

# ADABAS D

## CONTROL



**Manual Order Number: ESD611-034WOU**

This **document** is applicable to ADABAS D Version 6.1.1 PE and to all subsequent releases, unless otherwise indicated in new editions or technical newsletters.

Specifications contained herein are subject to change and these changes will be reported in subsequent revisions or editions.

Readers' comments are welcomed. Comments may be addressed to the Documentation Department at the address on the back cover.

© April 14, 2025, SOFTWARE AG, Germany & SOFTWARE AG of North America, Inc.

All rights reserved

Printed in the Federal Republic of Germany

The SOFTWARE AG documentation often refers to numerous hardware and software products by their trade names. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies.

## 1 Introduction

**CONTROL** is used to control and monitor the ADABAS server, and to execute the backup and recovery procedures.

CONTROL supports the following operations:

- Installing the database server,
- Loading the system tables,
- Starting and shutting down the database server,
- Starting and shutting down the remote SQL server,
- Monitoring the database server,
- Backing up the database and log,
- Restoring the database and log,
- Expanding the disk capacities of the database server,
- Running the Diagnose tool.

Remote access to non-local database servers by CONTROL is not yet supported.



## 2 Concepts

### 2.1 SERVERDB Structure

One or more instances of ADABAS can be installed and operated on a computer. Each ADABAS instance is called a SERVERDB and has the following structure:

System Devspace	Transaction Log Devspace	Archive Log Devspaces ... ( 0..7 )	Data Devspaces ... ( 1..64 )
--------------------	-----------------------------	--	------------------------------------

The term "devspace" denotes a physical disk or part of a physical disk, e.g., a UNIX raw device or a file.

ADABAS assumes that each devspace is located on a different disk. If this is not true, decreased performance is to be expected.

The ADABAS devspaces have the following meanings:

System Devspace:                      The configuration data and the mappings of the logical page numbers to physical page addresses are administered on this devspace. The size of the system devspace therefore depends directly on the database size.

Transaction Log Devspace:            Modifications to the data are recorded in this log and written to disk at the end of the transaction. The transaction log can be used to ROLLBACK transactions and is written cyclically. Its size must be sufficient to receive the modifications of all open transactions.

**Archive Log Devspaces:** To ensure the recovery of the database contents after a media failure, all modifications made to the database contents are recorded in the archive log devspace. Using the backup functions (SAVE LOG, SAVE LOG SEGMENT), the contents of the archive log devspace are saved to tape (DAT, Video8) and the used space is then released. The size of archive log devspace must therefore be sufficient to receive all modifications that occur during two backups. The archive log can comprise several devspaces.

**Data Devspaces:** The user data (tables, indexes) and the SQL catalog (schema information) are stored in the data devspaces. As a rule, an ADABAS-internal striping algorithm evenly distributes the data belonging to a table across all data devspaces. The storage space defined by all data devspaces is the total size of the database.

The data devspaces are not directly related to the storage of database objects. An assignment of tables to data devspaces is not possible and not necessary. A table or an index can use one page (4 KB) as a minimum; or a table can use all data devspaces (i.e., the whole database) as a maximum. A table increases or decreases in size automatically without administrative intervention.

To obtain a higher degree of availability, the system devspace and all data devspaces of a SERVERDB can be mirrored. Write operations are performed on each of the two mirrored data devspaces, while read operations alternate between one mirrored data devspace and the other to distribute the I/O load.

If the data devspaces become full, database operation stops and ADABAS performs an "emergency shutdown". The devspace usage level of a SERVERDB is therefore a critical parameter of database operation and must be monitored. A SERVERDB can be expanded by additional data devspaces, if necessary, while the database is operational.

## 2.2 Logging

To meet different data protection and computer configuration requirements, ADABAS provides a gradual logging concept. The configuration parameter LOG MODE can be used to select one of the following variants:

- |         |  |
|---------|--|
| NORMAL: | This log mode is the recommended default mode. It requires at least one archive log devspace in addition to the transaction log devspace. The archive log devspace(s) must be located on disks different from all the other devspaces (system devspace, transaction log devspace, data devspaces). A minimum configuration for this log mode therefore requires at least 2 (physical) disks.   |
| DUAL:   | For a still higher degree of data protection requirements, the archive log can be doubled, requiring at least one more devspace to mirror the archive log. The minimum configuration for this log mode thus comprises at least 3 disks. This configuration has the following advantages: a failure of the transaction log devspace or of one of the archive log devspaces does not interrupt database operation, and once the defective devspace has been repaired, it can be updated while the database is operational. |

The log modes SINGLE and DEMO are not intended for production environments but for test and demonstration purposes in configurations with a small number of disks.

**SINGLE:** In this configuration, there is no archive log, only the transaction log is written. This allows ADABAS to be used in configurations with one disk. Transactions can be rolled back; but if a device failure occurs, the contents of the database are destroyed. The only course of action then is to restore a previous database state (RESTORE DATA, CLEAR LOG).

**DEMO:** In this configuration, too, there is no archive log, and only the transaction log is written. In contrast to log mode SINGLE, the transaction log is cyclically overwritten to prevent it from being filled completely. The other characteristics are the same as described for the log mode SINGLE.

If the archive log becomes full, database operation stops and ADABAS performs an "emergency shutdown". The usage level of the log is therefore a critical parameter of database operation and must be monitored. The archive log can be expanded by additional archive log devspaces, if necessary, while the database is operational.

Either the whole log or selected parts of the log can be saved. This requires the definition of the maximum size of log segments. The size of the log segments is independent of the size of the individual archive log devspaces; it must not exceed the size of the whole archive log. If a log segment is used to 70%, ADABAS tries to complete the log segment. The oldest completed log segment can be saved to an external medium (tape). Afterwards, the space used by this log segment is released. This procedure can be automatically performed whenever a log segment is filled.

If no log segment size has been defined, the whole archive log is considered a log segment.



## 2.3 Backup

ADABAS supports complete and incremental data backups providing the required restore and restart functions in order to make databases operational again after power and media failures. Periodic backups are indispensable for production database environments.

### ***Complete Backup of the DATABASE (SAVE DATA)***

A complete backup of the data of a SERVERDB (SAVE DATA) can be performed while the database is operational (and, of course, when the database is in COLD mode). For backups performed while the database is operational, the state of the database at the beginning of the backup, not that at the end of the backup, is written to an external medium (e.g., tape). A complete backup is the basis for any restore operations that might be required.

A complete backup makes sense before and after large-scale modifications to the SERVERDB and before stopping the system, e.g., at the weekend.

### ***Incremental Backup, logical (SAVE LOG, SAVE LOG SEGMENT)***

As a rule, the contents of the archive log are used for an incremental backup. All modifications made to the database are recorded in the archive log from which the data can be logically reconstructed. As an overflow of the archive log (LOG FULL) has the effect that database operation is interrupted, the log or a log segment must be backed up in time to release the used log area. It is not possible to backup the log in log mode DEMO.

### ***Saving a Log Segment***

If the archive log (SAVE LOG SEGMENT) is backed up while the database is operational (or when the database is shut down), a completed segment of the log is written to an external medium (tape). The devspace area of the saved log segment is cleared and can be used for the recording of future database modifications. In

addition, CONTROL offers an automatic backup of completed log segments (AUTOSAVE LOG).

***Backing Up the Archive Log While the Database is Operational***

In addition to saving completed log segments, the whole archive log and the currently active, incomplete log segment can be written to an external device (tape). This can be done using SAVE LOG while the database is operational. Afterwards, the whole archive log devspace of the SERVERDB is cleared.

A backup of the archive log performed while the database is not operational is only used to prepare a database recovery and should not be considered for the backup procedure in the Schedule Manager because the log is not cleared.

***Incremental Backup, physical (SAVE PAGES)***

An alternative to saving the log is backing up the database pages that have been modified since the last backup (SAVE PAGES).

***Recommendations for Backing Up the SERVERDB***

We recommend to perform a complete data backup after the installation of the database and then to schedule a backup in regular time intervals (weekly).

If the database modifications are spread across the whole database, we recommend to use the archive log for incremental backups. These should be performed regularly, at best automatically (AUTOSAVE LOG).

For large-scale database modifications that only relate to a part of the database, we recommend to backup the modified database pages. As a rule, an incremental backup by SAVE PAGES increases the data volume but speeds up the restore times because the database pages must only be reloaded; whereas for a restore with the archive log, previous logs must be reapplied.

To accelerate the backup and restore operations for large databases (SAVE DATA, SAVE PAGES, RESTORE DATA), it is recommended to use several tape devices simultaneously. The backup and restore times then only depend on the number of tape devices used and on the disk capacity of the largest data devspace.

## 2.4Restore

If a database failure other than a devspace failure occurs, the restart of ADABAS ensures that the last consistent database state is reestablished using the transaction log; this means, the effects of committed transactions are reapplied on the data devspaces, and the effects of open transactions are rolled back.

If a media failure occurs on the system devspace or a data devspace, database operation ends (unless the data devspaces and the system devspace are mirrored). After repairing the media failure, the database must be recovered starting with the last complete backup (RESTORE DATA).

If the archive log had not been saved in the meantime, the restart has the effect that the database modifications recorded in the archive log are reapplied, thus reestablishing the last consistent state of the database.

If the archive log had been saved in the meantime, RESTORE DATA must be performed and the backup of the archive log must be restored with RESTORE LOG. But before performing RESTORE LOG, it is essential to save the current contents of the archive log (SAVE LOG performed in COLD mode), because otherwise it would be overwritten by RESTORE LOG. If several log backups have been made after the complete backup, RESTORE LOG must be performed with each of these log backups. The last backup to be restored contains the current contents of the archive log, thus producing the last consistent state of the database. A subsequent RESTART then enables database operation (WARM operating mode).

If the database needs to be reset to a previous state for organizational reasons, restore the most recent complete backup (RESTORE DATA) that was made previous to the desired date and time, save the current archive log, and restore the log backups subsequent to the complete backup (RESTORE LOG). The desired database state can be determined by specifying a date and a time (RESTORE LOG UNTIL).

A similar procedure is used to restore modified database pages instead of log backups. In this case, too, there will be a sequence of tapes written after the last complete backup using SAVE PAGES. These tapes must be restored one after the

other using RESTORE PAGES. The current contents of the archive log do not need to be saved; subsequently, during the restart, they are used to reestablish the last consistent database state.

If a media failure occurs on the transaction log devspace or archive log devspace, database operation ends unless the log mode DUAL defines a mirrored archive log. After repairing the media failure and performing RESTORE LOG FROM DEVSPACE and a subsequent restart, the database is in a consistent state again.

## **2.5 Consistency Check and Optimizer Support**

To ensure a safe database operation and good performance, two other activities must be done from time to time: VERIFY and UPDATE STATISTICS.

VERIFY can be executed while the database is operational. It checks the consistency of internal chains within the B\* trees used. If inconsistencies are discovered, the database must be restored. A VERIFY is recommended before each complete backup of the database.

VERIFY in COLD mode (i.e., before a RESTART of the SERVERDB) has an additional property: pages wrongly recorded as used since an irregular end of database operation are released to the free space management.

UPDATE STATISTICS determines the number of rows in tables and the selectivity of individual columns. The ADABAS optimizer needs these specifications to determine the best strategy for the processing of complex SQL statements. If the sizes or the value assignments in the database have changed considerably, a new UPDATE STATISTICS must be performed. UPDATE STATISTICS should be executed once a week.

If ADABAS determines differences between the optimizer assumptions from the last UPDATE STATISTICS and the current state of a table, it attempts to perform an implicit UPDATE STATISTICS. If there are conflicting locks, this attempt might be aborted, so that the implicit UPDATE STATISTICS is not a complete equivalent of the explicit UPDATE STATISTICS.



## 2.6 Caches

Read and write operations to the devspaces are buffered in order to save disk accesses. The pertinent main memory structures are called caches. They can be dimensioned. ADABAS defines the following caches:

Data Cache:	This cache contains the last read- or write-accessed pages of the data devspaces. The data cache is shared by all simultaneously active users. The hit rate in the data cache is decisive for the performance.
Converter Cache:	The converter cache and its hit rate are also decisive for the performance. The converter cache contains the last read- or write-accessed pages of the system devspace. The converter cache is shared by all simultaneously active users. For the converter cache, you should strive for hit rates as close to 100% as possible.
Proc Code Cache:	This structure contains the code of the last executed DB procedures, triggers, or DB functions. The proc code cache is shared by all simultaneously active users.
Proc Data Cache:	This cache exists for each active user (or for each database session). It contains the parameters or variables belonging to the last executed DB procedures, triggers, and DB functions.
Catalog Cache:	This cache exists for each active user (or for each database session). It contains the last catalog objects used by a database session and the internal representation (application plans) of the last executed commands. Displacements from the catalog cache first move the data into the data cache.

Temp Cache: This cache exists for each active user (or for each database session). It contains the last database objects (SELECT results, temporary tables) generated or temporarily used by a database session. Applications that generate large join results or frequently work with temporary tables can improve their performance by configuring a temp cache with an appropriate size. Displacements from the temp cache first move the temporary data into the data cache.



## 2.7 Multiprocessor Configurations

For an optimal usage of multiprocessor configurations, ADABAS supports an external/internal tasking that can be configured. The aim hereby is to support as many database sessions as possible with a minimum number of operating system processes. One operating system process is required for each CPU that resides in the computer and is to be used by an ADABAS SERVERDB.

The degree of the external/internal tasking is controlled by the two configuration parameters MAXUSERTASKS and MAXCPU.

The parameter MAXUSERTASKS indicates the maximum number of simultaneously active users (database sessions). Overconfiguration exceeding the actual requirements results in increased address space (especially shared memory) requirements.

The parameter MAXCPU indicates the number of CPUs to be made available to the SERVERDB.

For example, if you want to use a maximum of 800 simultaneously active database sessions on a 4-processor computer, MAXUSERTASKS must be set to 800 and MAXCPU to 4. The SERVERDB can then utilize the 4 processors by establishing 4 operating system processes each of which performs an internal tasking for up to 200 users.

If the number of configured database sessions is exhausted, no other user can connect to the SERVERDB. The number of active sessions is therefore a critical parameter of database operation and must be monitored.

## 2.8 Client Server

To open a SERVERDB for remote SQL client operation, only the remote SQL server must be started which acts as agent for the remote clients.

To be able to use this connectivity built into ADABAS via TCP/IP sockets, the corresponding TCP/IP entries must have been previously configured. Information required for this purpose is contained in the platform-specific User manuals.

To connect to a SERVERDB, the name of the SERVERDB and the network name of the corresponding computer or network node (SERVERNODE) must be specified in addition to a valid user name/password combination. When connecting to a local SERVERDB, the SERVERNODE specification can be omitted.

## 2.9 Distribution

When installing and configuring an ADABAS SERVERDB, you must determine whether this SERVERDB is to be used autonomously (stand-alone) or as part of a distributed configuration.

After installing a stand-alone SERVERDB, other SERVERDBs can be added to form a distributed configuration. These SERVERDBs can be installed either on the same computer or on other computers. When installing new SERVERDBs in a distributed configuration, these are provided with both the current global catalog and the current state of all replicated tables of the distributed configuration. When doing so, a SERVERDB (SPONSOR SERVERDB) within the distributed configuration must be specified. The names of the SERVERDBs must be unique within a distributed configuration.

To process SQL statements in a distributed configuration, internal orders are sent to several SERVERDBs and executed there, and the result data or the return code is returned. SERVERTASKS on each SERVERDB are used for such distributed command processing. The number of these SERVERTASKs is determined by the configuration parameter MAXSERVERTASKS. If too few SERVERTASKs have been configured, wait states occur during the processing of SQL orders sent to

other SERVERDBs. It must be taken into account that SERVERTASKs are also needed for other purposes, e.g., for SAVE/RESTORE operations.

SERVERDBs in a distributed configuration are completely autonomous with regard to restart/shutdown, save/restore, etc. The effects of CREATE/ALTER/DROP statements are known on all nodes because of the concept of a global catalog. Each user is assigned to the SERVERDB at which the CREATE USER statement was performed for him. All catalog entries for his database objects (e.g., tables) are stored on his SERVERDB. All SERVERDBs of a distributed configuration know all users and their SERVERDBs. Therefore, the pertinent catalog entries must be retrieved from the corresponding SERVERDB when executing a statement that concerns database objects of a non-local user. These catalog entries are stored in the catalog cache of the corresponding database session to save network communication for the repeated execution of the same commands.

For tables defined WITH REPLICATION, not only the user data but also the catalog information concerning the replicated tables is stored redundantly on all SERVERDBs belonging to the distributed configuration. Thus, it is possible to perform SQL statements concerning the replicated tables without accessing (network communication to) catalogs located on other SERVERDBs.

Replicated tables can be modified even if single SERVERDBs have failed or are shut down. A majority principle prevents table replications in separate subnetworks from being modified in different ways. Replicated tables can only be modified as long as the majority of the SERVERDBs belonging to a distributed configuration is operational and accessible. This means that a smaller subnetwork has no right to make modifications. For an equal number of SERVERDBs, the internal numbering of SERVERDBs is decisive.

When restarting, replicated tables are automatically updated to the most recent state within the network either by redoing the modifications made during the time of failure or by completely transferring the latest table contents.

In addition to replicated tables as synchronous copies, ADABAS supports asynchronous copies in the form of snapshots. Snapshot tables typically require a distributed configuration in which data from one SERVERDB is provided on other SERVERDBs in the form of extracts, the snapshot tables. Snapshot contents are explicitly updated with the REFRESH statement. For snapshots based on source

tables, a snapshot log can be used to incrementally transfer the modifications. For snapshots with a more complicated structure (joins), the complete snapshot contents are always transferred with REFRESH.

## 2.10 Special Users

For each SERVERDB, the following special users exist:

**CONTROL USER:** This user has the right to perform all functions of CONTROL. The CONTROL user can connect several times to his SERVERDB, for example, to retrieve information about operating parameters while performing a long-time backup.

In contrast to all the other users, this user is only known on his SERVERDB in a distributed configuration. On other SERVERDBs, the same user name/password combination can therefore be used for the CONTROL user.

**SYSDBA:** This user is the only user who can define other users, especially those with DBA status. In distributed configurations, users thus defined are known within the whole network.

Another special user is the user DOMAIN who is the owner of the catalog views.

## 2.11 Availability

The availability of a SERVERDB can be increased by using the corresponding hardware, operating system, or database features.

For mission-critical applications, we recommend RAID-5 configurations for data devspaces. These configurations require about 20 to 30% more disk capacity because of their internal redundancy. A failure and the exchange of a disk, however, does not impair the database operation.

The same applies to operating system mirror disks. These, however, require double disk capacity.

Regardless of the hardware and operating system properties, ADABAS provides a mirroring of system devspace and data devspaces. (Independent of these mirrored devspaces, LOG MODE DUAL can be used to define that the archive log devspaces are mirrored.) Mirroring the system devspace and data devspaces is controlled by the configuration parameter MIRRORED and requires the definition of a corresponding number of mirrored devspaces. In a mirrored configuration, read operations alternate between the original and mirrored devspace; write operations concern both devspaces.

