBC:START:RUNNING DBCNTROL FOR:MTODB2 [1]
* 5755 12:20 NXSERVICES: Attached IMACKP IP address = 10.0.0.10
* 5751 12:20 DBC:MTODB1:COLLECTOR OPTION RESET
* 5751 12:20 DBC:MTODB1:STATISTICS OPTION DISABLED
* 5751 12:20 DBC:MTODB1:REPORTING TO LIBRARY IS INACTIVE
* 5751 12:20 DBC:MTODB1:IDLE TIMEOUT IN MINUTES = 0
* 5751 12:20 DBC:MTODB1:METALOGIC/DBCNTROL/MTODB1/PARAMETERS ON DEV.
* 5751 12:20 DBC:MTODB1>Loading params:
* 2958 12:20 DBC:START:RUNNING DBCNTROL FOR:MTODB1 [0]
```

Changing ALLOWEDCORE

Using its own default run-time parameters, each DBControl sub-Task will wake up every 20 seconds and re-calculate DMSII Allowedcore settings for the database it is monitoring.

Transient databases

It has been observed that multiple DBCONTROL tasks have entered the system for individual databases that entered and left the mix very quickly, staying active for only a few seconds. A second DBCONTROL process *can* appear if the previous DBCONTROL son has connected to another DMSUPPORT library associated with an earlier invocation of the database.

In such circumstances, but depending on the DBCONTROL interval (default 2 minutes), these additional tasks would remain active unless manually quit.

DBCONTROL performs significant checking on the active state of a database during the DMSUPPORT linking phase. If such a transient database is detected, the DBCONTROL son will either abort himself (if no new database exists) or will attach himself to the "new" database if one is present. The following messages may be seen in such circumstances:

```
DBC: TRY: (TST)TESTDB2: (TST)TESTDB2 - CANNOT FIND DMSUPPORT
DBC: TRY: (TST)TESTDB1:ABORT:DATABASE has already gone to EOJ
DBC: TRY: (TST)TESTDB1:ABORT:DATABASE already active
DBC: TRY: (TST)TESTDB1:Database mix substitution detected for DBNAME
```

The first three messages are associated with an abort of the DBCONTROL son. The last message indicates a successful switch from a database already gone to EOJ to a new invocation of the same database.

Database terminations

If the DBCONTROL sub-task discovers that it is the last remaining user of the database, the process will remain active for a further 5 minutes before going to end of task, though this time-out is alterable by the user.

If DBCONTROL is using the restore option for the specified database, the original DMSII settings will be preserved into a disk file with the same name as the DBCONTROL task name, but with an extra node appended, "RESTORE".

```
*METALOGIC/DBCONTROL/<USERCODE>/<DATABASE>/RESTORE
```

Meanwhile, the DBCONTROL library stack will be checking all BOJ events invoking new sub-tasks, where appropriate, to monitor any new databases.

Recovery and Reorganizations

If DBCONTROL becomes active during a run of SYSTEM/DMRECOVERY, it is possible that the DBCONTROL process will fail to detect an in-use CONTROL file for a recovering database. Typically, DMRECOVERY can open and close a CONTROL several times during recovery and it is possible for DBCONTROL to encounter the CONTROL file in a closed state.

In such circumstances, DBCONTROL will recheck the CONTROL file every 30 seconds for a maximum of 5 attempts. The following information message will be displayed during this process:

```
SEARCHING FOR OPEN CONTROL FILE
```

Similarly, if DBCONTROL started to monitor a database undergoing an on-line reorganization, DBCONTROL will handle linkages to DMSUPPORT libraries which have name changes i.e. a reorganization-generated DMSUPPORT may be named with a suffix of '/<number>'.

Terminating the DBCONTROL library

The main DBCONTROL stack can be terminated by issuing either of the following ODT commands to the DBCONTROL library:

```
<mixnumber>AX QUIT DBCONTROL  
TT DBC QUIT
```

All DBCONTROL sub-tasks will automatically be terminated, though where DBCONTROL is monitoring many databases, this may take some time.

When terminated, each DBCONTROL sub-task will automatically restore the original settings of the database, if required. If either the DBCONTROL library or one of the sub-tasks terminates abnormally, DBCONTROL will still attempt to restore the original settings. It is strongly recommended that DBCONTROL is always terminated using one of the above QUIT commands.

Terminating a DBCONTROL sub-task

Individual DBCONTROL sub-tasks may also be terminated if desired by using one of the following methods:

```
<mixnumber>AX QUIT  
TT DBC STOP <DB mixnumber>
```

where <mixnumber> is the current DBCONTROL sub-task mixnumber and <DB mixnumber> is the mixnumber of the database stack that you wish to stop monitoring. The main DBCONTROL library will then mark this database as "excluded" and will not try to run against it until either the database or DBCONTROL itself is restarted.

As before, the original DMSII settings will be restored during DBCONTROL's termination unless the NORESTORE option is set.

It should be also noted that any manual changes to the ALLOWEDCORE DMSII parameters using the Visible DBS (i.e. the ODT SM command) will be lost if DBCONTROL is already monitoring a database.

It is possible to prevent the original parameter settings from being restored for an individual database by setting the NORESTORE option in the PARAMETERS file for the specified database, or by changing it using the MONITOR COMS-window interface (see [MONITOR COMS interface](#)).

Excluding databases

As discussed earlier, DBCONTROL will by default attempt to monitor all DMSII databases that are encountered during initialisation or whose BOJs have been detected. However, in some circumstances, there may be a requirement for certain databases to be excluded from this automatic behaviour which can be effected by the behaviour of the configuration variable, DBC_RESTRICTED coupled with the residence and characteristics of specifically named files, recognisable to

DBCONTROL. There are three variants of these “marker” files.

All have the title format:

```
*METALOGIC/DBCONTROL/<USERCODE>/<DATABASE> ON <FAMILY>
```

where <USERCODE> is the usercode of the database, <DATABASE> is the name of the database. Please note that the <USERCODE> node will be absent for databases that do not run under a specific usercode. <FAMILY> should be the the same family where *METALOGIC/DBCONTROL runs from, normally DISK.

The “Include” file must have a filekind of DCALGOLCODE.

The “Job” file must have a filekind of JOBSYMBOL

The “Exclude” file may have any filekind except DCALGOLCODE or JOBSYMBOL.

Monitor most databases

Set DBC_RESTRICTED to FALSE and create an “exclude” file for the databases to be ignored.

Exclude most databases

Set DBC_RESTRICTED to TRUE and create an “include” file for the databases to be monitored.

Precise interaction of DBC_RESTRICTED and marker files

A resident “Job” file will be started and the database will be monitored

A resident “Exclude” file will prevent the database from being monitored

If DBC_RESTRICTED=TRUE then an “include” file or “Job” file must be resident for each Database to be monitored and there must be no “exclude” file for that database.

US date formats

Some of the statistics reports and monitor screens generated by DBCONTROL will display military dates in European format, by default (i.e. DD/MM/YY). For customers wishing to show US-style date formats (i.e. MM/DD/YY), this option can be easily changed using the INSTALL program:

```
U META/INSTALL US_DATES=TRUE
```

If this option is changed whilst DBCONTROL is active, the program must be QUIT and restarted for the new date formats to be displayed.

Future actions

Once DBCONTROL has been established and is running against all or some of your databases, it is important to review some of the run-time parameters that

DBCONTROL will use for each database. In particular, the values for CUMAX and CUMIN provide thresholds above and below which ALLOWEDCORE may not be set. By default CUMIN is 5000 words of memory whilst CUMAX is 500000 words.

In reality, it is likely that there will some very large databases, which will need much higher thresholds than these to run effectively. The values may be changed using the AX or TT interfaces (see [Operator Interfaces](#)), by creating or altering the PARAMETERS file (see [PARAMETERS Files](#)) or directly using the MONITOR COMS window interface (see [MONITOR COMS interface](#)).

Troubleshooting

There are various error and informational messages that DBCONTROL can display from both the main library stack and the sub-tasks. Most of the errors relate to license key expiration or invalidation and, in particular, the following message is fatal:

MAGUS: UNAUTHORISED ATTEMPT TO USE PRIVILEGED PROCEDURE

This error will occur during DBCONTROL initialisation and has a number of possible reasons:

- The trial period for DBCONTROL has now expired.
- DBCONTROL does not have a valid license key for the host machine, as designated by style and system serial number.
- In these cases, please contact Metalogic for further assistance.
- The DBCONTROL codefile has the following constraints:
- The codefile must reside under the '*' directory
- The codefile name must have the text "METALOGIC" as one of its nodes
- The codefile must have been conferred privilege status using the ODT command:

MP *METALOGIC/DBCONTROL + PU

Operator Interfaces

There are two possible means of communicating with both the DBCONTROL library and its sub-tasks. The older mechanism, which relies on an operator AX interface, is now being replaced by a simple ODT command syntax, which uses the TT prefix (this same principle is used by Metalogic's SUPERVISOR product).

This section is divided into two subsections: the first deals with the entire available AX commands available to both the library and sub-task. They are presented in alphabetical order but only a few of the commands are applicable to both and will be marked as such. The subset of AX commands available to the main DBCONTROL library have global effects and allow the program to be terminated whereas the set of commands available to each sub-task will only affect the operation of that individual process and the database to which it is linked.

| Command | Available to library | Available to sub-task |
|------------|----------------------|-----------------------|
| ACONLY | x | YES |
| COLLECTOR | x | YES |
| CUMIN | x | YES |
| CUMAX | x | YES |
| HELP | YES | |
| INTERVAL | YES | YES |
| NORESTORE | x | YES |
| PRIORITY | x | YES |
| QUIT | YES | YES |
| RESTART | YES | x |
| SAMPLES | x | YES |
| SAVE | x | YES |
| STATISTICS | x | YES |
| STATUS | x | YES |
| TIMEOUT | x | YES |

The next subsection deals with an alternative and user-friendlier interface, which uses the TT ODT command to communicate with the DBCONTROL library stack. Using this common interface, commands may be input which affect the main DBCONTROL library and individual sub-tasks as well.

It is recommended that you use this interface for rapid control of the DBCONTROL environment and use the COMS-window interface for the control and monitoring of individual databases. Although the AX mechanism is still viable, the convenience of the other two alternative interfaces is generally preferred.

ACONLY

[DBCONTROL sub-task only]

———— <sub-task mixnumber> — AX — ACONLY —————|

The ACONLY command prevents DBCONTROL from changing the BUFFER or OVERLAYGOAL settings of an individual DMSII database, and restricts it to modifying the ALLOWEDCORE setting only. This option is appropriate if the database buffer configurations have been carefully optimised by a database administrator and should not be changed.

If ACONLY has been saved in the local parameters file and it is desired to disable the option, this cannot be done through the AX interface. The parameters file must be manually amended to remove the relevant line. Alternatively, the DBCONTROL monitor COMS-window facility can be used to reset the option and preserve the parameters file with ease.

The following display message will be generated by the DBCONTROL sub-task for the METATAPELIB database when the ACONLY command is issued:

```
1234AX ACONLY
DBC:METATAPELIB:ACONLY OPTION SET
```

NOTE: Setting this option only takes effect at the next initiation of the DBCONTROL task for the relevant database and that a SAVE must be input to preserve the run-time parameters.

See also:

[SAVE command](#)
[MONITOR COMS-window interface](#)

COLLECTOR

[DBCONTROL sub-task only]

```
——— <sub-task mixnumber> — AX — COLLECTOR — 

|  |   |  |
|--|---|--|
|  | + |  |
|  | . |  |

 ———
```

The COLLECTOR option, when set with the '+' modifier, forces DBCONTROL to run in a special statistics-only mode which allows the collection of database performance information without DBCONTROL making changes to ALLOWEDCORE assignments.

This feature means that the "before" and "after" effects of DBCONTROL on one or more databases can be judged by comparing performance information with COLLECTOR deactivating DBCONTROL's memory management algorithms.

When COLLECTOR is set, the STATISTICS option is also automatically set and will override any such assignment in the parameters file. The performance information will be written to a data file, which is named differently from the usual statistics file naming convention:

```
*DBC/<DBUSER>/<DBNAME>/STATISTICS/<DAYOFWEEK>/NOAC
```

The suffix of 'NOAC' means NOALLOWEDCORE and signifies that DBCONTROL

See [STATISTICS Files](#) for more details.

```
1234AX COLLECTOR+
DBC:METATAPELIB:COLLECTOR OPTION SET - NO ALLOWEDCORE CHANGES
1234AX COLLECTOR-
DBC:METATAPELIB:COLLECTOR OPTION RESET
```

Note that, as with all DBCONTROL sub-task commands, the **SAVE** command is required to make this change permanent.

See also:

SAVE command

MONITOR COMS-window interface

CUMAX

[DBCONTROL sub-task only]

<sub-task mixnumber> — **AX** — **CUMAX** — **<integer>**

Sometimes it may be felt that it is acceptable for a database to endure performance problems during heavier loads, to protect other applications. To assist with this concept, the CUMAX parameter enforces an **upper limit** on the ALLOWEDCORE setting that can be assigned to the database by DBCONTROL. Even if DBCONTROL calculates that more memory than this maximum value is required, it will not increase ALLOWEDCORE any further.

It should be noted that any attempts to circumvent this behaviour by manually assigning a higher ALLOWEDCORE through the Visible DBS interface, would be overridden by DBCONTROL during its next monitor cycle.

By default, DBCONTROL will use a value for CUMAX of 500,000 words; its maximum permitted value is 9,999,999,999. When changed through the AX interface, the following message will be displayed for, say, a database called METATAPELIB:

```
1234AX 1500000
DBC:METATAPELIB:MAX_ALLOWEDCORE (IN WORDS) = 1500000
```

Changes to this value are dynamic; DBCONTROL will use the new threshold immediately. But, as with other sub-task commands, a **SAVE** is required to make the change permanent.

Alternatively, the DBCONTROL monitor COMS-window facility can be used to change CUMAX and apply it to the parameters file with ease.

See also:

CUMIN command

SAVE command

MONITOR COMS-window interface

CUMIN

[DBCNTROL sub-task only]

<sub-task mixnumber> — AX — CUMIN — **<integer>**

When there is no user activity against a database for long periods of time,, DBCONTROL will gradually decrease that database's ALLOWEDCORE allocation until it reaches a specified threshold, called CUMIN. Similar to the effect of CUMAX, DBCONTROL will not drop ALLOWEDCORE below this value.

Typically, CUMIN should be a high enough value to ensure that sufficient memory requirements are available for a new database transaction to be processed in a reasonable time. Generally, DBCONTROL's memory algorithms are such that if there is a drastic increase in transactions for a database that had previously been dormant for a long time, then DBCONTROL is able to detect the significance of such a "spike" and respond rapidly. It should be remembered that DBCONTROL can only allocate sufficient ALLOWEDCORE to this database if both memory resource and system performance permits.

By default, DBCONTROL will use a value for CUMIN of 5,000 words; its maximum permitted value is 9,999,999,999. When changed through the AX interface, the following message will be displayed for a database called, say, METATAPELIB:

1234AX 140000

```
DBC:METATAPELIB:MIN ALLOWEDCORE (IN WORDS) = 140000
```

Changes to this value are dynamic; DBCONTROL will start to use the new threshold immediately. But, as with other sub-task commands, a SAVE is required to make the change permanent.

See also:

CUMAX command

SAVE command

MONITOR COMS-window interface

INTERVAL

[Main DBCONTROL library]

`<library mixnumber>` `AX` `INTERVAL` `<integer>`

When issued to the main DBCONTROL library, the INTERVAL parameter controls

the rate at which DBCONTROL samples the mix for new database stacks entering the system. By default, DBCONTROL will "poll" the system every 120 seconds, by default, though this "polling" is not strictly necessary since DBCONTROL receives event notifications from the MCP whenever a database job starts. The `<integer>` modifier reflects the interval value in seconds.

Also during the mix check, the library will also check that all DBCONTROL sub-tasks are active and will restart monitoring against previously excluded databases, if required.

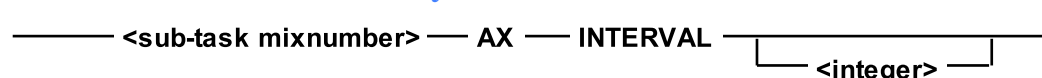
For example, changing the INTERVAL sample period to 3 minutes display the following from the DBCONTROL library:

1000 AX INTERVAL 180
INTERVAL = 180 SECONDS

Note that this setting is temporary and cannot be saved over restarts of DBCONTROL.

INTERVAL

[DBCCONTROL sub-task only]



At the sub-task level, the INTERVAL parameter controls the frequency with which an individual DBCONTROL task will evaluate the state of the database.

If the **STATISTICS** parameter has been set for the relevant DBCONTROL, the **INTERVAL** parameter also defines the frequency at which statistical , information is updated. However, it is the **SAMPLES** parameter that dictates how often this information is physically written to the statistics files.

By default, DBCONTROL will use a value for **INTERVAL** of 20 seconds; its maximum permitted value is 9,999,999,999. When changed through the AX interface, the following message will be displayed:

```
1234AX INTERVAL 40
INTERVAL = 40 SECS
```

Some caution should be used when changing the **INTERVAL** parameter; too small a value will increase the processor overhead for an individual DBCONTROL sub-task. Too large a value for **INTERVAL** and the DBCONTROL will not be able to respond rapidly enough to volatile, variable database activity.

Although, a change to **INTERVAL** is dynamic, a **SAVE** command must be issued to save the parameters file and make the change permanent.

See also:

[SAMPLES command](#)

[SAVE command](#)

[MONITOR COMS-window interface](#)

NORESTORE

[DBCNTROL sub-task only]

———— <sub-task mixnumber> — AX — NORESTORE —————|

The **NORESTORE** parameter inhibits DBCNTROL from restoring the allowedcore settings of the database to its original values i.e. before DBCNTROL had started running against the database. , as can be seen in the response to an SM STATUS command. The default value is always FALSE.

If **NORESTORE** has already been saved in the local parameters file and it is desired to disable the option, this cannot be done through the AX interface. The parameters file must be manually amended to remove the relevant line. Alternatively, the DBCNTROL monitor COMS-window facility can be used to reset the option and preserve the parameters file with ease.

The following display message will be generated by the DBCNTROL sub-task when the ACONLY command is issued:

```
1234AX NORESTORE
NORESTORE OPTION SET
```

NOTE: The NORESTORE has no effect for a database that is running within DBCNTROL's COLLECTOR mode.

See also:

[SAVE command](#)

[MONITOR COMS-window interface](#)

PRIORITY

[DBCNTROL sub-task only]

———— <sub-task mixnumber> — AX — PRIORITY —————|
 | |
 | <integer> |

In general, it is more effective to allocate resources purely on the basis of total system efficiency. However, occasionally, a need arises to give greater resources to one process at the expense of total system capacity. This can be achieved in DBCNTROL by setting the PRIORITY parameter for an individual database. The <integer> value must be between 1 and 99 and the PRIORITY corresponds to a modified system overlay rate calculated by the following algorithm:

```
MODIFIED OLAY RATE% =  
(OLAY RATE%* MIN((100-PRIORITY) , 50) /MAX(MIN(PRIORITY, 50) ,1))
```

Since DBCONTROL tries to balance the current system overlay rate to against the database's current overlay rate, it can be seen from this formula that changes to PRIORITY will strongly influence DBCONTROL's memory balancing algorithms.

The default value for PRIORITY is 50, which means that the above equation will always evaluate such that the actual and modifier overlay rates are the same:

```
MODIFIED OLAY RATE% = (OLAY RATE%* 50) /50)
```

However, if PRIORITY was set to, say , 75:

```
MODIFIED OLAY RATE% = (OLAY RATE%* 25/50) = OLAY RATE% * 0.5
```

In this case, a PRIORITY of 75 would make DBCONTROL believe that the system overlay rate was half of its actual value; this could allow DBCONTROL to allocate higher ALLOWEDCORE settings than usual for a database (if resources permit) to help minimise database overlays.

In practice, Metalogic advises against changing PRIORITY from the default value however but there are some exceptions where its use may be advantageous:

- If running DBCONTROL on a very memory-bound system tends to slow the database application response down consistently, look to see whether DBCONTROL is setting the ALLOWEDCORE higher or lower than normal. If it is higher, try reducing the PRIORITY; if lower, try increasing it.
- If the system seems to be performing better generally, but an important database is slowing down, try increasing the PRIORITY for that database.
- If a large test database tends to hit the machine badly whenever it runs, even with DBCONTROL running, try reducing its PRIORITY.
- Although, a change to PRIORITY is dynamic, a SAVE command must be issued to save the parameters file and make the change permanent. Alternatively, as with all other database parameters, the DBCONTROL COMS monitor interface can be used to change and save the setting of PRIORITY.

See also:

[SAVE command](#)
[MONITOR COMS-window interface](#)

QUIT

[\[Main DBCONTROL library\]](#)