To avoid that a complete breakdown of a computer configuration impairs the database operation, mirrored server configurations should be built. On two different hardware systems, a SERVERDB is installed in a distributed configuration and all tables are replicated for this purpose. Consequently, both SERVERDBs have the same, transaction-consistent database state. Both SERVERDBs share the user sessions and thus the work load. If one hardware system fails, the user performs a new connect to the other SERVERDB. In this case, the aborted transaction must be repeated.

## 2.12 Configuration Parameters

### MAXSERVERDB

defines the maximum number of SERVERDBs in a configuration. If a stand-alone SERVERDB is to be configured, specify 1 here. For a distributed configuration, specify the maximum number of SERVERDBs involved.

### MAXBACKUPDEVS

Saving and restoring the database and log can be accelerated if several tape devices are used simultaneously. This parameter defines the maximum number of tape devices to be used simultaneously.

### MAXSERVERTASKS

SERVERTASKs in a distributed configuration help to process SQL statements of other SERVERDBs. SERVERTASKs in a stand-alone configuration accelerate the save and restore operations.

### MAXUSERTASKS

This parameter restricts the number of simultaneously active user sessions on this SERVERDB.

### MAXCPU

This parameter assigns the SERVERDB a number of CPUs. For multi-processor computers, the number of CPUs utilized by a database can be thus defined and restricted. For single-processor computers, specify 1.

**DATA\_CACHE\_PAGES**

This parameter defines the size of the data cache. The specification is made in 4 KB pages.

**PROC\_DATA\_PAGES**

This parameter defines the size of the proc data cache. The specification is made in 4 KB pages.

**PROC\_CODE\_PAGES**

This parameter defines the size of the proc code cache. The specification is made in 4 KB pages.

**TEMP\_CACHE\_PAGES**

This parameter defines the size of the temp cache. The specification is made in 4 KB pages.

**CATALOG\_CACHE\_PAGES**

This parameter defines the size of the catalog cache. The specification is made in 4 KB pages.

**CONV\_CACHE\_PAGES**

This parameter defines the size of the converter cache. The specification is made in 4 KB pages.

### MAXLOCKS

This parameter defines the maximum size of the lock list in which held and requested row and table locks are recorded for all users.

### RUNDIRECTORY

The protocol files of some ADABAS tools are stored in the specified directory.

### OPMSG1

To inform about exceptional situations, ADABAS displays messages. Priority 1 messages are displayed either on the specified terminal or output to the specified file.

### OPMSG2

To inform about exceptional situations, ADABAS displays messages. Priority 2 messages are displayed either on the specified terminal or output to the specified file.

### DEFAULT CODE

The internal code defined here is used to store CHAR values. For open systems, this is usually the ASCII code.

### DATE TIME FORMAT

This parameter is used to define the default representation of DATE and TIME values.



### SESSION TIMEOUT

This parameter defines the maximum time of inactivity allowed for all database sessions. The time is specified in seconds. If no SQL statement is issued within the specified time, the database session concerned is implicitly closed with ROLLBACK WORK RELEASE.

### LOCK TIMEOUT

This parameter defines the maximum time of inactivity allowed for all database sessions holding locks. The time is specified in seconds. If no SQL statement is issued within the specified time, the transaction concerned is implicitly rolled back with ROLLBACK WORK. However, this is only done if there are other users waiting for the lock to be released.

### REQUEST TIMEOUT

This parameter restricts the waiting time for the release of locks for all database sessions. The time is specified in seconds. If a lock request cannot be satisfied within the time thus defined, a message is returned to the waiting database session.

### LOG MODE

Here, you enter the log mode selected for the SERVERDB.

### LOG SEGMENT SIZE

Here, you define the size of a log segment. The specification is made in 4 KB pages.

### NO OF ARCHIVE LOG DEVSPACES

Here, you enter the number of archive log devspaces.



### NO OF DATA DEVSPACES

Here, you enter the number of data devspaces.

### MIRRORED

Here, you determine whether or not the system devspace and the data devspaces are to be mirrored.

### DISTRIBUTED CONFIGURATION

This parameter defines whether or not the SERVERDB is to become part of a distributed configuration.

### FIRST SERVERDB

This parameter defines a distributed configuration which at first consists of only one SERVERDB; or it defines a stand-alone SERVERDB to be the first SERVERDB of a distributed configuration to be built. If the SERVERDB concerned is not the first SERVERDB in the distributed configuration, this means that 'N' was specified, a screen is displayed when scrolling down where the SPONSOR SERVERDB must be entered.

### DEVSPACES

The type (raw device or file), the size (in 4 KB pages), and a path name are specified for any devspace required for the configuration. If 0 is specified as the size of raw devices, the size of the devspace is implicitly determined.

## 3Call and Installation

### 3.1Calling CONTROL

CONTROL can be called from the operating system level (UNIX, Windows NT, OS/2) using the following command:

**xcontrol -d** *<serverdb name>* **-u** *<controluser name>*,*<password>*

If CONTROL has not yet been installed for the specified server database, the installation screen appears. If no parameters have been specified, the connect screen appears.

In the connect screen, the CONTROL user identification, the CONTROL user password, and the name of the SERVERDB must be entered.

### 3.2 Installing a New SERVERDB

If a non-existent SERVERDB is specified for the call of CONTROL (i.e., no RUNDIRECTORY is defined), the following screen is displayed:

```
Install Serverdb <serverdb> on <dbnode>
```

---

CONTROL USER NAME...:	PASSWORD...:
SYSDBA NAME.....:	PASSWORD...:
DOMAIN USER NAME...: DOMAIN	PASSWORD...:

Serverdb does not exist  
If you want to create a new serverdb fill this form and press "Ok"

*Fig.: Installation Screen 1*

SERVERDB name and SERVERNODE are preset to the predefined data. CONTROL supports identifications and passwords with a maximum length of 18 characters. The password must be entered twice. When the specifications are complete, the next screen is displayed showing the default parameters. In a special window, a description is output for the parameter on which the cursor is placed.

Install Serverdb <serverdb> on <dbnode>

MAXSERVERDB	1
MAXBACKUPDEVS	2
MAXSERVERTASKS	4
MAXUSERTASKS	50
MAXCPU	1
DATA_CACHE_PAGES	200
PROC_DATA_PAGES	130
PROC_CODE_PAGES	76
TEMP_CACHE_PAGES	30
CATALOG_CACHE_PAGS	816
CONV_CACHE_PAGES	100
MAXLOCKS	1500
RUNDIRECTORY	/u/rel61/usr/wrk/DBDEMO

Number of SERVERDBs in a distributed database

Next Prev Explain Print Cancel

*Fig.: Installation Screen 2*

Install Serverdb <serverdb> on <dbnode>

OPMSG1	/dev/syscon
OPMSG2	sqlbdbdiag
DEFAULT CODE	ASCII
DATE TIME FORMAT	INTERNAL

Name of the destination to which priority 1 message will be sent

Next Prev Explain Print Restore Cancel

*Fig.: Installation Screen 3*

**EXPLAIN** can be used to display the computation formula of the numeric parameters and their dependencies of the other parameters.

Clicking on the RESTORE menu function copies the installation from a tape containing the backup of another SERVERDB (INIT CONFIG RESTORE).

In the next screen, the timeout values, the LOG, and the DEVSPACES must be specified. Also, it must be defined whether the SERVERDB is to be installed as a remote SERVERDB; i.e., whether it will operate with other SERVERDBs in a distributed database configuration.

Install Serverdb <serverdb> on <dbnode>

---

TIMEOUTS -----	
SESSION	900
LOCK	360
REQUEST	180
DEVSPACES -----	
LOG MODE	NORMAL
LOG SEGMENT SIZE	1500
NO OF ARCHIVE LOGS	1
NO OF DATADEVSPACES	2
MIRRORED	(Y/N) N
DISTRIBUTION -----	
DISTRIBUTED CONFIGURATION	(Y/N) Y
FIRST SERVERDB	(Y/N) N

ROLLBACK RELEASE when the time between two SQL commands is more than the SESSION TIMEOUT ( 30 sec - 32400 sec or 0 = OFF )

Next
Prev
Print
Cancel

*Fig.: Installation Screen 4*

If "Y" is entered for **DISTRIBUTION** and "N" for **FIRST SERVERDB**, at least one other SERVERDB has been started that is to run in the distributed database together with the SERVERDB to be installed now. To establish the connection, fill in the following screen:

```
Install Serverdb <serverdb> on <dbnode>

DISTRIBUTION  -----
SPONSOR SERVERNODE.....:
SPONSOR SERVERDB.NAME....:
SPONSOR SYSDBA NAME.....:
SPONSOR SYSDBA PASSWORD...:

Nodename of a running SERVERDB

Next Prev Print Cancel
```

*Fig.: Installation Screen 5*

As SPONSOR SERVERDB, any SERVERDB can be selected from the set of already installed, distributed SERVERDBs. It must be ensured that the parameter MAXDISTRIBSERVER is set to a value at least corresponding to the maximum number of distributed SERVERDBs.



Depending on the specification of the number of DEVSPACES, the following screen is initialized with the corresponding number of lines. If **MIRRORED** was marked with Y, double the number of lines appears for the system and data devspaces. In total, 64 DATADEVSPACES and 8 ARCHIVE LOGs are supported.

An R in the column TYPE indicates a raw device, an F indicates a file, and an L indicates a symbolic link.

For raw devices with the size specification '0', the total size of the device is automatically determined.

Install Serverdb <serverdb> on <dbnode>

NAME	TYPE	SIZE	DEVSPACE PATH
SYSTEMDEV	F	-	/u/dev/SYS1
TRANS LOG	R	3000	/dev/log0DB1
ARCHLOG 1	R	3000	/dev/log1DB1
DATDEV 01	R	50000	/dev/dat01DB1
DATDEV 02	R	50000	/dev/dat02DB1

Ok

Prev

Print

Cancel

*Fig.: Installation Screen 6*

When this screen has been filled completely and confirmed with "Ok", the actual installation begins.

The progress of the installation can be seen from the position of the arrow and the status message ACTIVE. If an installation step was terminated successfully, the status 'OK' is displayed and the next action becomes ACTIVE.

```
Install Serverdb <serverdb> on <dbnode>

---> INSTALL PARAMETERS..... ACTIVE
      START SERVERDB COLD..... --
      FORMAT DEVSPACES..... --
      START SERVERDB WARM..... --
      LOAD SYSTEM TABLES..... --
```

*Fig.: Status Screen 1*

After installing the SERVERDB kernel parameters, the minimum values required for the configuration parameters of the operating system kernel are displayed, so you can check the settings. Adapt the operating system kernel to these requirements, if necessary, and then resume work with INIT CONFIG.

Example under UNIX:

INSTALL PARAMETERS Protocol

NPROC39

NREGION117

NCALL39

MAXUP39

MSGMNI1

SEMMNI27

SEMMAP27

SEMMNS27

SHMMNI7

SHMSEG7

SHMMAX4083974

SHMALL4157702

Minimum size of real memory needed for the database kernel  
to prevent swapping or paging. Memory needs of the OS  
kernel or running applications are not considered.

RAMSIZE\_MB16

Ok

End

*Fig.: Parameter Check Screen*

If the status changes to ERROR, an error occurred. Select the "Protocol" button to display the installation log file. CANCEL can be used to return to the installation screen 1. Use **Next** and **Prev** to alternate between the installation screens to adapt the parameters to the error situation.

If the installation was aborted completely, it can be repeated at a later point in time. For the next call of CONTROL, the message 'INST NOT COMPLETE' is displayed in the main screen. 'Configuration/Init Config' can be used to start a new installation for which the values previously entered are displayed.

When the installation has reached the point "LOAD SYSTEM TABLES" in the status screen 1, the following screen is displayed to output more detailed information about the installation procedure:

```
Load System Tables for Complete Installation

---> Create general systemtables..... OK
Load messages and help infos..... ACTIVE
Load SET defaults..... --
Load system tables for precompilers..... --
Load system tables for QUERY..... --
Load system tables for EASY..... --
Load system tables for SQL-PL..... --
Load SQL-PL WORKBENCH..... --
Load system tables for QueryPlus..... --
Create system views..... --
Create ODBC tables..... --
Load data dictionary META DATA..... --
Load system DB-PROCEDURES..... --
```

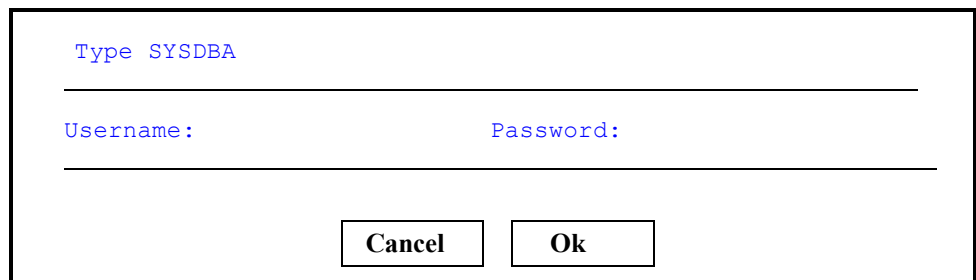
*Fig.: Status Screen 2*

If an error occurs, the installation is aborted and the status ERROR appears behind the action just performed. Select the "Protocol" button to display the LOAD log file.

If all actions were performed free of errors, all lines end with "OK" and the display changes to the CONTROL main screen.

### 3.3 The Additional Installation of CONTROL

If CONTROL is called for the first time on a SERVERDB that was installed without CONTROL, the Systemdba is requested first because he holds the system tables. For the additional installation of CONTROL, the existing local SERVERDB must be in WARM mode.

A screenshot of a graphical user interface window titled "Input Screen for Systemdba Definition". The window has a black border and contains the following elements: a label "Type SYSDBA" in blue text above a horizontal input line; a label "Username:" in blue text above a horizontal input line; a label "Password:" in blue text above a horizontal input line; and two buttons at the bottom, "Cancel" and "Ok", each in a rectangular box.

*Fig.: Input Screen for Systemdba Definition*

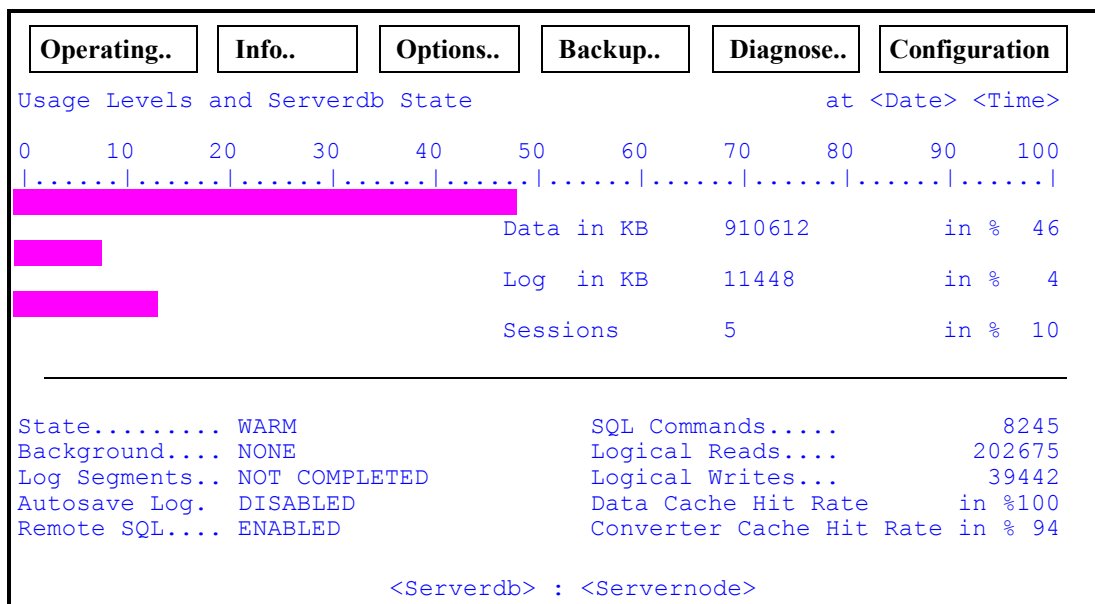
After acknowledging the screen, the CONTROL system files and system tables are created, and the main screen of CONTROL is displayed.

**WARNING !**

If you leave the screen with , the systemdba is unknown to CONTROL and the system tables cannot be created.

### 3.4 The Main Screen

After a successful connect, the main screen is displayed showing the most important round-the-clock system activities of the ADABAS server:



*Fig.: Main Screen of CONTROL*

The data fields in the main screen have the following meanings:

**<Serverdb>:**

Name of the server database.

**<Servernode>:**

Name of the computer within the network where the database server is installed.

**<Date> <Time>:**

Date and time at which the displayed data was retrieved.

***Database Usage Levels:***

describes the current utilization level of the database. This measured value informs the operator whether the physical disk space must be expanded soon. The ADABAS server performs an *"emergency shutdown"*; i.e., it shuts down automatically when the usage level of the database is 100%.

***Log Usage Levels:***

describes the current utilization level of the log. The database server shuts down (*"emergency shutdown"*) when the usage level of the log is 100%.

The operator should therefore observe the log usage in order to start a manual backup of the log or of a log segment, if this should become necessary.

After a save of a log segment, the utilization level of the log will be decreased accordingly.

***Sessions Usage Level:***

describes the percentage use of the configured database server connections. When the maximum value is reached, no more connections can be established to the database server. Application programs attempting to connect to the database server receive a corresponding error message. This maximum value can be decreased or increased using the **Configuration Parameters** menu function, by modifying the system parameter MAXUSERTASKS.

***Serverdb State:***

describes the current operating mode of the database server. The following table shows all operating modes of the ADABAS server and their meanings:

Operating Mode	Meaning
OFFLINE	The ADABAS server is not running. The database server kernel has not been started yet. Usually, the operator starts the database system directly in warm mode using the <b>Operating Restart</b> menu function.
COLD	The ADABAS server has been started successfully. All system parameters which were changed using the <b>Configuration Parameters</b> menu function are effective. Database activities are <u>not</u> possible. Only in this mode, some maintenance activities such as restoring the database or log, or modifying certain system parameters can be performed.
WARM	This is the normal operating mode in which users can work with the ADABAS server.

***Background:***

shows whether a backup, a verify devspaces, or an update statistics is active in the background.

***Log Segments:***

shows whether a log segment is full and thus ready for saving.

***Autosave Log:***

shows whether the automatic log backup has been enabled.

***Remote SQL:***

shows whether the remote SQL server has been started, thus allowing users to access the server database from other computers.



***SQL-Commands:***

shows the number of SQL statements issued since the last counter reset or start of CONTROL.

***Logical Reads:***

shows the number of read accesses to the data cache performed since the last counter reset or start of CONTROL.

***Logical Writes:***

shows the number of write accesses to the data cache performed since the last counter reset or start of CONTROL.

***Data Cache Hit Rate:***

shows the percentage hit rate of accesses to the data cache relative to the physical I/O.

***Converter Cache Hit Rate:***

shows the percentage hit rate of accesses to the converter cache. The converter cache hit rate should be as close to 100% as possible. If necessary, the system parameter CONV\_CACHE\_PAGES can be increased using the **Configuration Parameters** menu function.

### 3.5CONTROL Menu Structure and Help Texts

<div> <div>Operating..</div> <div>Info..</div> <div>Options..</div> <div>Backup..</div> <div>Diagnose..</div> <div>Configuration</div> </div>					
Refresh..	Activity	Reset Counters	Save..	Op Messages	Alter Parameter..
Restart..	Configuration	Remote SQL Server..	Restore..	Log Messages	Alter Config..
Shutdown..	Users	Accounting..	Show History	Inst Protocol	Load Systables
Update Statistics..	Caches	Access Mode..	Show Protocol		Init Config
Exit F3	I/O Accesses	Kernel Trace..	Media Manager		Clear Serverdb
	Locks	Autosave Log..	Label Manager		
	Log	Schedule..	Schedule Manager		
	Processes				
	Regions				
	Memory				
	Version				
	Help F1				

The menu bar can be activated in different ways.

- Each menu can be selected directly by pressing the highlighted letter and the **Control** key at the same time.
- **F12** can be used to address the menu bar. To reach the adjacent pulldown menu, use the **left**/**right** cursor keys.

A function of a menu item is activated either by positioning the cursor and pressing the **Enter** key or by selecting the highlighted letter.

If a help function is available, it can be called using the **F1** key. A help screen or a selection of values is displayed. One of the displayed values can be selected. To obtain help on further subjects from within the help screens, position

the cursor on the corresponding catchword and press **F1** . **F3** , **End** returns to the previous screen.

In the main screen, the function keys are set to the following functions:

<b>F1</b>		- <b>Help</b>
<b>F3</b>	<b>End</b>	- <b>Cancel</b>
<b>F5</b>	<b>Enter</b>	- <b>Ok</b>
<b>F9</b>		- <b>Refresh</b>

In the installation screens, the function keys are set to the following functions:

<b>F3</b>	<b>End</b>	- <b>Cancel</b>
<b>F4</b>		- <b>Print</b>
<b>F5</b>	<b>Enter</b>	- <b>Ok</b>
<b>F7</b>	<b>Pgup</b>	- <b>Prev</b>
<b>F8</b>	<b>Pgdn</b>	- <b>Next</b>

In the info screens, the function keys are set to the following functions:

<b>F2</b>		- <b>In Pages / In KB</b>
<b>F3</b>	<b>End</b>	- <b>End</b>
<b>F4</b>		- <b>Print</b>
<b>F5</b>	<b>Enter</b>	- <b>Ok</b>
<b>F6</b>		- <b>Edit</b>
<b>F7</b>	<b>Pgup</b>	- <b>Prev</b>
<b>F8</b>	<b>Pgdn</b>	- <b>Next</b>
<b>F9</b>		- <b>Refresh</b>

In the Schedule Manager, the function keys are set to the following functions:

<b>F1</b>	
<b>F2</b>	
<b>F3</b>	, <b>End</b>

- **Help**
- **Reset**
- **End, Quit, Cancel**  
If values have been modified, a warning is displayed.
- **Insert**
- **Ok, Update, Confirm**
- **Delete**
- **Prev, Scroll Up**
- **Next, Scroll Down**
- **Search**

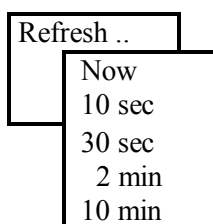
<b>F4</b>	
<b>F5</b>	, <b>Enter</b>
<b>F6</b>	
<b>F7</b>	, <b>Pgup</b>
<b>F8</b>	, <b>Pgdn</b>
<b>F9</b>	



## 4Operating Menu Function

The **Operating** menu offers the following functions: refreshing the contents of the screen, starting and shutting down the database server, performing update statistics, and exiting CONTROL.

### 4.1Operating / Refresh

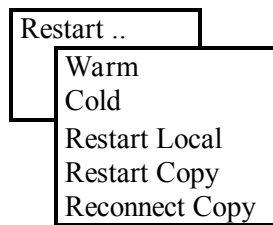


This menu function updates the values of the main screen.

An automatic refresh can be started at intervals of 10 sec, 30 sec, 2 min, or 10 min.

The function can be cancelled with  .

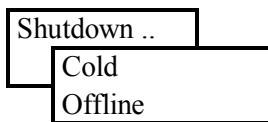
## 4.2 Operating / Restart



The **Restart** menu function starts the database server from OFFLINE or COLD operating mode into WARM mode. If the WARM operating mode is not reached after activating the **Restart** menu function, the possible error cause can be determined by using the **Diagnose Operating Messages** menu function.

If the SERVERDB belongs to a distributed database, the **Restart Local**, **Restart Copy**, and **Reconnect Copy** functions are provided in addition, to connect the SERVERDB to a distributed database.

### 4.3 Operating / Shutdown



The **Shutdown** menu functions switch the database server from WARM into COLD mode. This is displayed as OFFLINE mode.

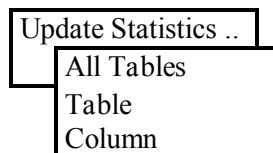
If 'exit' is used in COLD mode to leave CONTROL, a window is displayed providing the 'STOP' button. This button can be used to completely shut down the database server, before terminating CONTROL.

The **Shutdown Cold** menu function is only provided in WARM mode to switch the ADABAS server into COLD mode. A screen is displayed which allows the user to cancel the function, perform a normal **Shutdown**, or perform a **Shutdown Quick** that aborts any existing connections and transactions.

The **Shutdown Offline** menu function shuts down the ADABAS server. In WARM mode, a screen is displayed which allows the user to cancel the function, perform a normal **Shutdown**, or perform a **Shutdown Quick** that aborts any existing connections and transactions before the shutdown. Afterwards, the ADABAS server is in OFFLINE mode.



## 4.4 Operating / Update Statistics



This function determines the number of entries, the size of tables and indexes, and the value distribution of indexes or columns; it stores this information in the catalog.

The ADABAS optimizer needs the data stored in the catalog for optimizing the processing of complex SQL statements. If the sizes or value assignments have considerably changed in the database, a new UPDATE STATISTICS is required. We recommend to perform an UPDATE STATISTICS once a week.

If ADABAS determines differences between the optimizer assumptions from the last UPDATE STATISTICS and the current state of a table, it attempts to perform an implicit UPDATE STATISTICS. If there are conflicting locks, this attempt might be aborted, so that the explicit UPDATE STATISTICS is not replaced completely.

The operation can be applied immediately to particular tables, particular columns or to all base tables of the database server. In the Schedule Manager, an UPDATE STATISTICS is always performed on the whole SERVERDB.

## 4.5 Operating / Exit

This function is used to leave CONTROL. If the database server is in COLD operating mode, the database can be switched OFFLINE by selecting the "STOP" button. All active database processes are closed.

## 5Info Menu Function

The functions of the **Info** menu can be used to retrieve information about the caches, logical and physical I/O, locking states of database objects, database connections, the state of the log and of the logging processes, the states of database processes, conflicting shared memory segments, main memory usage, the command or system profile as well as the version identification of ADABAS.

In the display screen, it is possible to page down using the **Next** menu function and to page up using the **Previous** menu function. The **End** menu function closes the display screen, returning to the main screen of CONTROL.

In some info screens, it is possible to display the sizes in the corresponding format by using the **In Pages** or **In KB** menu function. This setting is then also valid for the main screen until it is reset by activating the other menu function.

To interpret this information, an understanding of parts of the ADABAS implementation is required. Such an understanding may be obtained from the training program for ADABAS Database Administration. A detailed interpretation of these outputs is usually only needed for support purposes.

## 5.1Info / Activity

This menu function displays the most important events and activities of the ADABAS server, showing summarized information about the states of the data buffers, the number and type of the commands processed by the ADABAS server, the applied search strategies, the state of lock synchronization, and the activities of the logging process.

Example:

Cache activity	Size KB	Accesses	Hits	Hit rate %
Data.....	800	349	349	100
Catalog.....	3264	130	104	80
Converter.....	987	421	418	99
Temporary.....	120	0	0	0
Command activity				
SQL commands.....	32	Creates.....		4
Rollbacks.....	4	Alters.....		0
Commits.....	3	Drops.....		0
Prepares.....	11			
Executes.....	22	Catalog scans.....		7

In the first section, the display contains the following information about the ADABAS caches:

- Total number of accesses, successful and unsuccessful accesses to the ADABAS data cache as well as the percentage of successful accesses  
(Data)

The accesses to the buffers (caches) are shown in the same way for

- the *Catalog*
- the temporary data cache (*Temporary*)
- the converter

The hit rate for the cache areas indicates whether the configured size of these areas is sufficient. The data cache hit rate is especially important. It should be as close to 100% as possible. Values under 70% are insufficient and a sign of bad performance.

In the following, information about the processed SQL statements is shown:

- Number of SQL statements  
(*SQL Commands*)
- Number of ROLLBACKs  
(*Rollbacks*)
- Number of COMMITs  
(*Commits*)

From the number of COMMITs and ROLLBACKs results the number of transactions.

- Number of dynamic SQL statements  
(*Prepares*)
- Number of executions of dynamic SQL statements  
(*Executes*)
- Number of SQL statements for the creation of database objects  
(*Creates*)
- Number of SQL statements for the alteration of database objects  
(*Alters*)
- Number of SQL statements for the dropping of database objects  
(*Drops*)
- Number of sequential read commands through the catalog  
(*Catalog scans*)

The next section shows the I/O activities:

I/O activity			
-----			
Physical reads.....	0	Logical reads.....	344
Physical writes.....	0	Logical writes.....	173
Locking activity			
-----			
Entries available...	5100	Row locks.....	4
Max. used...	4	Table locks.....	7
Avg. used...	0		
Lockholder.....	0	Collisions.....	0
Lockrequester.....	0	Escalations.....	0
Logging activity			
-----			
Logpages written....	0	Group commits.....	0
Waits for logwriter.	0	Log queue overflows....	0
Scan and sort activity			
-----			
Table scans.....	9	Cache sorts.....	9
Index scans.....	0	Row sorts.....	9

- Number of physical reads  
(*Physical reads*)
- Number of physical writes  
(*Physical writes*)
- Number of logical reads  
(*Logical reads*)
- Number of logical writes  
(*Logical writes*)

The following information about the locking activities is output:

- Maximum number of lock entries available  
(*Entries available*)

The maximum number of available lock entries can be specified using the MAXLOCKS system parameter of the **Configuration Parameters** menu function.

- Maximum number of entries used in the lock list  
(*Entries, Max. Used*)
- Average number of entries used in the lock list  
(*Entries, Avg. Used*)
- Number of active lockholders  
(*Lockholder*)
- Number of current lock requests  
(*Lockrequester*)
- Total number of row locks set  
(*Row locks*)
- Number of table locks set  
(*Table locks*)
- Number of lock collisions  
(*Collisions*)
- Number of lock escalations  
(*Escalations*)

The number of lock collisions and lock escalations in relation to the total number of locks shows whether the chosen number of configured lock entries is sufficient for the database tasks.

In the next section, some essential information about the logging activities is output:

- Number of log pages written  
(*Logpages written*)

The number of log pages written is an expression of the modifying activities on the database. In normal database work, the log devspace is the most I/O intensive area.

- Number of wait states for log write operations  
(*Waits for logwriter*)
- Number of group commits  
(*Group commits*)

If a transaction is completed, then, before writing the actual commit, a check is made as to whether other transactions have reached a commit as well. If so, the writing of the commits for the transactions is done in a single I/O operation.

- Number of log queue overflows  
(*Log queue overflows*)

In the last section, statistics about successful scan and sort activities are output:

- Number of sequential scans through the whole base table  
(*Table Scans*)

Sequential scans are the slowest search operations. Too frequent scanning can be a sign of poor database design. The creation of additional indexes on the respective tables could be the remedy.

- Number of scans for which an index was read sequentially without accessing base table rows  
(*Index Scans*)
- Number of sorts within the cache  
(*Cache sorts*)
- Number of sorts within rows after an insert

*(Row sorts)*



## 5.2Info / Configuration

This menu function produces both the current values of the database management system configuration and the current usage of storage space.

Example:

### SERVERDB Status

Database mode READ/WRITE is.....	ON
Accounting is.....	OFF
Monitoring is.....	ON
Max. database users.....	10
Max. server databases for distribution	3
Max. distribution tasks for SERVERDB..	6
Size of data cache .....in KB	6000
Size of procedure code cache.....in KB	80
Size of procedure data cache.....in KB	70
Size of temp cache .....in KB	55
Size of catalog cache.....in KB	212
Size of converter cache.....in KB	40
Max lock entries.....	1200
Vtrace .....	DEFAULT
Termchar set..... IBM437_GER	ASCII

### Database Configuration

Default code.....	ASCII
Date/Time format.....	INTERNAL
Session timeout.....	900
Lock timeout.....	360
Request timeout.....	180
Log mode.....	NORMAL
Log segment size.....	20000
No of Archive Logs.....	1
No of Data Devspaces.....	1
Mirrored Devspaces.....	NO

```

SYS DEVSPACE      : /u/sys
-----
    Installed size .....in KB                      14724
TRANSACTION LOG   : /u/tra
-----
    Installed size .....in KB                      120000
ARCHIVE LOG 1     : /u/arcl
-----
    Installed size .....in KB                      120000
DATA DEVSPACE 1   : /u/dat1
-----
    Installed size .....in KB                      1953736
    Available size .....in KB                      1953736
    Temporary extension size .....in KB              0
    Used space .....in KB                          1698612
    Used space ..... in %                           87
    Used temporary space .....in KB                  12000
    Used temporary space ..... in %                   1
    Free space .....in Kb                           243124
    Free space ..... in %                            12

```

The displayed values are composed of the parameters for the data devspaces, the log devspaces, the cache areas and states. Should modifications to these values become necessary, these can be made using the **Configuration Parameters** menu function.

### 5.3Info / Users

This menu function shows all database users who are currently connected to a database.

Example:

USERID	TERMID
MAYER	hst1tys3
SCHULZ1	hst1tys5
DEMO	hst3tys1
MAYER	hst1tys2
TEST02	hst2tys5

The following information is output:

USERID            the name of the connected user.

TERMID           a terminal identification which depends on the particular operating system.

## 5.4Info / Caches

This menu function shows information about the effectiveness of the ADABAS server's data buffering (caches). The ADABAS server keeps statistics on the current number of both physical and buffered read and write accesses.

Example:

Cache	Accesses	Successful	Failed	Hit rate%
Data.....	16299	15946	298	98
File Directory.....	1755	1522	233	84
FBM Cache.....	6	6	0	100
Converter.....	22	22	0	100
USM Cache.....	0	0	0	0
Log.....	8	8	0	100
Catalog.....	5941	4584	1357	77
Temporary.....	0	0	0	0

The display contains the following information about the ADABAS caches:

- Total number of accesses (*Accesses*) as well as number of successful (*Satisfied*) and unsuccessful (*Failed*) accesses to the ADABAS data cache
- Percentage of successful accesses to the database data cache (*Hit rate %*)

Analogous to the accesses to the ADABAS data cache, the values for the following caches are displayed:

- the file directory cache      *(File directory)*

The database system uses this area for internal organization. For example, the page addresses of the roots of the individual data trees are administered there.

- the FBM cache      *(FBM Cache)*

This area is used for the management of free disk blocks.

- the converter cache      *(Converter Cache)*

- the USM cache      *(USM Cache)*

This area is used for the User Storage Management.

- the log cache      *(Log)*

During rollback, the log cache helps to accelerate the process.

- the dictionary cache      *(Catalog)*

Information from the data dictionary of the database provided in the dictionary cache is available fast because no accesses to the disk are required. The hit rate for this area should therefore be more than 80%.

- number of accesses to the temporary data cache      *(Temporary)*

Temporary result sets are managed in the temporary cache. Such result sets are created, e.g., during the execution of a SELECT statement with ORDER BY clause.

## 5.5 Info / I/O Accesses

This menu function shows statistics on the number of physical and logical accesses to the different devspaces of an ADABAS database.

Example:

Datapages	Logical I/O		Physical I/O	
	Reads	Writes	Reads	Writes
Catalog.....	256	0	1	0
Permanent.....	1302	58	5	33
Temporary.....	3907	1950	0	0
Leaf.....	4079	1941	6	32
Level 1.....	1344	67	0	1
Level 2.....	38	0	0	0
Level 3.....	4	0	0	0
* Summary *	5465	2008	6	33

For each database devspace, the number of logical and physical read and write accesses is displayed:

**Catalog** denotes the area used by the data dictionary of the database system. Frequent writing to this area is a sign of modifications made to the database design.

**Permanent** is the actual data devspace for permanent data.

**Temporary** is a data devspace on the disk used temporarily. It is required, e.g., for the generation of selected datasets.

**Leaf, Level 1, Level 2, Level 3** display information about the structure of the data trees. ADABAS organizes the storage of data in form of B\* trees.

## 5.6Info / Locks

This menu function informs about the current locks and lock requests of the ADABAS server.

Example:

```

Lock List Statistics -----
Avg. Entries.....          7 Max. Entries.....        196
Collisions.....           0 Escalations.....          0
Row Locks.....          826 Table Locks.....          261

Lock State -----
DBADM.SUPPLIERS   9 00FFFE00000000000001
  LOCK ROW excl (360s) 00.00000058.000008 DBADM   qlcptys4 012

```

The upper part of the screen shows information about the average number of locks held and requested, the maximum number of locks held and requested, the number of collisions and escalations that have occurred, as well as the number of current row or table locks.

In the lower part of the screen, all current locks and lock requests are described in detail.

In addition to the table and row concerned, the kind of lock, the lock holder, and the lock holder's terminal are shown.

Share locks protect used data against modifications while being accessed. Other users can only read-access data thus locked.

The exclusive locks displayed prevent other users from accessing the same data. Not even read-access is possible. If exclusive locks are held for a longer time, they are critical for the performance of the system. In this context, LOCK REQUEST TIMEOUTs and LOCK TIMEOUTs are interesting. These configured time

parameters (see section 5.2 "Info / Configuration") serve to solve blocking and deadlock situations.

LOCK REQUEST TIMEOUT indicates the time a process may wait for the setting of a requested lock before it is aborted.

LOCK INACTIVITY TIMEOUT indicates the time a process may hold locks **without** activities, before it will be rolled back automatically (ROLLBACK).

## 5.7 Info / Logs

This menu function displays the state and activity of the logging process.