```
———— <sub-task mixnumber> ——— AX ——— QUIT ——— DBCONTROL ———|
```

The QUIT command directed to the DBCONTROL main program causes DBCONTROL to cease monitoring all databases (as though QUIT commands had been entered for each database), and causes the main library program to terminate as soon as the last sub-task has gone. Please note that this variant of the QUIT command must have the additional text "DBCONTROL" or the quit will be refused.

1000AX QUIT DBCONTROL
QUITTING - ALL DBCONTROL TASKS WILL BE TERMINATED

All DBCONTROL sub-tasks will de-link from their relevant DMSUPPORT libraries and, unless the NORESTORE option is set for, the original DMSII parameters of each database will be restored.

[[DBCONTROL sub-task only](#)]

———— <sub-task mixnumber> — AX — QUIT ——————|

QUIT causes DBCONTROL to stop controlling the database. restoring the original database parameters before going to EOJ (unless the NORESTORE option is set). If the database continues to remain active, DBCONTROL will mark it as "excluded" which will prevent the DBCONTROL library from trying to monitor the database again at a later time.

1234AX QUIT
DBC:METATAPELIB:QUITTING

To reactivate an excluded database, the RESTART AX command must be input to the main DBCONTROL library to either restart all or individual excluded databases. The exclusion and restart of DBCONTROL tasks can also be controlled using the TT command interface which allows a more flexible option than the AX interface (see later in this section).

If a halt-load occurs while DBCONTROL is active, each individual sub-task does not have the opportunity to restore individual database parameters. If it is required to restore database parameters in this case, running DBCONTROL and QUITting it, once all the DBCONTROL sub-tasks are active, will automatically restore all database parameters.

REPORT

[[DBCONTROL sub-task only](#)]

———— <sub-task mixnumber> — AX — REPORT ——————|
 + LIB ——————|
 - ——————|

The REPORT command allows an alternative method of recording time-based statistical data from individual DBCONTROL tasks instead of using the STATISTICS option. When REPORT is activated using the "REPORT +" command, the relevant DBCONTROL task will expect to call an entry point in a SL-ed library, called DBCREPORTLIB, after every sample point passing a single message with data such as overlay rates, ALLOWEDCORE etc. Where appropriate, DBCONTROL may also pass some informational messages. In practice, this feature allows the

user to write his own tailored program interfaces to handle DBCONTROL statistical data.

The layout of the message passed by DBCREPORTLIB library has the following format:

```
AL = AAAA CU = CCCC OR = RRR SO = SSS
```

These figures reflect the latest DBCONTROL figures for the specified database, where AL is the ALLOWEDCORE, CU is the buffer core in use, OR is the current buffer overlay rate, and SO is the system overlay rate.

The details for the DBC_ REPORT entry-point in the DBCREPORTLIB library must be as shown below:

```
PROCEDURE DBC_ REPORT  
    (DBSMIXNO, DBUSER, DBNAME, ALC, CUC, ORC, SOC, MSG) ;  
VALUE DBSMIXNO, ALC, CUC, ORC, SOC ;  
INTEGER DBSMIXNO,  
    ALC,           %ALLOWEDCORE  
    CUC,           %BUFFER CORE IN USE  
    ORC,           %CURRENT BUFFER OVERLAY RATE  
    SOC;           %SYSTEM OVERLAY RATE  
STRING DBUSER,      %USERCODE OF DATABASE  
    DBNAME,         %NAME OF DATABASE  
    MSG;            %TEXT AS WOULD HAVE BEEN DISPLAYED IF  
                        %REPORTING TO SCREEN
```

REPORT- will deactivate the reporting feature; if no "+/-" appears, the current status is displayed. The default value is OFF.

```
1234AX REPORT + LIB  
DBC:METATAPELIB:REPORTING TO LIBRARY ACTIVATED
```

Values for the minimum and maximum values for DBCONTROL's allowed core calculations determined during the execution will be displayed at EOT if reporting is enabled. If reporting is not enabled, these values will be logged via an ODT LJ command.

Two information messages may also be sent via the DBC_ REPORT entry-point and may also be seen on the ODT:

```
DBC:METATAPELIB:CONFIGURING DB FOR ZERO USERS
```

This message means that the DBCONTROL sub-task is the only user of the database, so the database will remain in a quiescent state. No further displays will appear unless another program uses the database. DBCONTROL itself will quit after TIMEOUT minutes.

```
DBC:METATAPELIB:NORMAL DB ACTIVITY REQUIRED TO REDUCE CORE  
USAGE
```

This means that the database is so quiet that the normal memory control mechanisms cannot be used effectively, so there may be a delay before

ALLOWEDCORE changes affect the current core in use. Please note that both of the above messages will be displayed as system messages if the "REPORT -" option is disabled.

Since the release of DBCONTROL version 40.400.39, the REPORT options PRINTER and PACK have been de-implemented since the new statistics and COMS on-line monitoring capabilities were implemented. commands now replace these functions.

RESTART

[Main DBCONTROL library]

```
——— <sub-task mixnumber> — AX — RESTART ————  
                                     |  
                                     | <db mixno> |  
                                     | ALL      |  
                                     |—————|
```

To resume control of a specific "excluded" database for which DBCONTROL had been previously terminated, the **RESTART <db mixno>** command can be used to monitor the database stack, specified by the **<db mixno>** parameter.

```
1234 AX RESTART 4321
```

DBCONTROL will not issue any response to this command unless an error is detected i.e. the mixnumber is invalid or not a database.

To resume the control of all "excluded" databases, the **RESTART ALL** command can be used. Note that the normal exclusion mechanism using DBC_RESTRICTED and the codefile checks will still override a **RESTART ALL**.

This command is a quick way of forcing DBCONTROL to start monitoring a database that may not yet have been detected. It becomes a "last resort" option that invariably will coerce DBCONTROL to start monitoring any missing databases.

SAMPLES

[DBCONTROL sub-task only]

```
——— <sub-task mixnumber> — AX — SAMPLES ————  
                                     |  
                                     | <integer> |  
                                     |—————|
```

This command need only be used for a DBCONTROL database that is writing data to statistics files (i.e. the STATISTICS option is set). The parameter controls the number of normal DBCONTROL statistics samples that should be processed before writing a record to the statistics file. This allows the user to control the number of disk I/Os over time, dictated by the balance of **INTERVAL** time and **SAMPLES** rate. The default value is 12.