The following information is output:

- Maximum storage space available on the log devspace in KB  
(*Max. size in KB*)
- Size of a segment in KB  
(*Segment size in KB*)

Segmentation of the log devspace is especially advantageous for the log mode AUTOSAVE. Completed log segments are automatically saved while the database is operational. Afterwards, the saved segment is available again for the logging mechanism.

- Size of space reserved on the log devspace  
(*Reserved size in KB*)
- Mode in which the log is operated  
(*Mode*)
- Size of used space  
(*Used size in KB*)
- Percentage of the devspace used  
(*Used in %*)

The utilization level of the log devspace should be monitored carefully. If the utilization level reaches 100%, the database is shut down automatically. Work



can only be continued after the log devspace has been saved and cleared. If the utilization level exceeds 60%, messages are output to the diagnostic file of the database.

- Space available in KB  
(Free size in KB)
- Percentage of the devspace unused  
(Free size in %)

Example:

Log -----			
Max. size in KB.....	159988	Reserved size in KB....	4
Segment size in KB.....	40000	Log mode.....	NORMAL
Used size in KB.....	17316	Used size in %.....	11
Free size in KB.....	142672	Free size in %.....	89
Since database restart -----			
Physical writes.....	41	Segments completed.....	0
Physical reads.....	0		
Since last save of log -----			
Savepoints written.....	31	Checkpoints written....	1
KB per savepoint.....	558	KB per checkpoint.....	17312
Queue -----			
Queue size in KB.....	20	Group commits.....	0
Queue allocated in KB..	8	Waits for logwriter....	38
Queue entries.....	41	Max. waits.....	1
Queue overflows.....	0	Avg. waits.....	1

The next section displays the write and read activities (*physical writes, physical reads*), as well as the number of completed segments (*Segment completed*) in the log devspace since the last restart of the database system.

The next section displays statistical information about save and checkpoint activities:

- Number of SAVEPOINTS written  
*(Savepoints)*
- Number of CHECKPOINTS written  
*(Checkpoints)*
- Average SAVEPOINT distance, measured in KB  
*(KB per savepoint)*
- Average CHECKPOINT distance, measured in KB  
*(KB per checkpoint)*

The next section displays information about the log queue before the logging process:

- Current size of the log queue in KB  
*(Queue size in KB)*
- Number of group commits  
*(Group commits)*
- Devspace currently allocated in KB  
*(Queue allocated in KB)*
- Number of wait states for log write operations  
*(Waits for logwriter)*
- Maximum size of log queue in KB  
*(Queue entries)*
- Maximum number of wait states per log page  
*(Max. waits)*
- Number of log queue overflows  
*(Overflows)*
- Average number of wait states per log page  
*(Avg. waits)*

## 5.8Info / Processes

This menu function shows the states of all database processes that are currently active.

Example (of UNIX):

ID pid	UNIX pid	TYPE pid	APPL	State	Timeout	Region cnt idx	Wait sec	UKPsleep
T1	7888	Timer	-1	Vsleep	0	0 0	0	2670294 (s)
T2	7888	Logwr1	-1	Vsuspend	0	0 0	0	2670294 (s)
T3	7888	Logwr2	-1	Vsuspend	0	0 0	0	2670294 (s)
T4	7887	Bufwr.	-1	Vsuspend	0	0 0	0	590 (s)
T5	7888	Bufrd.	-1	Vsleep	0	0 0	0	2670294 (s)
T6	7888	Sender	-1	Vsuspend	0	0 0	0	2670294 (s)
T7	7888	Receiv.	-1	Vsuspend	0	0 0	0	2670294 (s)
T9	7888	Server	-1	Vsuspend	0	0 0	0	2670294 (s)
T10	7888	Server	-1	Vsuspend	0	0 0	0	2670294 (s)
T11	7888	Server	-1	Vsuspend	0	0 0	0	2670294 (s)
T12	7888	Server	-1	Vsuspend	0	0 0	0	2670294 (s)
T13	7888	Server	-1	Vsuspend	0	0 0	0	2670294 (s)
T14	7888	Server	-1	Vsuspend	0	0 0	0	2670294 (s)
T15	7888	User	26798	Command wait	-1	0 0	0	2670294 (s)
T16	7888	User	26794	Command wait	-1	0 0	0	2670294 (s)
T17	7888	User	26799	Command wait	-1	0 0	0	2670294 (s)
T18	7888	User	26800	Command wait	-1	0 0	0	2670294 (s)
T19	7888	User	26797	Command wait	-1	0 0	0	2670294 (s)
T20	7888	User	26795	Command wait	-1	0 0	0	2670294 (s)
T21	7888	User	26796	Command wait	-1	0 0	0	2670294 (s)

3 Tasks are in State 'Connect wait'

The most important process states are the following:

Command Wait	kernel waits for command
Vsuspend	wait state, e.g., for system resources
IOWait	process is waiting for I/O
Vsleep	process is not active for a short time
Vbegexcl	process is waiting for Critical Region (latch)
Running	process is running
Runnable	process is operable, but waits for CPU allocation
Vresume	prompting command

## 5.9Info / Regions

This menu function shows the states of all shared memory segments to which access is synchronized via semaphores. Especially interesting are the collision rates on certain shared memory segments. Collision rates over 10% are critical.

Evaluation of this data requires an understanding of the internal structures of ADABAS.

Example (of UNIX):

Index	Region	Owner	Get-Cnt	Tas-Cnt	Coll.	Excl	Coll/Get*100
1	BACKUP		0	0	0	0	0.00 %
2	BREAK		0	0	0	0	0.00 %
3	BUFWRTR		515823	76	3618	0	0.70 %
4	BUF2WRTR		3	0	0	0	0.00 %
5	CONFIG		239	0	0	0	0.00 %
6	DATAACH		1032765	0	1	0	0.00 %
7	DIAGCACH		692	0	0	0	0.00 %
8	DRDA		0	0	0	0	0.00 %
9	ERRTXT		0	0	0	0	0.00 %
10	FLUSH		0	0	0	0	0.00 %
11	KEYMEM		1	0	0	0	0.00 %
12	LOCK		33442	0	32	0	0.10 %
13	LOG		3677	0	4	0	0.11 %
14	LOGWRITE		3805	0	0	0	0.00 %
15	NET		40	0	0	0	0.00 %
16	NETDOWN		3	0	0	0	0.00 %
17	NETSEND		0	0	0	0	0.00 %
18	PERMFDIR		21942	0	0	0	0.00 %
19	PSM		8938	0	2	0	0.02 %
20	SURROGAT		321	0	0	0	0.00 %
21	TEMPFDIR		206	0	3	0	1.46 %
22	TRACE		498504	0	25	0	0.01 %
23	TREE		188800	0	0	0	0.00 %
Alone :							
Count : 45939			Collisions : 2				
Sleep Count : 38							

### 5.10Info / Memory

This menu function shows information about the database server's address space requirements.

Example (of UNIX):

Size	of	code	KB	3216
Size	of	task stacks	KB	3874
Size	of	shared data	KB	19772
Unused		shared dyn pool	KB	27
Unused		shared dyn data	KB	12
Number	of	existing tasks		30

### 5.11Info / Version

This menu function displays the current version of the database kernel and of CONTROL.

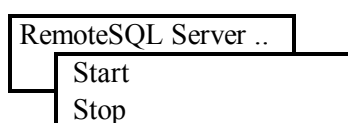


## 6 Options Menu Function

### 6.1 Options / Reset Counter

This menu function resets the database activity counters to zero.

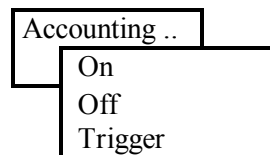
### 6.2 Options / Remote SQL Server



This menu function starts and shuts down the *Remote SQL Server* that is required for a client/server connection. If the *Remote SQL Server* is started, application processes running on another computer (client) can directly connect to the ADABAS server database and open database sessions there.

The **Remote SQL Server Start** menu function starts the remote SQL server. The **Remote SQL Server Stop** menu function aborts all connections that users have established from other computers to this database server.

### 6.3 Options / Accounting



The **Accounting On** menu function enables the recording of statistical information about resources used. This information is kept for a particular session. It is entered into the table SYSACCOUNT which has the following structure:

```

CREATE TABLE SYSACCOUNT (
    SERVERDBNO    FIXED    ( 4) KEY,
    SESSION       FIXED    (18) KEY,
    USERNAME      VARCHAR  (18),
    GROUPNAME     VARCHAR  (18),
    SENDERID      CHAR     ( 8),
    DBANAME       VARCHAR  (18),
    CONNECTDATE   DATE,
    CONNECTTIME   TIME,
    RELEASEDATE   DATE,
    RELEASETIME   TIME,
    COMMANDCOUNT FIXED    (10),
    CPUTIME        FIXED    (10),
    IOCOUNT      FIXED    (10),
    SESSIONEND     CHAR     ( 8),
    DBPROGTYPE     VARCHAR  ( 8),
    DBPROGOWNER    VARCHAR  (18),
    DBPROGNAME     VARCHAR  (18))
  
```

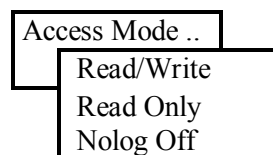
The data collected in the table SYSACCOUNT can be evaluated for a user-specific accounting. The data is not implicitly deleted or overwritten.

For sessions that do not leave traces in the table SYSACCOUNT although they run for a very long time, there is the **Accounting Trigger** menu function. This function can be used to signal each user session to enter a row of information into the table SYSACCOUNT. This signal is always checked before a new SQL statement is executed. After processing the signal, it has become meaningless. This means that a



new entry can only be made using **Accounting Trigger** again or at the end of the session.

## 6.4 Options / Access Mode

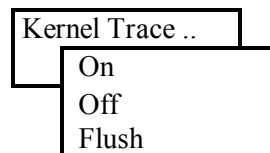


The **Access Mode** menu function enables or disables **write** access to the database. Usually, it is possible to write- and read-access the database. The **Access Mode/Read-Only** menu function prevents the database from being modified. Write transactions that are active while changing from READ/WRITE mode to READ-ONLY mode can terminate their write operations in a regular way.

The function is useful, e.g., when save operations are performed in WARM database mode. If no write commands are executed, no log is written. The data can no longer be modified.

In exceptional cases, the **Nolog Off** menu function can be used to cancel write protection for tables loaded with NOLOG. Usually, this is done when backing up the database.

## 6.5 Options / Kernel Trace

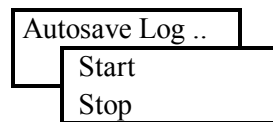


The **Kerneltrace On** menu function enables an ADABAS kernel trace for a particular command.

The **Kernel Trace Off** menu function disables the tracing. The **Kernel Trace Flush** menu function is only meaningful if the ADABAS kernel trace has been previously enabled. **Kernel Trace Flush** must be performed to write the remaining buffered entries to the trace file. This file cannot be read directly. It can only be interpreted by ADABAS customer support.

The trace can be evaluated with the Diagnose tool provided in the support mode of CONTROL.

## 6.6 Options / Autosave Log



The start option can be used to enable an automatic backup of log segments to a backup medium. To be able to assign a medium to the backup, the media list as described in the section 7.1 "Backup / Save" is displayed for selection. Whenever a log segment has been completed, the backup is automatically performed in background. If there is no log segment sufficiently filled to be saved, the system waits two minutes before checking for another completed log segment.

For the automatic backup of the log segments, we therefore recommend to use a separate backup device.

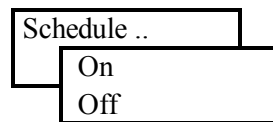
The wait state of the automatic log segment backup must be terminated before a *data backup* is performed ad hoc or within the Schedule Manager. This prevents a simultaneous backup of the data and a log segment. We recommend to change the tape used for the automatic log segment backup while saving the data to avoid that the "old" backups are overwritten after re-enabling the automatic backup. If the automatic log segment backup is enabled, backups of the *log* or of *log segments* cannot be executed ad hoc or within the Schedule Manager.

We recommend to use a log devspace that consists of at least two segments. Whenever a segment has been completed, the backup and subsequent clearing of this segment is automatically initiated. This has the advantage that a log overflow is almost impossible. The use of this mechanism is especially recommended for intensive write operations and long-running modifying transactions. Thus, the utilization level of the log devspace does not need to be monitored constantly.

It must be ensured, however, that there is sufficient free space on the backup medium for the resulting data stream. If the end of the tape has been reached and no media size has been specified for the medium, the backup terminates with the error message NEXT VOLUME REQUIRED. In this case, only the tape must be

changed. With the next start of Autosave Log, the corresponding log segment will be completely written to the new tape.

## 6.7 Options / Schedule

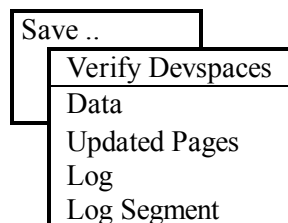


This option can be used to control the execution of scheduled actions of the Schedule Manager. If the Schedule option is enabled (**On**), all actions confirmed are executed. If the Schedule option is disabled (**Off**), actions can be scheduled and confirmed in the Schedule Manager. These actions, however, will not be started for execution.

## 7Backup Menu Function

This menu function allows you to save and restore the complete database contents and log contents, or only the modifications made since the last database or log backup. The medium used for the backup can be a tape or a file. The save or restore operations apply to only the addressed server database (SERVERDB). Ad-hoc backups are performed interactively; i.e., CONTROL expects that the required parameters are entered in screens. Input can also be expected for started backups, e.g., if the capacity of the backup medium is not sufficient for the backup. Incorrect entries or a timeout occurring because of delayed input can have the effect that the backup is aborted.

### 7.1Backup / Save



The following objects can be saved:

- |                      |  |
|----------------------|--|
| <b>Data</b>          | - saves the complete contents of the database<br>(SAVE DATA)         |
| <b>Updated Pages</b> | - saves all modifications made since the last backup<br>(SAVE PAGES) |
| <b>Log</b>           | - saves the complete log<br>(SAVE LOG)                               |
| <b>Log Segment</b>   | - saves an individual log segment<br>(SAVE LOG SEGMENT)              |

Each backup is done to a backup medium that can be selected from the Media Manager. The Media Manager is displayed after selecting the corresponding type of backup in the menu allowing the backup medium to be selected or defined (see section 8.2 "Backup / Media Manager" ). We recommend, however, to define the backup media in the Media Manager in advance.

After selecting, the backup expects a storage device in the specified backup medium. The storage device is provided with a label that indicates, e.g., the type of the backup. The label used for the backup is displayed and must be confirmed.

After selecting the corresponding type of backup in the menu, the Media Manager appears to select or define the backup medium (see section 8.2 "Backup / Media Manager").

### **Backup / Save / Verify Devspaces**

This menu function checks the consistency of the internal data structures in WARM or COLD operating mode. If there are serious inconsistencies, the database must be restored in the same way as after a disk failure.

In COLD operating mode, free storage pages wrongly recorded as used since an irregular end of database operation are released to the free space management.

We recommend a VERIFY before performing a complete backup of the database.

### **Backup / Save / Data**

The **SAVE DATA** menu function creates a backup version of the contents of the SERVERDB.

For backups in WARM mode, it must be taken into account that the database is saved with the state when the backup operation was started. Modifications to the contents of the database made during the backup operation are not saved.

In COLD mode, **SAVE DATA** can only be executed when the database is in a consistent state, i.e., after a normal SHUTDOWN.

It is necessary to perform a complete backup in adequate time intervals (e.g., at least weekly).

### **Backup / Save / Updated Pages**

The **SAVE UPDATED PAGES** menu function creates an incremental backup version of the local SERVERDB's data devspace. This backup version contains all pages updated since the last **SAVE DATA** or **SAVE UPDATED PAGES** (**SAVE PAGES**).

For backups in WARM mode, it must be taken into account that the database is saved with the state when the backup operation was started. Modifications to the contents of the database made during the backup operation are not saved.

In COLD mode, **SAVE UPDATED PAGES** can only be executed when the database is in a consistent state, i.e., after a normal SHUTDOWN.

Incremental backups are only advantageous when the database modifications concentrate on partial database devspaces. As a rule, backup times are quite short with **SAVE LOG**, but recovery may take more time.



### **Backup / Save / Log**

The **SAVE LOG** menu function creates a backup of the whole log. After successful termination of the backup operation, the log is cleared if the server database is in WARM operating mode.

**SAVE LOG** cannot be executed in log mode DEMO.

The log can be saved in WARM and in COLD mode. In COLD mode, the log is **not** cleared. The log must be backed up in adequate time intervals (e.g., daily), unless the backup is performed on log segments.

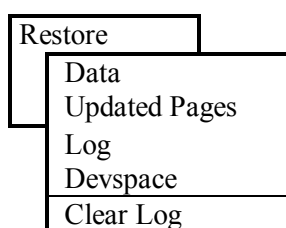
### **Backup / Save / Log Segment**

The **SAVE LOG SEGMENT** menu function creates a backup version of the oldest completed log segment. After successful termination of the backup operation, the log segment is cleared.

The log segments can be saved in WARM and in COLD mode. The backup of log segments must be performed in adequate time intervals (e.g., daily). It is recommended to use the option **AUTOSAVE LOG** in the **Options** menu to save a log segment automatically as soon as it has been completed.

For configurations without log segments, the **SAVE LOG SEGMENT** menu function is only provided in COLD mode. In this case, the whole log is saved as **one** segment and the log is cleared after successful termination of the backup.

## 7.2 Backup / Restore



The restore functions can only be used when the database system is in COLD operating mode (exception: RESTORE FROM MIRRORED DEVSPACE).

The following backups can be restored with **RESTORE**:

- |                      |  |
|----------------------|--|
| <b>Data</b>          | - restores a complete backup of the database     |
| <b>Updated Pages</b> | - restores an incremental backup                 |
| <b>Log</b>           | - restores the log devspace                      |
| <b>Devspace</b>      | - restores a devspace from the mirrored devspace |

After selecting the corresponding menu item, the Media Manager is displayed to select the restore medium (see section 8.2 "Backup / Media Manager").

For the restore, a storage device is expected in the specified backup medium. The label available on the tape is displayed and must be confirmed.

### Backup / Restore / Data

The **RESTORE DATA** menu function restores a backup version of the SERVERDB.

The configuration of the SERVERDB is not read from the backup version, but from the system devspace. Therefore, the name of the configuration parameter SYSTEM DEVSPACE must be identical with that of an intact system devspace.

### **Backup / Restore / Updated Pages**

The **RESTORE UPDATED PAGES** menu function restores an incremental backup version of the data devspace. Depending on the number of incremental backup versions, **RESTORE UPDATED PAGES** can be repeated in succession as often as is necessary. In doing so, be careful to restore the incremental backup versions exactly in the SAVE order.

## Backup / Restore / Log

The **RESTORE LOG** menu function restores a backup version of the log devspace, redoing the transactions recorded there.

This function overwrites the existing log devspace. Therefore, the current contents of the log must be saved with **SAVE LOG COLD** to an external backup device or into an external backup file, before the command **RESTORE LOG** can be executed for the first time.

Example:

The following illustration shows a data backup cycle, where three points in time of media failure occurrences are marked for the following recovery examples. Examples of tape labels that CONTROL uses to identify the individual save actions are given in parentheses on the right.

SAVE DATA	(1)	(DATA_A0)
Media Failure A.....		
	SAVE LOG SEGMENT (1)	(LOG_A1)
	SAVE LOG SEGMENT (2)	(LOG_A2)
	SAVE LOG SEGMENT (3)	(LOG_A3)
SAVE PAGES (1.1)		(DATA_A4)
	SAVE LOG SEGMENT (4)	(LOG_A5)
	SAVE LOG SEGMENT (5)	(LOG_A6)
	SAVE LOG SEGMENT (6)	(LOG_A7)
SAVE PAGES (1.2)		(DATA_A8)
	SAVE LOG SEGMENT (7)	(LOG_A9)
	SAVE LOG SEGMENT (8)	(LOG_A10)
Media Failure B.....		
	SAVE LOG SEGMENT (9)	(LOG_A11)
SAVE DATA (2)		(DATA_B0)
Media Failure C.....		
	SAVE LOG SEGMENT (10)	(LOG_B1)

The recovery procedures outlined in the following three examples require the existence of at least one intact log image which can be used in log mode DUAL or NORMAL to recover a defective log image using RESTORE LOG FROM DEV.

### Recovery After Media Failure A

To recover the database after the media failure A, only the first backup version of the data area must be restored using RESTORE DATA. The subsequent RESTART completes the data devspace redoing the transactions recorded in the log.

```
restore data (1)    (DATA_A0)
restart
```

### Recovery After Media Failure B

When media failure B occurs, there are several ways of recovering the SERVERDB. The quickest method of recovery consists of reloading the database using RESTORE DATA and subsequently reloading the modified pages using RESTORE PAGES. Finally, RESTORE LOG must be performed to restore the backups of the log segments 7 and 8.

The correct choice of the log segments 7 and 8 after restoring the pages (1.2) corresponds to the ascending SERVERDB version which can be found in the protocol of the corresponding data backup. A correct selection avoids failures.

Since RESTORE LOG overwrites the log, the current log version, which is not contained in the log segments saved so far, must be backed up to an external backup device or a host file using SAVE LOG COLD. Then the backups of the log segments can be restored using RESTORE LOG. In log mode DUAL or NORMAL, the backup of a log segment is implicitly copied onto both log images. Once the log segments 7 and 8 are restored, the copy of the current log that was backed up using SAVE LOG COLD must be restored using RESTORE LOG. Restart completes the recovery.

First restore variant:

```
restore data      (1)          (DATA_A0)
restore pages     (1.1)        (DATA_A4)
restore pages     (1.2)        (DATA_A8)
save log cold     (current log)
restore log       (log segment 7) (LOG_A9)
restore log       (log segment 8) (LOG_A10)
restore log       (current log)
restart
```

There is a choice of previous backups of the log segments which can be used for the recovery of the SERVERDB.

Second restore variant:

```
restore data      (1)          (DATA_A0)
restore pages     (1.1)        (DATA_A1)
save log cold     (current log)
restore log       (log segment 4) (LOG_A5)
...
restore log       (log segment 8) (LOG_A10)
restore log       (current log)
restart
```

Third restore variant:

```
restore data      (1)          (DATA_A0)
save log cold     (current log)
restore log       (log segment 1) (LOG_A1)
...
restore log       (log segment 8) (LOG_A10)
restore log       (current log)
restart
```

### Recovery after Media Failure C

When media failure C occurs, the SERVERDB can be recovered in the following way:

Only the last backup version of the data devspace needs to be restored. If this version is not readable for some reason, older data backup versions can be restored which require that the corresponding log segments are redone.

First restore variant:

```
restore data      (2)          (DATA_B0)
restart
```

Second restore variant:

```
restore data      (1)          (DATA_A0)
restore pages     (1.1)        (DATA_A4)
restore pages     (1.2)        (DATA_A8)
save log cold     (aktueller Log)
restore log       (Logsegment 7) (LOG_A9)
restore log       (Logsegment 8) (LOG_A10)
restore log       (Logsegment 9) (LOG_A11)
restore log       (aktueller Log)
restart
```

The same procedure must be used if organizational reasons require an older database state to be restored. RESTORE LOG UNTIL can then be used to select the point in time of the desired database state.

## Backup / Restore / Devspace

Depending on the state and configuration of the database, the following functions can be used:

### ***RESTORE LOG FROM DEVSPACE*** (COLD mode)

This function is used to recover a damaged log. In the event of a media failure in log mode DUAL or NORMAL causing an *Emergency Shutdown*, the defective log can be restored using **RESTORE LOG FROM DEVSPACE** (once the media failure has been corrected).

In log mode NORMAL, the Archive Log cannot be restored completely, because only the transactions relevant to the RESTART are copied from the Transaction Log to the Archive Log. In this case, this means, the Archive Log cannot be used to restore the database using a former backup version of the data area. Therefore a new backup of the data area is required which can only be performed after putting the data area into a consistent state by RESTART.

This function is only active if the database is in COLD mode and a DUAL LOG is defined.

### ***RESTORE FROM MIRRORED DEVSPACE*** (WARM mode)

This function is used to recover a damaged mirrored devspace. In the event of a media failure in mirrored devspace operation, the database continues working without this devspace, only accessing the intact devspace of the pair of mirrored devspaces concerned. Once this devspace has been repaired, it can be restored from the intact one using **RESTORE FROM MIRRORED DEVSPACE**. Afterwards, both devspaces work in normal mirrored devspace operation.

This function is only active if the database is in WARM mode and mirrored devspaces are configured.

In this sense, the log mode DUAL represents mirrored devspace operation. This means, in the event of a media failure in the log (Transaction Log, Archive Log or mirrored Archive Log), the database disables the defective devspace. Once the disabled devspace has been repaired, it can be re-integrated using **RESTORE FROM MIRRORED DEVSPACE**.



In the event of a second media failure in another log of the same level (e.g. Transaction Log failure when an Archive Log is already defective) results in an *Emergency Shutdown*. In this case, the log damaged last must be restored first using **RESTORE LOG FROM DEVSPACE** in COLD operating mode.

### **Backup / Restore / Clear Log**

This function is used to clear the complete log in order to resume the work on an old database state that was restored using a data backup reloaded with **RESTORE DATA**.

The command **CLEAR LOG** cannot be executed in a distributed database configuration for consistency reasons.

### 7.3 Backup / Show History

When selecting this menu item, the results of the backup operations performed so far are shown with the backup identification, backup type, date and time of backup start and of backup end, backup device, and current mode. This information is stored in the database in tabular format.

### 7.4 Backup / Show Protocol

The **SHOW PROTOCOL** menu function shows the end of the backup protocol file, where information about the last save or restore operation is recorded. It is possible to page up to the top of the file.

## 8 Advanced Backup Functions

### 8.1 Concepts

CONTROL supports the backup and restore procedures in a convenient way. These procedures can be activated immediately as described above; or they can be performed at fixed times based on a weekly timetable. If desired, a segment of the archive log is automatically saved to a particular tape device whenever the log segment has been completed.

#### *Ad -hoc Backups*

In WARM and COLD database mode, CONTROL provides the interactive backup of **Data** (physically complete), **Updated Pages** (physically incremental), **Log** and **Log Segment** (logically incremental) under the **Backup../Save..** menu item.

#### *Automatic Backup of Log Segments*

The oldest log segment can be automatically backed up in CONTROL as soon as the log segment has been completed. For this purpose, the automatic backup of the log can be enabled or disabled in CONTROL either under the Options../Autosave Log menu item or in the schedule using the actions AUTO-ON and AUTO-OFF. For the automatic backup of the log segments, a separate backup device must be used which must be accessible to the backup process at any time.

#### *Backups in the Schedule Manager*

The schedule under the Backup../Schedule Manager menu item can be used to plan the actions SAVEDATA (physically complete), SAVEPAGES (physically incremental), SAVELOG and SAVELOGSEG (logically incremental) as well as the actions AUTO-ON and AUTO-OFF.

### ***Verification and Optimizer Support in the Schedule Manager***

VERIFY and UPDSTAT (UPDATE STATISTICS) must be performed from time to time to guarantee a secure database operation and good performance. See the sections 2 "Concepts", 2.5 "Consistency Check and Optimizer Support" as well as "Backup / Save / Verify Devspaces" or 4.4 "Operating / Update Statistics".

The automatic backup of log segments as well as the backups, verification, and optimizer support in the schedule are performed in batch operation; i.e., the parameters must be defined before starting the action or when defining it using the Schedule Manager.

## **Backup Media**

Each backup action is assigned one or more backup media. A backup medium identifies the parameters of a physical backup device (tape device) and must be defined in the Media Manager before the backup. The backup is performed by sending the data to the tape device using the backup medium.

### ***Sequential Backup***

SAVE LOG and SAVE LOG SEGMENT are sequential backups.

A *continuation medium* can be defined for each medium. A sequential backup is continued there. Thus, a sequence of backup media to tape devices is defined. When backing up, the data is written successively to the defined continuation media, starting with the specified medium. No intervention by the operator is required if the capacity of the medium or the sequence of media is sufficient for the backup. CONTROL requests a (further) continuation medium if the specified medium does not suffice. In interactive mode, a form appears; otherwise, the backup is terminated with an error message.

If several tapes are used, the order of the tapes is relevant to a restore.

### ***Parallel Backup***

SAVE DATA and SAVE PAGES are parallel backups.

Media can be comprised to form a group of *parallel media* and be named. Parallel media are simultaneously written or read. This increases data throughput - and thus the speed of backup or restore. The name of a group of parallel media appears as an additional backup medium and can be selected for parallel backup actions according to the context.

If the capacity of the tapes is sufficient for the backup, no intervention of an operator is required during the backup. CONTROL requests more tapes if the backup has not been terminated although the tape is full. For this purpose the Media Size must have been specified for the media description.

A tape that might have been mounted later is parallel to any other tape used during the backup. A parallel backup can be done to one backup medium as a minimum.

The order of the tapes is ***not*** relevant for the restore of a parallel backup.

For a *sequential backup*, the assignment to a group of parallel media, if any, is not taken into account. For a *parallel backup*, the continuation media, if defined, are meaningless.

### ***Backups Performed Using Third-Party Backup Tools***

Presently, ADSM (ADSTAR Distributed Storage Manager) is supported.

Special names starting with 'ADSM' must be used for the backup media. CONTROL does not require any operator intervention during the backup. The sequence of the tapes for a restore is defined by the backup tool of the manufacturer.

### **Backup Generations**

In practice, it can happen that structural or media failures are already contained in the backup of a SERVERDB or tape. Sometimes there may be organizational reasons to restart with a previous state of the database.

This is possible if several backup generations are used. A backup generation consists of a complete backup and any number of subsequent incremental physical and logical backups. The next complete backup starts a new backup generation.

The administration of several backup generations provides several reentry positions for the recovery of a SERVERDB thus increasing data protection.

The backup generations are denoted by letters. The number of backup generations can be defined under the **Backup.. / Label Manager / Generations** menu item.

### **Media Label**

Each backup action is done to a storage device (tape) using a backup medium. CONTROL automatically provides the tapes with an identification (media label) that specifies the type of backup (complete or incremental backup of data or log), the backup generation, the number of the continuation tape or a letter for a parallel tape. The media label uniquely identifies the tape and must be written to the sticker of the tape. See section "Backup/Label Manager/The Media Label".

### A Concrete Backup Scheme

We recommend:

1. **four** backup generations
2. **one** SAVE DATA per week
3. regular or automatic backups of the log segments (AUTOSAVE LOG, AUTO-ON) between the complete backups (SAVE DATA). Note that AUTOSAVE LOG requires a tape device of its own.
4. SAVE PAGES in special application situations only

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1st Week	DATA_ A0		LOG_ A1	LOG_ A2	LOG_ A3	LOG_ A4	LOG_ A5
2nd Week	DATA_ B0		LOG_ B1	LOG_ B2	LOG_ B3	LOG_ B4	LOG_ B5
3rd Week	DATA_ C0		LOG_ C1	LOG_ C2	LOG_ C3	LOG_ C4	LOG_ C5
4th Week	DATA_ D0		LOG_ D1	LOG_ D2	LOG_ D3	LOG_ D4	LOG_ D5
5th Week	DATA_ A0		LOG_ A1	LOG_ A2	LOG_ A3	LOG_ A4	LOG_ A5

### An Example Timetable of the Backup Scheme:

Timetable..		Action..		Tools..		Help	
Timetable name : TIMETAB-1				Date : ...			
Timetable with 6 action(s)				Time : 10.00.00			
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
SAVELOGSI 18.00.00	SAVELOGSI 18.00.00	SAVELOGSEC 18.00.00	SAVELOGSEC 18.00.00	SAVELOGSEC 18.00.00	SAVEDATA 01.00.00		

For illustration purposes, the backup of log segments is explicitly scheduled in this example and was not implicitly initiated by AUTO-ON. This timetable can be applied to the schedule, e.g., starting on a Saturday. Thus the timetable corresponds exactly to the scheme above. The first backup then is a complete backup. The automatic backup of log segments (AUTOSAVE LOG) must not be enabled in this timetable.

As an alternative, the timetable can be defined with an automatic backup of the log segments, as recommended. In this case, the time of the log segment backups is not known in advance.



**Simplified Timetable Having the Same Effect:**

<b>Timetable..</b>	<b>Action..</b>	<b>Tools..</b>	<b>Help</b>			
Timetable name : TIMETAB-1 Timetable with 1 action(s)		Date : ... Time : 10.00.00				
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
					SAVEDATA 01.00.00	

For this timetable, the automatic backup of the log segments (AUTOSAVE LOG, AUTO-ON) must be enabled once. To avoid conflicts, it is implicitly disabled during the complete backup (here on Saturday) and be enabled afterwards.

## 8.2Backup / Media Manager

The MEDIA MANAGER can be used to create, update, or delete descriptions of the backup devices.

### Media List

If backup media are already defined when you start the Media Manager, the following Media List is displayed for selection:

FILE1		F	V	0	/backup/dblog.save	( )
FILE2		F	N	0	/backup/dblog.save	( )
DAT90	DAT120	T	N	4000	/dev/rmt0	(x)
DAT120		T	N	5000	/dev/rmt1	(x)
TWO		T	N	4000	/backup/dblog.save /dev/rmt0	( )
FOR-DATA					PARALLEL	
						Parallel-Id : FOR-DATA
Please select a backup medium from the list displayed above Confirm a selection with Show, otherwise CANCEL						
Show		Cancel		Drop		Edit
						New

*Fig.: Media List*

The following functions are possible on the medium at the cursor position:

<b>Edit</b>	- Modifies the medium using the Media Manager.	<b>F10</b>
	- A Parallel-Id cannot be edited.	
<b>Show</b>	- Displays the medium.	<b>Enter</b>
<b>Drop</b>	- Deletes the medium.	<b>F6</b>
<b>New</b>	- Defines a new medium. Therefore an empty media editor is displayed.	
<b>Cancel</b>	- Ends the media list.	<b>End</b> , <b>F3</b>

## Parallel Media

Use a mark in the last column of the form to display the assignment of the backup medium to the group of parallel media. The group is formed as soon as a mark is set. The name (Parallel-Id) of the group must be specified and must not violate the conventions of special media. The Parallel-Id is inserted into the media list and displayed in the form. This name must be used to select the group of parallel media for a backup. If the Parallel-Id is dropped, the group of parallel media is removed and all marks are deleted. If all marks are deleted, the Parallel-Id is dropped as well. A Parallel-Id cannot be edited.

## Media Editor

After selecting a medium or for the first call of the MEDIA MANAGER, the following input form is displayed if no media have been defined:

Media Manager	
Medium.....:	Next Medium.....:
Device Type.....: T	
Path1:	
Path2:	
Overwrite(Y/N/V): N	Media Size in Pages...: 0
<input type="button" value="Create"/>	<input type="button" value="Cancel"/>
<input type="button" value="Update"/>	<input type="button" value="Drop"/>
<input type="button" value="Prev"/>	<input type="button" value="Next"/>
<input type="button" value="Back"/>	

*Fig.: Media Editor, Media Manager*

The following information is required in the input screen:

MEDIUM	An identifier of up to 8 characters which can be chosen freely to determine the backup medium. See section "Special Media".
NEXT MEDIUM	An identifier of up to 8 characters which can be chosen freely to determine the continuation medium for a multi-volume backup performed without operator intervention. If the continuation medium is defined, the prompt for mounting another volume is suppressed and the backup is automatically continued with the specified continuation medium.
DEVICE TYPE	Indicates whether the backup medium is a tape device <b>T</b> or a file <b>F</b> .
PATH1	The name of the backup device or backup file.
PATH2	A second name can be specified here to create simultaneously a copy of the backup. The names must comply with the operating system conventions. Length: maximum of 64 characters.
OVERWRITE	This option is only evaluated when saving into a file. It allows an existing backup file to be overwritten or renamed. Valid input values are "Y", "N", and "V". "V" means version and provides a version identification for an existing file.
SIZE IN PAGES	The maximum number of 4KB pages to be written to a backup device or backup file. In case of multi-volume processing, subsequent screens request new volumes to be mounted or more files to be specified, until the backup is completed. If '0' is entered here, only one volume or one file is used for the backup in this medium; i.e., no multi-volume processing is performed. This requires that the capacity of the backup

medium is sufficient to store a complete database or log backup.

The following functions are possible:

<b>Create</b>	- Creates a new medium.	<b>Enter</b>	<b>F5</b>
<b>Cancel</b>	- Ends the media manager.	<b>End</b>	<b>F3</b>
<b>Update</b>	- Modifies the definition of the medium.		<b>F9</b>
<b>Drop</b>	- Drops the displayed medium.		<b>F6</b>
<b>Back</b>	- Returns to the list of defined media.		
<b>Prev</b>	- Displays the previous medium.		<b>Pgup</b>
<b>Next</b>	- Displays the next medium.		<b>Pgdn</b>

The functions **Prev** and **Next** can be activated according to the context.

### Special Media

To be able to use backup tools of different manufacturers, CONTROL expects the definition of special media. These must be identified by their names. The name of the backup medium controls the usage of these backup tools. Presently, CONTROL supports the backup tool ADSM. Any medium used for a backup with ADSM must begin with 'ADSM'.

The specified path name PATH1 denotes a named 'pipe'. CONTROL opens this 'pipe' to perform the backup or restore.

#### Convention for Special Media:

1. A special medium must not belong to a group of parallel media. For a parallel backup, the media ID must be specified directly.
2. A special medium must not be used as continuation medium.
3. The name of a special medium must not be used as Parallel-Id.