```
1234AX SAMPLES 20
```

```
DBC:METATAPELIB:SAMPLES PER STATS RECORD = 20
```

For example, a DBCONTROL sub-task which has an INTERVAL of 60 seconds and a default SAMPLES rate of 12, will enforce one statistics record to be written every 720 seconds, or 12 minutes.

Decreasing the **SAMPLES** parameter and keeping **INTERVAL** the same will increase the number of records written to the statistics file. This sort of value is entirely dependent on the needs of the user and how much statistics information he requires. See [Statistics files](#) for more details on the format of information written to these files.

Although, a change to **SAMPLES** is dynamic, a **SAVE** command must be issued to save the parameters file and make the change permanent. Alternatively, as with all other database parameters, the **DBCONTROL COMS** monitor interface can be used to change and save the setting of **SAMPLES**.

See also:

[INTERVAL](#) command

[SAVE](#) command

[MONITOR COMS-window interface](#)

SAVE

[[DBCONTROL sub-task only](#)]

———— <sub-task mixnumber> — AX — SAVE —————|

The **SAVE** command cause all of the current user-defined parameter settings for an individual database to be written to a **JOBSYMBOL** file. This created file, called **PARAMETERS**, allows the automatic re-load of user settings during a **DBCONTROL** restart and has the following name:

***METALOGIC/DBCONTROL/<USERCODE>/<DATABASE>/PARAMETERS**

Since the file is of type **JOBSYMBOL**, it may be easily amended by the user if required. It should be noted that a **SAVE** will not created a new file if all the settings are default values. A **SAVE** is required if any **DBCONTROL** sub-task run-time parameter has been changed and needs to be preserved. Any existing file will always be automatically overwritten.

1234AX SAVE

New params file: *METALOGIC/DBCONTROL/<USERCODE>/<DATABASE>/PARAMETERS

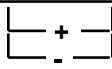
The **SAVE** command may also be used from the **COMS-window** monitor interface, where parameters files can be easily modifier and saved for individual databases.

See also:

[MONITOR COMS-window interface](#)

STATISTICS

[DBCONTROL sub-task only]

—— <sub-task mixnumber> — AX — STATISTICS — 

When the STATISTICS option has been set, the relevant DBCONTROL sub-task will start recording statistical information to a flat file which resides on the family specified by the DBC_FAMILY configuration variable. The statistical data includes mean ALLOWEDCORE, current core, database and system overlay rates plus interval-based values for normal and forced database overlays and transaction count.

This file holds information in CSV (Comma-Separated-Variable) format and may be easily read from CANDE. The CSV format allows the [STATISTICS files](#) to be readily downloaded by various Unisys terminal emulation programs directly into a PC or Macintosh spreadsheet. It is also very accessible via the browsing facilities within the MONITOR interface.

Unlike previous implementations in DBCONTROL, statistical information is not immediately written to disk after each sample. To reduce disk space and make the statistics information more controllable, a data record is only written after a user-specified number of samples have been processed. A "sample" is defined as the time at which a DBCONTROL sub-task extracts memory allocations and overlay rates from the system and the relevant database. This frequency is controlled by the **SAMPLES** parameter, whose default value is 12, and **INTERVAL** which is the time in seconds between two samples. So, for an **INTERVAL** of 20 seconds and a **SAMPLES** value of 12, a new statistics record will be written every 4 minutes (12*20 =240 seconds).

Each statistics file is called:

***DBC/<DBUSER>/<DBNAME>/STATISTICS/<DAYOFWEEK>**

where <DBUSER> is the usercode of the running database (if present; if the database is non-usercoded then this level is **not** used), <DBNAME> is the name of the database and <DAYOFWEEK> is Monday, Tuesday etc.

As close to midnight as possible, each DBCONTROL will close the old statistics file and open a new one with a title reflecting the change of day.

INTERVAL command
SAMPLES command
STATISTICS Files
MONITOR COMS-window interface

[DBCCONTROL sub-task only]

The STATUS command will give a series of simple display messages indicating the current settings of an individual DBCONTROL sub-task parameters.

The above command will generate messages similar to the following:

This information is also available using the on-line monitor interface, which is more convenient to use than the AX command interface and is discussed in more detail in **MONITOR COMS-window interface**.

[DBCONTROL sub-task only]

When a DBCONTROL sub-task discovers that it is the only program using a database, it will wait TIMEOUT minutes before going to end of task. The default value is 5 minutes. After this period, the DBCONTROL task will go to EOJ, restoring all original database parameters if the NORESTORE parameter is not set.

DBC:METATAPELIB:IDLE TIMEOUT (IN MINUTES) = 5

Although, a change to **TIMEOUT** is dynamic, a **SAVE** command must be issued to save the parameters file and make the change permanent. Alternatively, as with all other database parameters, the DBCONTROL COMS monitor interface can be used to change and save the setting of **TIMEOUT**.

See also:

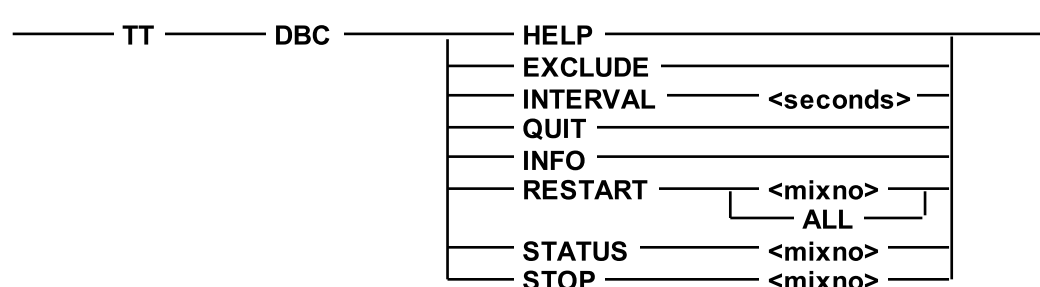
[SAVE command](#)

[MONITOR COMS-window interface](#)

TT Command Interface

Instead of using the AX interface, DBCONTROL can also detect input from the ODT or from a Metalogic SUPERVISOR COMS window. Similar to the SUPERVISOR product, DBCONTROL can detect **TT** operator commands allowing the user to pass requests directly without having to know DBCONTROL library or sub-task mix numbers.

The full set of commands available to DBCONTROL is shown below:



All commands destined for DBCONTROL must have the prefix "**TT DBC**"; this ensures that SUPERVISOR, who also monitors for TT commands, will know to ignore this particular command. Similarly, DBCONTROL will ignore all commands that do not have the "**DBC**" prefix. Also, it should be noted that, for ODT responses, DBCONTROL is not currently capable of multi-page output.

If SUPERVISOR is available but DBCONTROL is **not** active then SUPERVISOR will return the following error message:

```
'METALOGIC/DBCONTROL IS NOT AVAILABLE'
```

Although the subset of commands is not as rich or functional as the AX or COM monitor interfaces, **TT DBC** commands are quick and easy to use and very convenient if working from an ODT.

The list of available commands are discussed in the following sections:

DBC EXCL command

The **EXCL** command will show all current "excluded" databases i.e. those that have been stopped (by the **DBC STOP** command) or an individual DBCONTROL sub-task has been **QUIT** using the AX interface.

Example

```
TT DBC EXCL
```

```
DBCONTROL:EXCLUDE LIST: 7533,4567,1290,4872
```

The mixnumber shown refer to active databases only; if one of the entries should subsequently terminate, it will be removed from the list and its excluded status will be reset. If the same named database is then restarted, DBCONTROL will automatically start monitoring it.

DBC HELP command

The DBC HELP command will show the following information page