## 8.3 Backup / Label Manager

CONTROL provides storage devices used for backup and recovery with an identification, the media label. This label is displayed for a backup or recovery started ad hoc and must be confirmed.

### The Media Label

Each backup action is done to a storage device (tape, optical medium) using a backup medium. CONTROL automatically provides the tape with an identification (media label) which specifies the type of backup (complete or incremental backup of data or log) and the backup generation. The media label uniquely identifies the tape and must be written to the sticker of the tape.

#### Rules for Media Labels

1. Data backups are identified by 'DATA'; log backups by 'LOG'.
2. Each backup belongs to a backup generation. The generation of the backup can be recognized by the generation letter. All backups with the same letter belong to the same generation.
3. A complete backup of the database starts the next backup generation. The next letter is used.
4. Up to 26 backup generations can be defined (A,... Z).
5. For n generations defined, the (n+1)st 'SAVE DATA' is started again with the media label DATA\_A0\_A.
6. Complete backups receive the number '0'. Incremental data or log backups are numbered sequentially.
7. Sequential backups are written to continuation tapes. The order of the tapes is determined by the continuation medium. The first tape of a sequence (that can also consist of only one element) of backup media contains the continuation tape identification 1.
8. Parallel backups are written to parallel tapes. A parallel tape is identified by letters (A .. ZZ). Up to 16 backup media can be operated in parallel. The tape

devices accessed by the backup media can be multiply supplied with tapes. All tapes of such a parallel backup are parallel to each other.

9. Log segments backed up automatically (AUTOSAVE LOG, AUTO-ON) are continuously written to a tape and, if this is full, to a continuation tape. The labels of these backups have the same format as the continuation tapes described above. Note that several backups of log segments can be stored on a tape.

### Formal Description of the Media Labels

```

< medialabel > ::=          < saveindicator > _
                             < generation >
                             < number of incremental save > _
                             < successor or parallel indicator >

< saveindicator > ::=      DATA | LOG

< generation > ::=         A .. Z

< number of incremental save > ::=
    0                      for a complete save
    1 .. 99                for an incremental save

< successor or parallel indicator > ::=
    1 .. 999               for a sequential save
    | A .. ZZ              for a parallel save
  
```

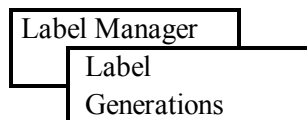
Examples :

DATA_A0_A	complete save of the database (SAVE DATA); first generation, parallel tape 'A'
DATA_C2_D	second incremental data save (SAVE PAGES), third generation, parallel tape 'D'
LOG_D3_2	third incremental log save (SAVELOG or SAVELOGSEG), fourth generation, second continuation tape.



## Backup Generation

The number of generations to be used for the backup can be defined under the **Backup / Label Manager / Generations** menu item.



For this purpose, the presently valid number of generations and the interval of letters resulting from this setting are displayed. The new number of generations can be specified. The corresponding interval of letters will be automatically updated when the input field is being left.

Number of Save Generations	
current value	: 4 ( A .. D )
new value	: 12 ( A .. L )
<div><input type="button" value="Ok"/> <input type="button" value="Cancel"/></div>	

The new value is valid at once and will be used as the new number of backup generations for the next backup. Four generations are defined as the default setting.

## 8.4 Backup / Schedule Manager

### *The Schedule Manager*

The **Backup/Save** menu function provides you with an option for performing immediate (ad hoc) saves. In addition, CONTROL includes a Schedule Manager for scheduling one-time and periodic save operations. The save mechanism provided by the Schedule Manager is optional and can be used together with ad hoc saving. All activities - whether ad hoc or scheduled - use the same save procedure and write log entries to a history file, thus providing you with an overview of previous and future actions at any time.

The schedule can be prepared starting from the current week and continuing for up to one year. You can display up to 255 past log entries.

### *Timetables*

In order to facilitate the scheduling of regularly recurring actions, the CONTROL Schedule Manager allows you to define timetables to be used as templates for actions or to accept timetables from the scheduling. A number of timetables can be defined and applied to specific weeks in the Schedule Manager. The timetable form and functions are the same as those of the Schedule Manager.

### *Using the Schedule Manager*

The Schedule Manager can be used only when the database is in WARM mode. When it is in any other mode, the **Backup/Schedule Manager** menu item is deactivated (and cannot be selected). Save operations must then be performed ad hoc.

### *Activating the Schedule Manager*

The actions scheduled in the Schedule Manager are performed only if you use the **Options/Schedule** option. You can activate and deactivate the schedule using the **Options/Schedule/ON** and **Options/Schedule/OFF** menu items contained in the CONTROL TOP form (possible with the WARM status only). You can change this option using the **Tools/Schedule/ON** or **Tools/Schedule/OFF** menu item in the Schedule Manager. Both the Schedule Manager form or the value of the option in the TOP form indicate whether the schedule is active or passive (the currently valid state cannot be activated). You can deliberately deactivate the schedule in order to allow for times when the database is shut down or for scheduling purposes. In the Schedule Manager, you can select and display any week in the past and all weeks in the future for up to one year (see section "Backup / Schedule Manager / Week").

### *Calling the Schedule Manager*

Select the **Backup/Schedule Manager** menu item to branch to the Schedule Manager. The schedule for the current week is displayed.

### *The Schedule Manager Form*

Below is an example of a Schedule Manager; the day selected is Wednesday, November 30, 1994 and the week is week 48:

Week..		Action..		Tools..		Help..	
Schedule : ON		Phase : Present		Date : 30.10.199			
Planned : 25		Last at : 11.01.1995		Time : 10.00.00			
Week 48.!							
Monday 28.11	Tuesday 29.11	Wednesday >30.11<	Thursday 01.12	Friday 02.12	Saturday 03.12	Sunday 04.12	
SAVEDB 18.00.00 SAVED		SAVEPAGES 12.00.00 WAIT		UPDSTAT 05.30.00 INSERT		SAVELOG 12.00.00 WAIT	
		SAVLOGSEG 14.00.00 WAIT		SAVEPAGES 10.00.00 INSERT			
		VERIFY 23.59.59 WAIT					

*Fig.: Timetable*

### *Scheduling Actions*

Actions can either be scheduled individually for a specific start time or transferred to the schedule by means of a timetable. In order for the actions to be performed, they must first be confirmed (see "Action List"). An action can only be confirmed if its start time is at least five minutes after the confirmation time and within a year.

### ***Saving the Schedule Status***

When you exit the Schedule Manager, you are informed of any actions that have been scheduled but not confirmed. You can then exit the Schedule Manager without losing the schedule status and resume scheduling at any time in the future.

### ***The Timetable Form***

Select the **Week/Timetable** menu item from the Schedule Manager to branch to timetable editing. If no timetables are defined, the timetable form is empty. If only one timetable is defined, the form contains this timetable. If more than one timetable is defined, a list of the names of the timetables is displayed from which you can make a selection.

The timetable form has the same divisions as the Schedule Manager form:

<b>Timetable..</b>		<b>Action..</b>		<b>Tools..</b>		<b>Help..</b>	
Timetable name : Timetab-1				Date : 30.10.1994			
Timetable with 6 action(s) and 2 change(s)				Time : 10.00.00			
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	
SAVEDB 18.00.00		SAVEPAGES 12.00.00	UPDSTAT 05.30.00		SAVELOG 12.00.001		
		SAVLOGSEG 14.00.00	SAVPAGES 10.00.00				

*Fig.: Timetable*

The two forms have the same layout; their appearance, menu bars, elements, and functions are shown and described in the sections below.

### The Information Lines

Information lines are displayed below the action bar for the Schedule Manager and for timetable editing:

#### *Schedule Manager Information Lines:*

Schedule : ON		Phase : Present		Date : 30.10.199		
Planned : 25		Last at : 11.01.1995		Time : 10.00.00		
Week 48.94						
Monday 28.11	Tuesday 29.11	Wednesday >30.11<	Thursday 01.12	Friday 02.12	Saturday 03.12	Sunday 04.12
...						13 >>

#### *Timetable Information Lines:*

Timetable name : Timetab-1				Date : 30.10.199		
Timetable with 6 action(s) and 2 change(s)				Time : 10.00.00		
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday

These information lines indicate whether the Schedule option is activated or deactivated (**ON/OFF**). Actions that are scheduled and confirmed will be performed only if the Schedule option is activated. You can change the status of this option using the **Tools/Schedule** menu item in the Schedule Manager or the **Options/Schedule** menu item in the CONTROL TOP form when the database is in WARM mode.

### ***Displaying the Current Date and Time***

The current date and the current time are displayed on the right-hand side in the information lines.

### ***Displaying the Schedule Phase (Schedule Manager only)***

Below the schedule status, 'Phase' specifies whether the selected schedule is in the future (Future), past (History), or present (Present).

### ***Displaying the Scheduled Actions (Schedule Manager only)***

Below the schedule status, the number of scheduled (confirmed or unconfirmed) actions is displayed. Next to it, the most future begindate is displayed.

### ***Displaying the Schedule Week (Schedule Manager only)***

The schedule week selected is specified above the day and date with the format WW.YY (WW = number of the calendar week; YY = last two digits of the year). The selected day is marked in the action schedule's date display by the symbols '>' and '<'. You can change the selected week or day using the **Week/Any** F2 menu function.

### ***Displaying Unconfirmed Actions (Schedule Manager only)***

Unconfirmed actions are identified in the action schedule by the INSERT, UPDATE, or DELETE status. The number of unconfirmed actions scheduled before and after the selected week are displayed on the line below the date display, to the left and right, next to the symbols '<<' and '>>'. Confirmed actions are identified in the Schedule Manager by the WAIT status.

***Displaying the Name (Timetables only)***

For timetables, the name of the schedule is displayed in place of the schedule week.

***Displaying the Actions and Changes (Timetables only)***

For timetables, the number of actions and changes since the last save of the timetable are displayed.

***Help Texts and Error Messages***

Help texts and error messages are displayed on the last line of the forms.

**The Action Schedule**

Actions that have been or will be performed on a particular day are listed below the relevant day in the order of their start times. The action schedule displays the name, start time, and status of each action.

SAVEPAGES	Action Name
12.00.00	Start Time
WAIT	Status



### ***Action Name***

The following actions are possible:

Action	Description
SAVEDATA	saves the database completely
SAVEPAGES	saves the database pages incrementally
SAVELOG	saves the log incrementally
SAVELOGSEG	saves the oldest log segment incrementally
AUTO-ON, AUTOON	activates autosave
AUTO-OFF, AUTOOFF	deactivates autosave
UPDSTAT	updates statistical data
VERIFY	checks the consistency of the database

*Fig.: Action List*

### ***Start Time***

For future actions, this field displays the scheduled start time; for past actions, it displays the actual start time.

### ***Status***

Modified actions can have the INSERT, UPDATE, or DELETE status. Scheduled and confirmed actions have the WAIT status (they are performed at the scheduled time if the schedule option is set). The status of actions that have already been performed is determined by the result of the save operation.  
In timetables, the status of an action is not displayed.

### ***Actions in Timetables***

The start date in a timetable is a day of the week. When applied, the action is extended to every occurrence of this day of the week within the application period (see section "Backup / Schedule Manager / Week").

### *Actions in the Schedule Manager*

Actions whose scheduled start times have expired are not displayed in the Schedule Manager. In their place is the log entry that was generated by these actions. You can display and edit the actions belonging to another calendar week by selecting this week (see section "Backup / Schedule Manager / Week").

### *Moving the Cursor in the Action Schedule*

You can move the cursor in the action schedule using the TABULATOR (= 'Tab'), ,  and cursor keys.

### *Scrolling in the Action Schedule*

Using  and , you can display up to ten actions per day. Each time you press one of these keys, you page through one action line (= four screen lines). The cursor continues to be positioned on the action for as long as it remains visible.

If more than ten actions have been scheduled or performed on a given day, a message to this effect is displayed at the tenth position in the action table. Since the action form also allows you to scroll through the actions of a selected week (see "Displaying Actions"), you can use it to edit the actions that are not displayed.

### *Displaying Actions*

Select the **Action/Zoom** menu item or press  to display or modify an action. The action on which the cursor is located is then displayed in the action form. If the cursor is not located on an action but is, instead, located on a future day or the current day, an empty action form is displayed. For actions that are represented by their log entries (i.e., past actions), a message is output indicating that they must not be modified or deleted. However, it is possible to change their date and reschedule them.

Action			
Action Name	: SAVEPAGES	Medium	: Tape1
Begin Date	: 30.11.1994	Begtime	: 12.00.00
Status	: WAIT	Creator	: CONTROLDBA
Create Date	: 29.11.1994	Create Time	: 14.03.26
End Date	:	End Time	:
Returncode	:	Generation	:

Press  to call up the help function for the "Action Name", "Medium", "Begin Date" and "Begin Time" input fields; you can then select the correct values. When you select "Insert"/, "Update"/, and "Delete"/ to transfer the data, all input values are checked for correctness. CONTROL sets the "Creator", "Create Date"/"Create Time", "End Date"/"End Time", "Generation" and "Returncode" output fields when the action is scheduled, confirmed, or performed.

The confirmation time must be at least five minutes before the scheduled start time ("Begin Date" and "Begin Time") (see "Action List") in order to allow sufficient time for the action to be stored.

If an action form is empty, the day on which the cursor is located is displayed as the default setting and can be overwritten. The default time is the current time plus half an hour. Input must be in the selected SET format. Press  to display and accept the selected SET format, the default date or the default time. The current date is used as the default setting for past actions that are to be modified.

**Note:**

Be sure to schedule a sufficient period of time between successive actions in order to avoid situations where actions must wait. Save actions do not wait, i.e., they are aborted with an error if an action is already running.

You must enter or select a medium for all actions requiring a medium (see section 8.2 "Backup / Media Manager"). In such cases, you can press  to display a list of media. No input is necessary for actions that do not use a medium.

In the case of scheduled actions, "Create Date"/"Create Time" indicate when the status was last modified. For ad hoc actions, these fields indicate the activation time. Depending on the hardware load, there may be a difference between the activation time and the start time.

"End Date"/"End Time", "Returncode" and "Generation" are used only when actions have already been performed. They are not displayed in timetables.

"Insert"/, "Update"/, and "Delete"/ allow you to insert, update or delete the displayed action. Select "Reset"/ to display the action that was originally selected by using "Zoom"/. "Prev" (=Previous)/ and "Next"/ serve to display the previous or next action in the selected week or in the timetable. Select "Cancel"/ to exit the form; if you have modified any values, a warning form is output.

### ***History, Present, Future (Schedule Manager only)***

If the calendar week selected is entirely in the past, the action table contains only the log entries for the actions that were performed; if the week is entirely in the future, it contains only those actions that are scheduled. If the week selected contains the current date and time, the log entries up until the current time are displayed along with the scheduled actions as of the current time.

### Backup / Schedule Manager / Week

The scheduling unit in CONTROL is one week. For this reason, the actions for a calendar week are displayed in an action schedule (see section "The Action Schedule"). The **Week** menu function allows you to change to a different calendar week.

When you select the **Week** menu function, the following pull-down menu is displayed:

<b>Week..</b>	<b>Action..</b>	<b>Tools..</b>	<b>Help..</b>
---------------	-----------------	----------------	---------------

Prev	F7
Next	F8
Any	F2
Timetable	
Quit	F3

**Week/Prev** F7

This menu item displays the Schedule Manager for the previous calendar week.

**Week/Next** F8

This menu item displays the Schedule Manager for the next calendar week.

**Week/Any**

When you select the **Week/Any** menu item, the date form is displayed in which you can enter any calendar week or date and thus change to a different schedule week. You can enter either the date or the calendar week in the relevant input field (Date or Week). The other input and output fields will be adapted accordingly. Select "Ok"/  to transfer them to the Schedule Manager.

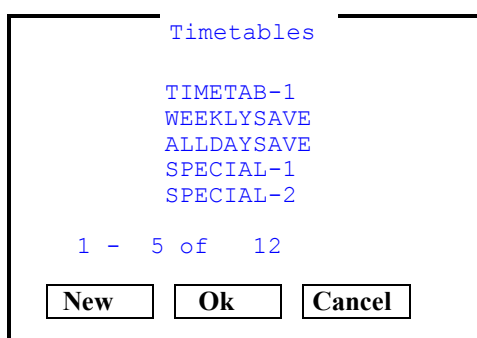
Date		
Date	: 30.10.1994	Week : 48.94 (WW.YY)
Day	: Wednesday	Monday: 28.10.1994
Phase	: Present	Sunday: 04.12.1994
<input type="button" value="Ok"/>	<input type="button" value="Cancel"/>	<input type="button" value="Date"/>
<input type="button" value="Prev"/>	<input type="button" value="Next"/>	

Select "Date"/  to set the current date. Select "Prev"/  or "Next"/  to obtain the week that precedes or follows the date displayed; the day of the week itself remains the same. All date information is expected in the selected SET format and the calendar week is expected in the format "WW.YY".

### **Week/Timetable**

The **Week/Timetable** menu item allows you to branch from the Schedule Manager to timetable editing. If no timetables are defined, the timetable form is empty. If only one timetable is defined, the form contains this timetable. If more than one timetable is stored, a list of the names of the timetables is displayed from which you can make a selection.

Example of a list of names:



```
Timetables

TIMETAB-1
WEEKLYSAVE
ALLDAYSAVE
SPECIAL-1
SPECIAL-2

1 - 5 of 12

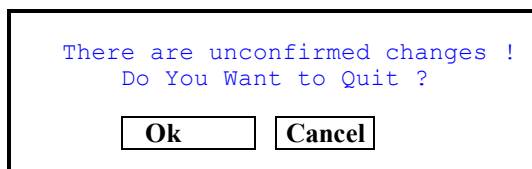
New Ok Cancel
```

Use "Ok"/ to select the name on which the cursor is located; use "New"/ to open an empty timetable form.

In forms in the Schedule Manager where names must be specified in input fields, you can press  to display such a kind of lists as help.

**Week/Quit**

Select this menu item to exit the Schedule Manager. If you have not confirmed all actions (see "Action List"), a warning is output:



```
There are unconfirmed changes !
Do You Want to Quit ?

Ok Cancel
```

Unconfirmed actions in the Schedule Manager are kept as schedule status.

### Backup / Schedule Manager / Action

When you select the **Action** menu function, the following pull-down menu is displayed in the Schedule Manager (in the case of timetables, the **Confirm**, **Search**, and **Apply** items are omitted):

Week..	Action..	Tools..	Help..
--------	----------	---------	--------

Zoom	Enter
Delete	F6
Confirm	F5
Search	F9
Apply Timetable	
Save As Timetable	

The **Zoom** and **Delete** menu functions apply to the action in the action schedule on which the cursor is currently located (see section "The Action Schedule"). **Confirm** applies to all actions that have not yet been confirmed and **Search** applies to a set of actions to be retrieved (see "Action List").

*Action/Zoom* Enter

When you select the **Zoom** menu item, the action form is displayed allowing you to view, enter, and modify the action. In the case of timetables, **Zoom** displays only the portion of the action form that is relevant here. If you select **Zoom** when the cursor is not located on a scheduled action in the future, an empty action form is opened.

**Zoom** can be used to modify the status that is displayed for an action (see "Action Schedule/Status").



**Action/Delete** F6

The **Delete** menu function allows you to delete an action from the schedule without applying **Zoom** to this action. The displayed status is changed to DELETE. Deleted actions are not removed from the schedule until you select **Confirm**, thus allowing you to continue to use an action marked with DELETE as a template for other actions or to modify this action and change its status to UPDATE.

**Action/Confirm** F5

**Confirm** allows you to confirm the scheduled actions. All actions with the INSERT, UPDATE, and DELETE status are displayed in an action list.

Example of an action list:

Actions to Confirm			
Action	Begin Date & Time		Status
UPDSTAT	01.12.1994	05.30.00	INSERT
SAVEPAGES	01.12.1994	10.00.00	INSERT
...			
1 - 10 of 15			
<span style="border: 1px solid black; padding: 2px;">Confirm</span>	<span style="border: 1px solid black; padding: 2px;">Cancel</span>	<span style="border: 1px solid black; padding: 2px;">Top</span>	<span style="border: 1px solid black; padding: 2px;">Bottom</span>

Select "Confirm" or F5 to confirm the actions in the schedule (INSERT or UPDATE) or to delete them from the schedule (DELETE). The status of confirmed actions changes to WAIT. A warning is displayed if the Schedule option is not enabled, so that the confirmed actions become not effective.

Select "Top"/F7 or "Bottom"/F8 to position the cursor at the beginning or end of the list. Use Pgup and Pgdn to scroll through the contents of a screen.

**Action/Search** 

When you select the **Search** menu function, the following selection form is displayed:

Search			
Action Name	: SAVEPAGES	Medium	: Tapel
Begin Date	: 30.11.1994	Begtime	: 12.00.00
Status	: WAIT	Creator	: CONTROLDBA
Create Date	: 29.11.1994	Create Time	: 14.03.26
End Date	: *	End Time	: *
Returncode	: *	Generation	: *
<input type="button" value="Show"/> <input type="button" value="Confirm"/> <input type="button" value="Delete"/> <input type="button" value="Reset"/> <input type="button" value="Cancel"/>			

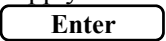
The selection form contains the same fields as the action form; in this case, all the fields are input fields. An asterisk '\*' can be used to represent any number of characters. Pressing "Reset"/ places an '\*' in the field on which the cursor is located; pressing this key again places '\*' in all fields of the form. The help key  allows you to display and, if desired, select correct values.

When you select "Show"/, "Confirm"/ or "Delete"/, the entered values are combined by AND to form a query on all scheduled actions. All actions that match this query are displayed in an action list and can be edited. If no action is found, a warning form is output.

You can specify a start time in the "Begin Date" and "Begin Time" fields. The result then includes all actions that are to be started after this start time. (The search does not find actions that have already been performed.)

You can specify an end time in the "End Date" and "End Time" fields. The result then includes all actions that should have been (!) started before this end time.

If you specify a "Creator", and particularly if you specify the name of a timetable, you can then select all actions generated by the specified creator. Press  to display a list of the names of all creators and timetables. You can specify a creation time in the "Create Date" and "Create Time" fields. The result then includes all actions that were scheduled after this creation time.

An action list is generated from the values entered in the selection form and then displayed. This action list allows you to apply the selected function to the selected actions using the associated button or . The selected actions are all confirmed at the same time.

#### *Action/Apply Timetable*

You can use the **Action/Apply Timetable** menu item to apply timetables from the Schedule Manager. If at least one timetable exists, the application form described under **Timetable/Apply** is displayed; you can then apply a timetable to the specified calendar weeks. The actions thus entered must still be confirmed in the Schedule Manager (with **Confirm**) before they can be activated. If no timetable is defined, a warning is output.

#### *Action/Save As Timetable*

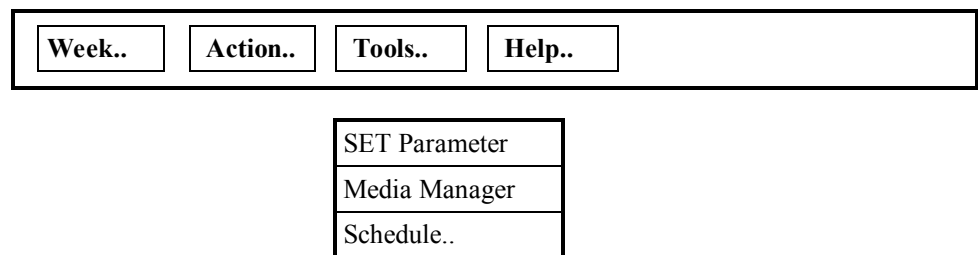
You can use the **Action/Save As Timetable** menu item to generate a timetable from the actions of the displayed week. The procedure and the corresponding form are equivalent to the function described in the section **Timetable/Store As**.

#### *Changes of Status*

Actions that have been deleted (Status: DELETE), inserted (Status: INSERT) or updated (Status: UPDATE) retain their status even after the end of a session (see "Week/Quit") if they have not been confirmed (see "Action List"). Thus, the schedule status is retained for subsequent sessions.

### **Backup / Schedule Manager / Tools**

When you select the **Tools** menu function, the following pull-down menu is displayed:



This menu item allows you to call other tools from CONTROL.

#### ***Tools/SET Parameter***

This menu item displays a form for entering and modifying the SET parameters. The following SET parameters can be selected:

TIME FORMAT (same as for other tools)

DATE FORMAT (same as for other tools)

#### ***Tools/Media Manager***

This option calls the Media Manager (see section 8.2 "Backup / Media Manager") from CONTROL, which allows you to enter, modify and delete media.

#### ***Tools/Schedule***

The option for the usage of the schedule can be changed. A menu allows you to activate the corresponding state (**ON**, **OFF**). When enabling the schedule, all

confirmed actions are accepted for the schedule. When disabling the schedule, all actions entered by the Schedule Manager are removed from the schedule.

### Backup / Schedule Manager / Help

This menu item displays a help form for the Schedule Manager. For the usage of help see section 3.5 "CONTROL Menu Structure and Help Texts".

### Backup / Schedule Manager / Timetable

When you select the **Timetable** menu function, the following pull-down menu is displayed in the timetable form:

Timetable..	Action..	Tools..	Help..
-------------	----------	---------	--------

Show	F9
New	F2
Delete	
Store	
Store As	
Apply	F5
Quit	F3

For the **Show**, **New**, **Apply**, **Delete**, and **Quit** actions, a warning is output if you modify the schedule displayed but do not store it (**Store**, **Store As**) since otherwise your modifications would be lost.

**Timetable/Show** F9

This menu item displays a list of the names of timetables for your selection.

**Timetable/New** 

This menu item displays an empty form for a new timetable.

**Timetable/Apply** 

This menu item applies the displayed timetable to the Schedule Manager.  
The following application form is displayed:

Apply	
Name : TIMETAB-1	
Week : 48.94	from to Week : 50.94
Date : 01.12.1994	to Date : 23.12.1994
Day : Thursday	Day : Friday
<input type="button" value="Reset"/>	<input type="button" value="Ok"/> <input type="button" value="Cancel"/>

The name of the displayed timetable is used in the form as a default setting. You can also enter the name of any other timetable or select one using .

You can specify the calendar weeks to which the timetable is to be applied. The first and last days of the week specified are then displayed in the fields beneath them. You also have the option of specifying a date. The calendar week or the specified date must be within a year. The timetable is applied within the specified time interval exactly on the specified day.

Select "Ok"/ to check all input fields for correct syntax and semantics and to enter the actions of the timetable in the specified weeks. The actions thus entered must still be confirmed in the Schedule Manager (with **Confirm**) before they can be activated.

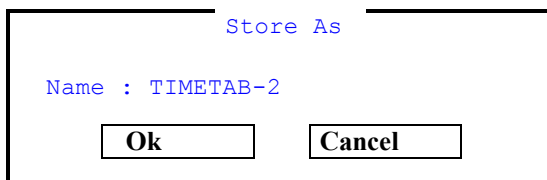
**Timetable/Store**

If the displayed timetable is named, it is stored under this name. If it is not named, you are asked to enter a name (see "Store As").



### *Timetable/Store As*

This menu item allows you to store the displayed timetable under a (different) name. You must enter the name in the name form:



Store As

Name : TIMETAB-2

Ok Cancel

Select "Ok"/**Enter** to store the displayed timetable under the specified name. A check is run to determine whether the specified name already exists; if it does exist, a warning is output. You then have the option of overwriting the existing timetable or specifying a different name. Press **F1** to display a list of the names that have already been assigned; this list must *not* contain the new name.

### *Timetable/Delete*

This menu item allows you to delete the displayed timetable. As a precaution, a warning form is output. Afterwards, this timetable can no longer be selected. Actions that have already been entered in the Schedule Manager by a previous application (**Apply**) are not modified. Especially, the entry of the creator remains in the scheduled actions even if the creating timetable has been deleted.

### *Timetable/Quit* **F3**

This menu item exits timetable editing. If you have not saved the timetable displayed (**Timetable/Save**, **Timetable/Save As**), a warning is output. If you exit a timetable that you have not saved, your modifications are lost.

## 8.5Batch Mode

In addition to the backup options described above (ad hoc, schedule, timetable), CONTROL provides a batch call which can be used to perform backup and verification actions.

### *Format*

**Call:**

```
xbackup [ Optionen ]
```

**the following options can be used :**

- |              |  |
|--------------|--|
| -a <action>  | The actions can be used as in the schedule.<br>(see Fig. "Action List")<br>Default : SAVEDATA  |
| -d <dbname>  | Name of the local SERVERDB<br>Default : \$SERVERDB or \$DBNAME   |
| -m <mediaid> | The backup medium denoted by <mediaid> must<br>have been previously created in the Media Manager<br>of CONTROL.<br>Default : none; if only one medium is defined in<br>CONTROL, this will be used. |
| -r <dbroot>  | Name of the ADABAS software directory<br>Default : \$DBROOT environment variable   |
| -q           | 'quiet';<br>an output is only generated in an error case.  |
| -V           | displays the version; no processing.   |



## 9 Diagnose Menu Function

### 9.1 Diagnose / Op Messages

This menu function shows the console log file of the ADABAS server. In the display screen, it is possible to page down using the **Next** menu function and to page up using the **Previous** menu function. The **End** menu function terminates the display screen and returns to the main screen of CONTROL.

The console log file helps you to interpret the events that occurred during ADABAS server operation. The entries are made in chronological order. When it reaches a certain size, the console log file is overwritten cyclically. A line with dashes denotes the current end of the console log file.

The console log file records events such as the following:

- Starting and shutting down the ADABAS server,
- Information about the physical storage areas (DEVSPACES) of the ADABAS server,
- Information about user processes which have established database sessions with the ADABAS server,
- Error messages of the ADABAS server system embedding which may be the result of selected invalid system parameter sizes,
- System error messages, e.g., after device failures,
- Error messages and warnings of the ADABAS kernel as output, e.g., after an *Emergency Shutdown* (see section 3.4 "The Main Screen" on the subject Usage Levels) or for internal inconsistencies, etc.

## **9.2 Diagnose / Log Messages**

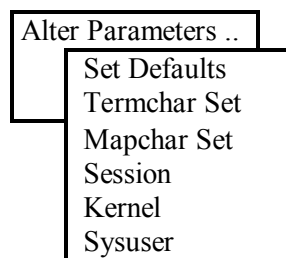
This menu function shows all actions executed so far in CONTROL. In addition, the results and errors which might have occurred are displayed for the activities performed. The cursor is always positioned at the end of the log messages, because the most recent information is recorded there.

## **9.3 Diagnose / Inst Protocol**

This menu function displays the console log resulting from the last installation or from loading the system tables. The cursor is always positioned at the end of the installation log, because the result of the installation and an overview of the errors occurred is recorded there.

## 10 Configuration Menu Function

### 10.1 Configuration / Alter Parameters



#### Configuration / Alter Parameters / Set Defaults

This menu function can be used to display and modify the default SET parameters of the SYSDBA.

### Configuration / Alter Parameters / Termchar Set

This menu function can be used to create, display, alter, and delete termchar sets. To define a new termchar set, alter the CHAR SET NAME of an existing termchar set. Afterwards, create is provided instead of alter.

Example:

Alter Parameter Termcharset <Serverdb> on <Servernode>							
CHAR SET NAME		IBM437_GER	CHAR SET CODE		ASCII		
ENABLE CHAR SET Y							
CODE	1	C4 8E	A-umlaut	CODE	2	E4 84	a-umlaut
CODE	3	D6 99	O-umlaut	CODE	4	F6 94	o-umlaut
CODE	5	C4 8E	U-umlaut	CODE	6	FC 81	u-umlaut
CODE	7	C4 8E	sharp s	CODE	8	A7 15	paragragh
CODE	9			CODE	10		
CODE	11			CODE	12		
CODE	13			CODE	14		
CODE	15			CODE	16		

Cancel
Alter
Drop
Print

Fig.: Termchar Set Screen

### Configuration / Alter Parameters / Mapchar Set

This menu function can be used to create, display, alter, and delete mapchar sets. To define a new mapchar set, alter the CHAR SET NAME of an existing mapchar set. Afterwards, create is provided instead of alter.

Example:

Alter Parameter Mapcharset <Serverdb> on <Servernode>									
CHAR SET NAME		Defaultmap	CHAR SET CODE		ASCII				
CODE	1	A1	!	CODE	2	BF	?		
CODE	3	C0	A	CODE	4	C1	A		
CODE	5	C2	A	CODE	6	C3	A		
CODE	7	C4	Ae	CODE	8	C5	Aa		
CODE	9	C6	Ae	CODE	10	C7	C		
CODE	11	C8	E	CODE	12	C9	E		
CODE	13	CA	E	CODE	14	CB	E		
CODE	15	CC	I	CODE	16	CD	I		

Next Page

Cancel

Alter

Drop

Print

Fig.: Mapchar Set Screen



**Configuration / Alter Parameters / Session**

This menu function can be used to alter the default code, the date and time format, and the timeout values. The modifications become effective only with the next RESTART. In the input screen, the parameters are set to the last defined values.

Example:

Alter Session Parameter <Serverdb> on <Servernode>	
<hr/>	
DEFAULT CODE	ASCII
DATE TIME FORMAT	INTERNAL
SESSION TIMEOUT	900
LOCK TIMEOUT	360
REQUEST TIMEOUT	180
<hr/>	
<div><input type="button" value="Ok"/> <input type="button" value="Print"/> <input type="button" value="Cancel"/></div>	

*Fig.: Session Parameter Screen*

### Configuration / Alter Parameters / Kernel

This menu function displays the configuration parameters of the ADABAS server. In the lower part of the screen, the parameters are described briefly.

Example:

Alter Kernel Parameter <Serverdb> on <Servernode>

---

SYSDEVSPACE	/u/dev/SYS1
MIRR SYSDEVSPACE	
TRANSACTION_LOG	/dev/log0DB1
ARCHIVE_LOG	/dev/log1DB1
MIRR ARCHIVE LOG	
MAXDEVSPACES	8
MAXDATADEVSPACES	3
MAXSERVERDB	1
MAXBACKUPDEVS	2
MAXSERVERTASK	6
MAXUSERTASK	50
MAXDATAPAGES	150000
MAXCPU	1

Logical name of the first ADABAS SYSTEMDEVSPACE

Next

Explain

Print

Ok

Cancel

*Fig.: Kernel Parameter Screen 1*

Alter Kernel Parameter <Serverdb> on <Servernode>

---

DATA CACHE PAGES	3000
PROC_DATA_PAGES	100
PROC_CODE_PAGES	50
TEMP_CACHE_PAGES	50
CATALOG_CACHE_PAGS	96
CONV CACHE PAGES	100
MAXLOCKS	800
RUNDIRECTORY	/sqldb/E20/db/wrk/E20
OPMSG1	/dev/syscon
OPMSG2	/dev/null

Number of 4KB blocks to be allocated for the data cache  
in main memory

Prev
Explain
Print
Ok
Cancel

*Fig.: Kernel Parameter Screen 2*

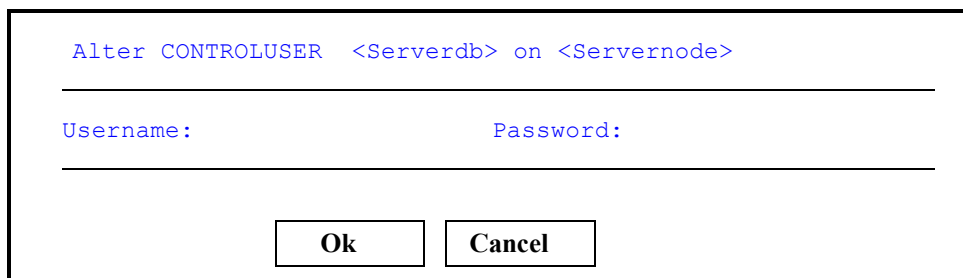
The user has the possibility to modify the system parameters. When the parameters are confirmed with **Enter**, all parameters are checked. If the modifications of the parameters lead to deviating computations for related values, the user can choose between the entered and the computed value. Other errors are shown in the displayed console log.

Finally, the minimum values required for the configuration parameters of the operating system kernel are displayed, so you can check the settings. If this should be necessary, adapt the operating system kernel to these requirements.

If parameters have been modified, they become effective only after a shutdown and subsequent restart of the SERVERDB.

### Configuration / Alter Parameters / Sysuser

Example:



Alter CONTROLUSER <Serverdb> on <Servernode>

Username: Password:

Ok Cancel

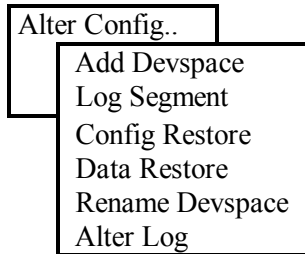
*Fig.: Alter Sysuser Screen*

The password of the SYSDBA and of the DOMAIN user as well as the name and password of the CONTROLUSER can be modified. The passwords of the SYSDBA and of the DOMAIN user can only be modified in WARM mode. The password of the CONTROLUSER can only be modified if the mode is **not** WARM.

After entering the user name and password of the previous user definition, the system uses the user name to determine the sysusertype to be modified. When entering the new definition, the password must be specified twice for security reasons.

If the SYSDBA has been modified outside CONTROL, the system recognizes during the logon to CONTROL that the definition is not identical to the profile data and requests the input of correct data for the SYSDBA.