```
TT DBC HELP

----- METALOGIC/DBCONTROL ON-LINE HELP -----

--- DBC ----- HELP -----
+----- EXCLUDE -----+
+----- INTERVAL -----+ <seconds>
+----- QUIT -----+
+----- INFO -----+
+----- RESTART -----+ <DB mixno>
|----- ALL -----|
+----- STATUS -----+ <DB mixno>
+----- STOP -----+ <DB mixno>

'DBC EXCL' will show current EXCLUDED databases
'DBC HELP' will show this information page
'DBC INTERVAL' will change the main DBCONTROL task's monitor interval
'DBC QUIT' will permanently terminate all DBCONTROL tasks
'DBC INFO' provides global DBCONTROL run-time information
'DBC STATUS' provides DBCONTROL run-time information about a specified DBS
'DBC RESTART' will restart DBCONTROL for ALL or specific EXCLUDED databases
'DBC STOP' will stop DBCONTROL for the specified database (EXCLUDEs)
```

DBC INFO command

This command provides run-time information about DBCONTROL itself and the databases that it is currently monitoring.

```
TT DBC

----- METALOGIC/DBCONTROL INFORMATION -----
METALOGIC DBCONTROL LIBRARY VERSION 53.530.1 (Mixno:2958)
MAGUS VERSION 53.530.27
MONITOR INTERVAL is 120 seconds
USE 'TT DBC HELP' FOR LIST OF AVAILABLE COMMANDS
ACTIVE DATABASES CURRENTLY BEING MONITORED:

5748:MT0DB1
5759:MT0DB2
```

For each database, the **DBC INFO** command will show mixnumber and name information. Up to a maximum of 30 databases only will be shown in the response.

DBC INTERVAL command

Similar to the **INTERVAL AX** command to the DBCONTROL library, this offers a simple alternative to will change the main DBCONTROL task's monitor interval to the parameter provided:

```
TT DBC INTERVAL 240

DBCONTROL:MONITOR INTERVAL NOW SET TO 240 SECONDS
```

Please see the AX interface command, [INTERVAL](#), earlier , for more information.

DBC QUIT

This command will permanently terminate all DBCONTROL sub-tasks and the library stack itself.

```
TT DBC QUIT
DBCONTROL:QUITTING ALL DBCONTROL TASKS NOW
```

```
C
```

```
---Job--Task-Time--Hist----- 254 COMPLETED ENTRIES -----
* 2958/ 2958 14:35 E0J  JOB *METALOGIC/DBCONTROL
* 2958/ 5751 14:35 E0T  *METALOGIC/DBCONTROL/MTODB1
* 2958/ 5770 14:35 E0T  *METALOGIC/DBCONTROL/MTODB2
```

Please see the AX interface command, [QUIT DBCONTROL](#), earlier , for more information.

DBC STATUS

The provides run-time information about a specified database stack and the associated DBCONTROL sub-task:

```
TT DBC STATUS 5759

----- DBCONTROL RUN-TIME INFO for DBS 5759 -----
Database      : MTODB2
DBC taskno    : 5761
ACONLY        : RESET
Priority       : 50
Statistics    : RESET
CUMIN         : 5000
CUMAX         : 500000
Interval      : 20 seconds
NoRestore     : RESET
Timeout       : 0
Collector     : RESET

----- DYNAMIC DATABASE INFO -----
Current core usage          1627
Current ALLOWEDCORE value   5000
Total FORCED overlays      0
Total audited transactions  1
Original ALLOWEDCORE        30000
Original OVERLAYGOAL        5 %
Current DB olay rate        0 %
```

DBC RESTART

This command will restart DBCONTROL for a specific "excluded" database or, alternatively, by using the **ALL** modifier, then monitoring for all such databases will be restarted:

```
TT DBC RESTART 5759
DBCONTROL:ATTEMPTING TO RESTART DBCONTROL FOR DATABASE 5759
```

Note that an excluded database is one that has had a DBCONTROL previously stopped and that closing and restarting a database does not retain any excluded status. In this case, as soon as the database is active again, it must again be stopped if the DBCONTROL needs to be disabled.

Please see the AX interface command, [RESTART](#), earlier , for more information.

DBC STOP

The DBC STOP command allow the termination of a DBCONTROL sub-task for a specific database. The mixnumber of the database, not the DBCONTROL sub-task, must be given.

```
TT DBC STOP 5759  
DBCONTROL:DBC TASK FOR DATABASE 5759 WILL BE TERMINATED
```

This is an alternative to the AX interface command, [QUIT](#) , discussed earlier.

Using PARAMETERS files

The settings of individual DBCONTROL parameters such as REPORT, INTERVAL etc. can be retained in a symbolic file, using the SAVE command, with the same name as the individual task's name but with the additional appended node "PARAMETERS". The title is constructed thus:

METALOGIC/DBCONTROL/MYUSER/MYDB/PARAMETERS

and will reside on the family assigned by the DBC_FAMILY configuration variable or, if that is unavailable, DBCONTROL will search all other disk families. If more than one such named file is found, then the file with the most creation date and time will be used.

However, during its initialisation phase, each DBCONTROL sub-task will search for a global parameters file named:

***METALOGIC/DBCONTROL/PARAMETERS**

This file can hold default DBCONTROL settings for all databases; DBCONTROL will initially search for the file on the DBC_FAMILY specification but, if unsuccessful, will search on all other disk families. After the sub-task has processed the global parameters file, then DBCONTROL will search for the individual database parameters. Settings provided in these individual database files will override any global parameter setting.

These files should be any symbol or data file kind (e.g. DCALGOLSYMBOL, DATA, JOBSYMBOL) and should have a SECURITYTYPE=PUBLIC and SECURITYUSE=IN or IO. Only the first 72 characters of each file record is used and should consist of the command that would be presented to the AX interface to change each setting.

For example, in CANDE:

```
MAKE TEMP DATA
100 INTERVAL 15
200 STATISTICS +
300 TIMEOUT 10
400 NORESTORE
500 CUMIN 20000
SAVE
```

This temporary file, TEMP, should then be loaded to the appropriate family with its name changed, e.g.

WFL CHANGE TEMP TO *METALOGIC/DBCONTROL/DBUSER/DBNAME/PARAMETERS

The sub-task AX command, [SAVE](#), will automatically preserve any non-default parameters to the above file that might be required for an individual database. When the DBCONTROL sub-task for the specified database is restarted and the parameters file has been found, a series of display messages will be generated indicating the new defaults and the file location.

Note also that PARAMETERS files can be easily created and modified from the DBCMONITOR window, allowing all run-time parameters to be updated simultaneously.

See Also

[MONITOR COMS interface.](#)

Monitor-COMS interface

DBCONTROL can also provide a COMS window interface which allows one or more users to access and change DBCONTROL run-time parameters and monitor real-time statistical information collected by individual DBCONTROL tasks. This interface, hereafter called MONITOR, provides a series of formed screens to any MARC station allowing all monitored databases to be viewed on-line and individually changed.

The implementation requires some set up using COMS utility to define a program and window and access to several ancillary programs:

```
*METALOGIC/DBCONTROL/MONITOR
*METALOGIC/DBCONTROL/SCREENLIB/MONITOR
```

These files will normally be loaded during the installation process.

It is suggested that the program and window are named DBCONTROL, but of course, this is not a requirement. The following COMS UTILITY definition for the DBCONTROL program is suggested below:

```
CREATE PROGRAM DBCONTROL
REMOTE_FILE = Y , REMOTE_USERS = 1
, TITLE= *METALOGIC/DBCONTROL/MONITOR
, USERCODE = ". "
, ACCESSCODE= ". "
, CHARGECODE= ". "
, TASK_EQUATION = "PRIORITY = 50; OPTIONS = FAULT,
  ARRAYS, AUTORM, LIBRARIES;"
, SECURITY_CATEGORY_LIST = ALL , DATABASE = NONE
, MIN_COPIES= 0 , MAX_COPIES = 3
, INITIATION_TIME_LIMIT = 00:00
, TERMINATION_TIME_LIMIT = 00:00 , INPUT_QUEUE_MEMORY_SIZE = 0
, INSTALLATION_DATA = NONE
;
```

In the above example, **MIN_COPIES** should always be 0 and **MAX_COPIES** can be set to any site-chosen value.

For the DBCONTROL window, the following definition is suggested:

```
CREATE WINDOW DBCONTROL
MAX_USERS = 2 , MAX_DIALOGS= 1
, MAX_TRANCODE_SIZE = 0
, REMOTE_FILE = Y
, REMOTE_PROGRAM = DBCONTROL
, MCS = N
, NOTIFY_OPEN = Y
, NOTIFY_ON = Y
, ON_TEXT = "]REFRESH"
;
```

The site should also add its own security restrictions using COMS **WINDOWLISTs** and **STATIONLISTs** to ensure that only specific users and/or stations can access

Once the program, window and security lists are defined, the user can invoke a DBCONTROL monitor station by entering from MARC:

Note that if DBCONTROL is inactive at this time, COMS will invoke the program manually.

You may terminate monitoring at any time by using **BYE** in the action line from any of the menus.

The following is the home screen once the window has been opened:

From this screen the user can select an individual database entry and use the specify key to examine run-time data associated with the DBCONTROL sub-task and the database itself. If there is more than 13 database entries, using '+' or '-' will traverse backwards and forwards through the list. 'BYE' will close the window and return to MARC.

The screen also includes a snapshot of the current memory availability and system overlay rates. This is of interest since DBCONTROL attempts to balance database and system overlay rates for optimal performance.

To provide real-time information about a specific database, pressing the specify key on one of the database entries will take you into the following screen.

DATABASE RUNTIME INFORMATION screen

The information presented on the screen shown below includes DBCONTROL-related parameters and various useful database run-time parameters . Data of interest include totals for forced and normal overlays and transactions which are indicators of database performance.

```

14:44 METALOGIC/DBCONTROL 05/11/09
Database Runtime Information

Database name : MTODB1
Mixnumber : 5748 DBControl taskno : 5777
AOnly : RESET Interval : 20 sec
Norestore : RESET Timeout : 0 min
Priority : 50 Samples/stats rec : 12
Specified CuMax : 500000 Reporting to lib : INACTIVE
Specified CuMin : 5000 Statistics option : INACTIVE
COLLECTOR mode : INACTIVE COLLECTOR pending : INACTIVE

**** You may choose CHANGE to modify the above parameters ****

DYNAMIC DATABASE INFORMATION

Current core usage : 0 Total FORCED o/lay : 0
Current ALLOWEDCORE : 5000 Total NORMAL o/lay : 0
Current olay rate : 0.00 % Total transactions : 0
Original ALLOWEDCORE : 30000 Original OVERLAYGOAL: 5 %

Action : ▶ ( Change Stat Return Bye ) or SPECIFY will RETURN ◀

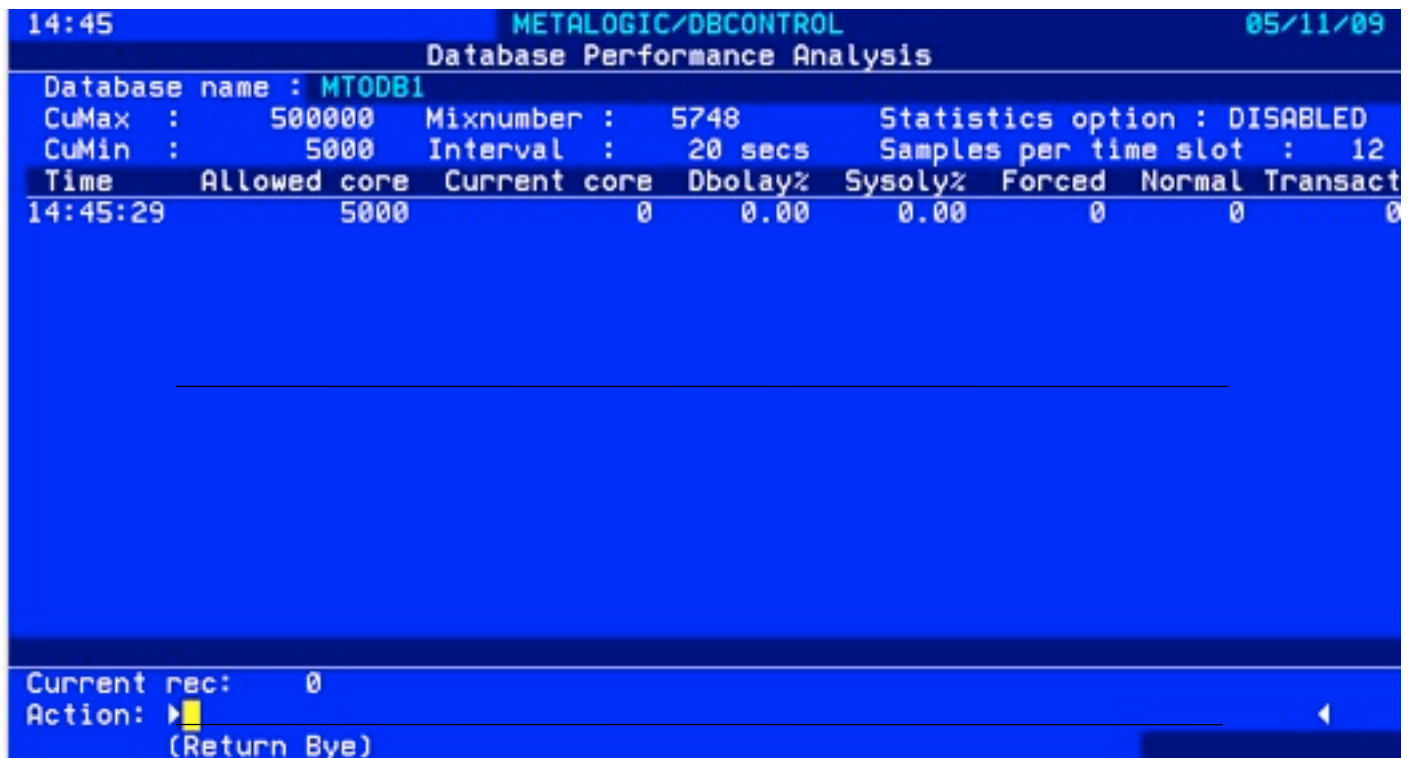
```

From this screen, you may change the DBCONTROL parameters using **CHANGE** or go directly into the real-time statistics screens by using the **STATS** command.

Input of **RETURN** will drop you back to the previous screen (as will using the specify key).

DATABASE PERFORMANCE STATISTICS screen

This screen is entered from the Runtime Parameters menu by using the **STATS** command.



| Time | Allowed core | Current core | Dbolay% | Sysoly% | Forced | Normal | Transact |
|----------|--------------|--------------|---------|---------|--------|--------|----------|
| 14:45:29 | 5000 | 0 | 0.00 | 0.00 | 0 | 0 | 0 |

Each entry you see on the screen consists of mean values for ALLOWEDCORE, the current in-use core, database overlay rate and system overlay rates, all averaged over time. A new record entry is created when the current number of samples equals the value of the **SAMPLES** parameter for the specified database; if the **STATISTICS** parameter is enabled then this current entry will also be written to the statistics file. In the figure above, the interval between samples is 20 seconds, the samples parameter is 12, so a file record will be written every 12×20 seconds = 4 minutes.

If the **STATISTICS** option is disabled then memory-only statistics are retained. Each time a new statistics entry is created, the oldest one is discarded (that is, last-in first-out). Only 15 entries are kept in memory at any time.

Note that the values for FORCED and NORMAL overlays and TRANSACTIONS are delta values i.e. they are the number of operations that were performed during the life of the relevant statistics record entry. For example, during 11:05 and 11:09, 17 forced overlays were performed for the METATAPELIB database. These figures are particularly useful for indicating the performance and activity for individual databases.

The use of the **SPECIFY** key will automatically update this screen with the latest information available otherwise the screen will be automatically updated every minute.

As with the other screens, **RETURN** takes you back to the previous menu and **BYE** will quit the monitor session.

If you are running with **STATISTICS** enabled, it is possible to "browse" the current

statistics file using '+' or '-' to move backwards or forwards through the file. Both '+' and '-' will take an optional integer and the monitor program will attempt to go forwards or backwards that number of records, where possible. If no integer is given, then a page of data (15 records) is assumed. Further, the commands 'FIRST' and 'LAST' will take the user to the beginning and the current end of the statistics file respectively.

Alternatively, if a time specification, in the format of HH:MM, is given then DBCONTROL will attempt to search the statistics file for the nearest matching time record.

CHANGE DBCONTROL RUNTIME PARAMETERS screen

This screen allows individual DBCONTROL-database parameters to be changed, as if multiple AX commands had been used.

From here, the user can rapidly configure the run-time parameters of each DBCONTROL-database combination without having to use the AX interface. By default, a SAVE will be done to preserve the changes BUT please note that the new PARAMETERS file will NOT be created until the next sample is ready to be taken. Messages will be displayed when the SAVE command is processed.

```
14:46 METALOGIC/DBCONTROL 05/11/09
Change DBCONTROL Runtime Parameters

Database name : MTODB1

Interval time between samples ..... 20 secs
Timeout before DBControl termination with no users.. 0 mins
Dbcontrol task priority ..... 50
Number of samples for each statistics record ..... 12
* DBcontrol 'COLLECTOR' mode only enabled..... RESET Set/Reset
Do not restore database parameters (NORESTORE)..... RESET Set/Reset
* Use Allowed-Core Only (ACONLY) option ..... RESET Set/Reset
Statistics option ..... RESET Set/Reset
Reporting option ..... RESET Set/Reset

Current Maximum Allowed Core (CUMAX) is ..... 500000 words
Current Minimum Allowed core (CUMIN) is ..... 5000 words

Save these new changes to the PARAMETERS file? ..... YES Yes/No

Action: ( Return Bye ) or SPECIFY will RETURN
** Changes to these parameters are not effective until next restart **
```

If you input any of the new parameters incorrectly, DBCONTROL will flag an error and position the cursor on the offending field. When your changes have been accepted, the modifications will take effect immediately and the relevant DBCONTROL task will use the new run-time parameters at the next sample time. The user is automatically returned to the Runtime Information screen upon a successful change.

You may drop out at any time by using the specify key or entering RETURN; both will take you back to the previous menu.

Statistics files

DBCONTROL statistics file which is used to hold daily performance data for the specified database in both collector and non-collector modes. This data can be uploaded to a PC for performance analysis. Files are generated on a daily basis for each individual database and are called:

***DBC/<USERCODE>/<DATABASE>/STATISTICS/<DAYNAME>**

where <DAYNAME> is SUNDAY, MONDAY etc. They will reside on the family specified by the DBC_FAMILY configuration variable, if assigned, otherwise the disk family where the relevant CONTROL file can be found. It is strongly recommended that the DBC_FAMILY assignment is used since this will ensure that all statistics files will be located in the same place.

The layout of each statistics record consists of the following fields:

<HHMM>, <ALLOWED>, <CURRENT>, <DBOLAY>, <SYSOLAY>, <NORMAL>, <FORCED>, <TRANS>, <CUMAX>, <CUMIN>

Each of the above fields has the following meaning:

| Field | Meaning |
|-----------|---|
| <HHMM> | Timestamp of statistics record I/O in hours, minutes, seconds |
| <ALLOWED> | Average ALLOWEDCORE database setting during sample period |
| <CURRENT> | Average database in-use core during sample period |
| <SYSOLAY> | Average system overlay rate used by DBCONTROL to monitor memory performance |
| <DBOLAY> | Average database overlay rate during sample period |
| <NORMAL> | Number of NORMAL overlays this database has done during sample period |
| <FORCED> | Number of FORCED overlays this database has done during sample period (not the current total) |
| <TRANS> | Number of transactions during sample period (not the current total) |
| <CUMAX> | Current CUMAX value at time of statistics file I/O |
| <CUMIN> | Current CUMIN value at time of statistics file I/O |

The files are created as normal A-Series data files with the following attributes:

FILEKIND = DATA
MAXRECSIZE = 15
BLOCKSIZE = 30
FRAME SIZE = 48

The FORCED and NORMAL overlays data are not used by DBCONTROL for controlling memory allocations, but these values are very useful for statistical

Copyright

Copyright © 1979-2019 Metalogic S.à R.L., Luxembourg.
All rights reserved.

Last revision: January 2019

METALOGIC believes that the information described in this manual is accurate and reliable, and much care has been taken in its preparation. However, no responsibility, financial or otherwise, is accepted for any consequences arising out of the use of this material.

The information contained herein is subject to change. Revisions may be used to advise of such changes and/or additions.