## 10.2 Configuration / Alter Config



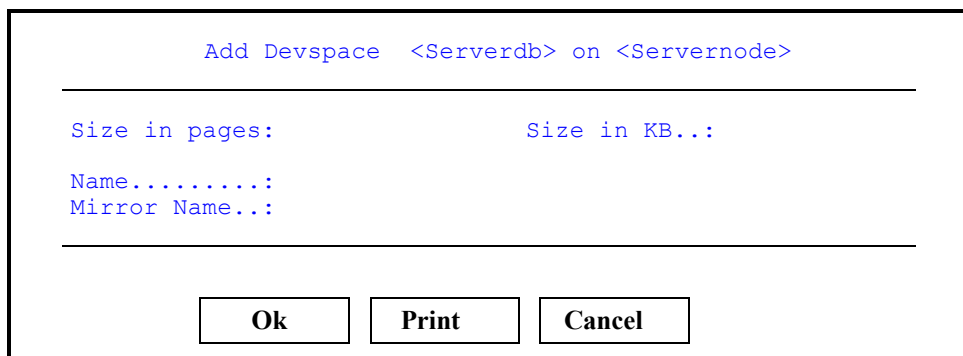
Alter Config can be used to modify the configuration of the SERVERDB at a later point in time. All functions, except Add Devspace, can be executed only if the SERVERDB is in COLD mode.

### Configuration / Alter Config /Add Devspace

The **Add Devspace** menu function expands the ADABAS server by the specified new physical storage area.

Activating the **Add Devspace** menu function displays a popup window on the main screen into which the size of the new physical storage area must be entered in 4 KB storage pages, along with the operating system name of the physical storage area, e.g., *Raw Device* under UNIX. Before the database server can be expanded, the access rights of the ADABAS server must be checked for the particular *Raw Device*.

Example:



Add Devspace <Serverdb> on <Servernode>

---

Size in pages:                      Size in KB..:

Name.....:

Mirror Name..:

---

Ok      Print      Cancel

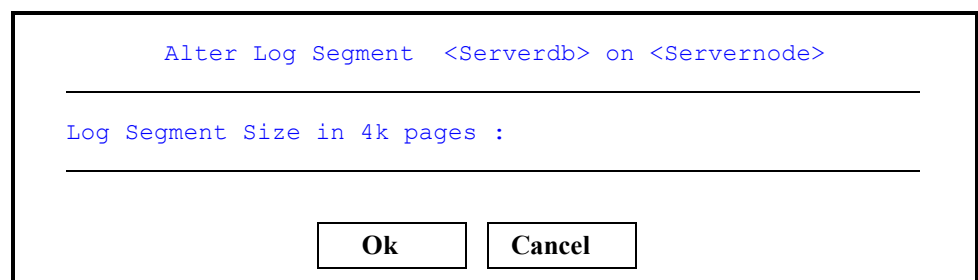
*Fig.: Add Devsapce Screen*

The Mirror Name is only required if the configuration is set to mirrored devspace operation (MIRRORED = Y).

### Configuration / Alter Config / Log Segment

This menu function can be used to alter the maximum segment size for log backups. If the log segment size is to be recuded, Save Log should be performed before Alter Log Segment, because otherwise the restart could fail for a high usage level of the log.

Example:



Alter Log Segment <Serverdb> on <Servernode>

---

Log Segment Size in 4k pages :

---

Ok Cancel

*Fig.: Alter Log Segment Screen*

The segment size, specified in KB, must not exceed the size of the archive log. The specification '0' means that the size of the log segment is identical to that of the archive log. The segment size is preset to the value valid so far and can only be modified in COLD mode.

### Configuration / Alter Config / Config Restore

This menu function can be used to modify the configuration of devspaces; it restores a backup version of the data area for this purpose. To ensure that the backed up dataset is identical to the last valid dataset, this backup version should have been generated in COLD mode just before Alter Config.

The procedure of Alter Config consists of several phases. First, the configuration is simply read from the backup version to initialize the input screens with the existing configuration parameters.

Alter Config Restore <serverdb> on <dbnode>	
LOG MODE	NORMAL
LOG SEGMENT SIZE	1500
NO OF ARCHIVE LOGS	1
NO OF DATADEVSPACES	2
MIRRORED	N
LOG MODE can be SINGLE, NORMAL, DUAL or DEMO	
Next Page	Print
Cancel	

*Fig.: Input Screen 1 for Alter Config Restore*



To make sure that the complete data from the backup version fits into the reconfigured SERVERDB, the size of the data devspace must not be decreased. After loading the backup version, the log devspace is reformatted.

Alter Config Restore <serverdb> on <dbnode>				
NAME	TYPE	SIZE	DEVSPACE PATH	
SYSTEMDEV	F	-	/u/dev/SYS1	
TRANS LOG	R	3000	/dev/log0DB1	
ARCHLOG 1	R	3000	/dev/log1DB1	
DATDEV 01	R	50000	/dev/dat01DB1	
DATDEV 02	R	50000	/dev/dat02DB1	

Ok
Prev Page
Print
Cancel

*Fig.: Input Screen 2 for Alter Config Restore*

After modifying the configuration, the parameters are reinitialized; then stop and restart are used to activate the parameters and make known the new configuration of the SERVERDB. Afterwards, the rest of the backup version is implicitly restored, redistributing the data pages across the devspaces.

Alter Config Restore <serverdb> on <dbnode>	
SET KERNEL PARAMETERS.....	OK
--> STOP AND START SERVERDB COLD.....	ACTIVE
ALTER CONFIG PARAMETERS.....	--
RESTORE DATA.....	--

*Fig.: Status Screen for Alter Config Restore*

### Configuration / Alter Config / Data Restore

This menu function can be used to alter the configuration of the data DEVSPACES before restoring a backup version of the data devspace.

The procedure of Alter Data Restore is analogous to that of Alter Config Restore. First, the name of the path where the backup is located must be specified. The backup version is displayed and must be confirmed. Then the names, sizes, and number of data DEVSPACES can be modified.

The log is kept.

### Configuration / Alter Config / Rename Devspace

This menu function can be used to rename the paths of the DEVSPACES.

Rename Devspace <serverdb> on <dbnode>			
NAME	TYPE	SIZE	DEVSPACE PATH
SYSTEMDEV	F	-	/u/dev/SYS1
TRANS LOG	R	3000	/dev/log0DB1
ARCHLOG 1	R	3000	/dev/log1DB1
DATDEV 01	R	50000	/dev/dat01DB1
DATDEV 02	R	50000	/dev/dat02DB1

*Fig.: Input Screen for Rename Devspace*

### Configuration / Alter Config / Alter Log

This menu function can be used to alter the log mode, as well as the names and sizes of the log DEVSPACES.

Alter Log <serverdb> on <dbnode>

---

LOG MODE	NORMAL
LOG SEGMENT SIZE	1500
NO OF ARCHIVE LOGS	1
NO OF DATADEVSPACES	2
MIRRORED	N

LOG MODE can be SINGLE, NORMAL, DUAL or DEMO

Next Page
Print
Cancel

*Fig.: Input Screen 1 for Alter Log*

Alter Log <serverdb> on <dbnode>

NAME	TYPE	SIZE	DEVSPACE PATH
SYSTEMDEV	F	-	/u/dev/SYS1
TRANS LOG	R	3000	/dev/log0DB1
ARCHLOG 1	R	3000	/dev/log1DB1
DATDEV 01	R	50000	/dev/dat01DB1
DATDEV 02	R	50000	/dev/dat02DB1

Ok
Prev Page
Print
Cancel

*Fig.: Input Screen 2 for Alter Log*

### 10.3 Configuration / Load Systables

For a new version, this function can be used to update the system tables.

```

Load System Tables for Update Installation

```

---

```

---> Create general systemtables..... ACTIVE
Load messages and help infos..... --
Load SET defaults..... --
Load system tables for precompilers..... --
Load system tables for QUERY..... --
Load system tables for EASY..... --
Load system tables for SQL-PL..... --
Load SQL-PL WORKBENCH..... --
Load system tables for QueryPlus..... --
Create system views..... --
Create ODBC tables..... --
Create SQL catalog views..... --
Load system DB PROCEDURES..... --

```

*Fig.: Status Screen for Load System Tables*

## 10.4 Configuration / Init Config

This menu function can be used to reconfigure the existing SERVERDB. The procedure corresponds to that of a first installation (see section 3.2 "Installing a New SERVERDB"), whereby the values valid so far are provided. In any case, the system tables must be loaded after the **Init Config** either by loading a DBEXTRACT using the component LOAD or by restoring a database backup using the CONTROL **Backup/Restore/Data** menu function or by loading the tables using the **Configuration/Load Systables** menu function.

**WARNING:** After Init Config, the old database contents are lost.

## 10.5 Configuration / Clear Serverdb

This menu function removes the current SERVERDB. When confirming this selection, **the complete data** of this database **will be lost**. Any information about the server database and its contents will be deleted. Afterwards, the server database name is available again.

## 11Index

### - A -

Access Mode.....	6-3
Accounting.....	6-2
action.....	8-22
action list.....	8-30
action schedule, moving the cursor in.....	8-23
action schedule, scrolling in.....	8-23
action, confirm.....	8-30
action, delete.....	8-30
action, displaying.....	8-23
action, edit.....	8-29
action, name.....	8-22
action, search.....	8-31
action, start time.....	8-22
action, status.....	8-22
action, zoom.....	8-23
actions, scheduling.....	8-17
Activity, information.....	5-2
Add Devspace.....	10-9
additional installation, CONTROL.....	3-10
ADMS.....	8-3
Alter Config.....	10-8
Alter Config Restore.....	10-11
alter configuration.....	10-8
alter configuration parameters.....	10-1
Alter Data Restore.....	10-13
Alter Kernel Parameters.....	10-5
Alter Log.....	10-14
Alter Log Segment.....	10-10
Alter Parameters.....	10-1
alter session parameter.....	10-4
Alter Session Parameters.....	10-4
Alter Set Default Parameters.....	10-1

Alter Sysuser.....	10-7
application form.....	8-35
Archive Log.....	2-2
Archive Log Devspace.....	2-2
AUTO-OFF.....	8-1
AUTOOFF.....	8-1
AUTO-ON.....	8-1
AUTOON.....	8-1
AUTOSAVE LOG.....	2-5; 6-5
availability.....	2-15

## **- B -**

backup.....	2-5; 7-1
backup copy.....	8-10
backup generation.....	8-4; 8-12; 8-14
backup medium.....	8-2
backup protocol, show.....	7-12
backup, ad hoc.....	8-1
backup, batch mode.....	8-37
backup, complete.....	2-5
backup, incremental.....	2-5; 2-6
backup, information.....	7-12
backup, log.....	7-4
backup, log segment.....	6-5; 7-4
backup, log segment, automatic.....	8-1
backup, logical.....	2-5
backup, parallel.....	8-3
backup, physical.....	2-6
backup, scheduled.....	8-1
backup, sequential.....	8-2
backup, SERVERDB.....	2-5
batch mode, backup.....	8-37

**- C -**

caches.....	2-9
Caches, Information.....	5-2; 5-10
call, CONTROL.....	3-1
Catalog Cache.....	2-9
Clear Log.....	7-11
Clear Serverdb.....	10-16
clear, SERVERDB.....	10-16
client server.....	2-12
complete backup.....	2-5
configuration parameters.....	2-16
configuration parameters, alter.....	10-1
configuration, alter.....	10-8
Configuration, information.....	5-7
console log, error display.....	9-2
console log, result display.....	9-2
console protocol, show.....	9-1
continuation medium.....	8-2; 8-10; 8-12
CONTROL USER.....	2-14
CONTROL, additional installation.....	3-10
CONTROL, call.....	3-1
CONTROL, exit.....	4-4
CONTROL, main screen.....	3-11
Converter Cache.....	2-9
copy, backup.....	8-10
counters, reset.....	6-1

**- D -**

data access, enable/disable.....	6-3
data backup.....	2-5
Data Cache.....	2-9
data consistency, check.....	7-2
Data Devspace.....	2-2
data devspace, restore.....	7-6
data devspace, save.....	7-3



database processes, information.....	5-17
dedicated users.....	2-14
DEVSPACE wiederherstellen.....	7-11
DEVSPACES, number.....	3-6
display of the CONTROL version.....	5-19
distribution.....	2-12; 3-5
DOMAIN USER.....	2-14

## - E -

error display.....	9-2
Exit.....	4-4
exiting from CONTROL.....	4-4

## - H -

help texts.....	3-15
History display.....	7-12

## - I -

I/O Accesses, information.....	5-12
I/O, information.....	5-4
incremental backup.....	7-3; 7-6
information about backup.....	7-12
information about caches.....	5-2; 5-10
information about configuration.....	5-7
information about database processes.....	5-17
information about I/O.....	5-4
information about lock list.....	5-5; 5-13
information about logging.....	5-6
information about logs.....	5-14
information about main memory usage.....	5-19
information about search operations.....	5-6
information about SERVERDB activity.....	5-2

information about shared memory segments.....	5-18
information about sorts.....	5-6
information about SQL statements.....	5-3
information about storage space.....	5-7
information about users.....	5-9
Init Config.....	10-16
Inst Protocol.....	9-2
installation protocol.....	9-2
installation, SERVERDB.....	3-2

## **- K -**

kernel parameters, alter.....	10-5
Kernel Trace.....	6-4
Kernel Trace Flush.....	6-4

## **- L -**

Label Manager.....	8-12
Load Systables.....	10-15
lock list, information.....	5-5; 5-13
Locks, information.....	5-13
log.....	2-3
Log Messages.....	9-2
log segment.....	2-4
log segment, alter.....	10-10
log segment, backup.....	6-5; 8-1
log segment, save.....	7-4
log, clear.....	7-11
log, restore.....	7-7
log, save.....	7-4
log, usage level.....	2-4
logging.....	2-3
logging, information.....	5-6
Logs, information.....	5-14

**- M -**

main memory, usage.....	5-19
main screen of CONTROL.....	3-11
main screen, updating values.....	4-1
mapchar set.....	10-3
MAXCPU.....	2-11
MAXUSERTASKS.....	2-11
media description, create.....	8-8
media description, delete.....	8-8
media description, update.....	8-8
media label.....	8-4; 8-12
Media Manager.....	7-2; 8-8
medium.....	8-2; 8-8
Memory, information.....	5-19
menu function Configuration.....	10-1
menu function Diagnose.....	9-1
menu function Info.....	5-1
menu function Operating.....	4-1
menu function Options.....	6-1
menu structure of CONTROL.....	3-15
MIRRORED.....	2-15
Mode Accounting On.....	6-2
Mode Accounting Trigger.....	6-2
Mode Database.....	6-3
multiprocessor configurations.....	2-11
multi-volume backup without operator.....	8-10

**- N -**

name form.....	8-36
name list.....	8-28

**- O -**

Operating Messages.....	9-1
-------------------------	-----

operating mode, SERVERDB.....3-13

## **- P -**

paging, in form.....9-1  
parallel medium.....8-3; 8-13  
Proc Code Cache.....2-9  
Proc Data Cache.....2-9  
Processes, information.....5-17  
Protocol.....7-12  
protocol, backup, show.....7-12  
protocol, console, show.....9-1  
protocol, enable/disable.....6-4  
protocol, installation.....9-2

## **- R -**

Read-Only Mode.....6-3  
Refresh.....4-1  
Regions, information.....5-18  
REMOTE INSTALL.....3-5  
Remote SQL Server.....6-1  
Rename Devspace.....10-13  
Reset Counter.....6-1  
RESTART.....2-7; 4-2  
RESTORE DATA.....2-7; 7-6; 7-8  
RESTORE DEVSPACE.....7-11  
RESTORE FROM MIRRORED DEVSPACE.....7-11  
RESTORE LOG.....2-7; 7-7; 7-8  
RESTORE LOG FROM DEVSPACE.....2-8; 7-11  
RESTORE LOG UNTIL.....2-7  
RESTORE PAGES.....2-7; 7-8  
RESTORE UPDATED PAGES.....7-6  
restore, data devspace.....7-6  
restore, log.....7-7  
restore, SERVERDB.....2-7; 7-6

result display.....9-2

### - S -

SAVE.....7-1  
 SAVE DATA.....2-5; 7-1; 7-3  
 SAVE LOG.....2-6; 7-1; 7-4  
 SAVE LOG COLD.....7-8  
 SAVE LOG SEGMENT.....2-5; 7-1; 7-4  
 SAVE PAGES.....2-6; 7-1; 7-3  
 save, log segment.....6-5  
 save, SERVERDB.....7-3  
 Schedule Manager.....8-15  
 Schedule Manager / Action Schedule.....8-21  
 Schedule Manager / Information Lines.....8-19  
 Schedule Manager, actions in.....8-23  
 Schedule Manager, activating.....8-16  
 Schedule Manager, calling.....8-16  
 Schedule Manager, displaying unconfirmed actions.....8-20  
 Schedule Manager, error message.....8-21  
 Schedule Manager, form.....8-17  
 Schedule Manager, save as timetable.....8-32  
 Schedule Manager, schedule phase.....8-25  
 Schedule Manager, using.....8-15  
 Schedule Manager/Action.....8-29  
 Schedule Manager/Help.....8-34  
 Schedule Manager/Timetable.....8-34  
 Schedule Manager/Tools.....8-33  
 Schedule Manager/Week.....8-26  
 Schedule option.....6-6; 8-16; 8-19; 8-33  
 schedule phase.....8-20  
 schedule status.....8-18  
 schedule week.....8-20  
 schedule, changing the display.....8-26  
 search operation, information.....5-6  
 selection form.....8-31  
 SERVERDB.....2-1

SERVERDB activity, information.....	5-2
SERVERDB structure.....	2-1
SERVERDB, alter devspace.....	10-13
SERVERDB, alter log.....	10-14
SERVERDB, backup.....	2-5
SERVERDB, clear.....	10-16
SERVERDB, configure.....	10-16
SERVERDB, expand.....	10-9
SERVERDB, installation.....	3-2
SERVERDB, Konfiguration verändern.....	10-11
SERVERDB, operating mode.....	3-13
SERVERDB, rename data devspace.....	10-13
SERVERDB, restore.....	2-7; 7-6
SERVERDB, save.....	7-3
SERVERDB, shut down.....	4-3
SERVERDB, start.....	4-2
SERVERTASKS.....	2-12
session parameter, alter.....	10-4
Set Default parameters, alter.....	10-1
shared memory segments, information.....	5-18
shut down, SERVERDB.....	4-3
Shutdown.....	4-3
sorts, information.....	5-6
SPONSOR SERVERDB.....	2-12
SQL statements, information.....	5-3
start, remote SQL server.....	6-1
start, SERVERDB.....	4-2
statistics, access to SERVERDB.....	5-12
statistics, resources used.....	6-2
statistics, update.....	4-4
stop, remote SQL server.....	6-1
storage space, information.....	5-7
SYSDBA.....	2-14
System Devspace.....	2-1
system tables, update.....	10-15
system user, alter.....	10-7

**- T -**

Temp Cache.....	2-9
termchar set.....	10-2
timetable.....	8-34
timetable, apply.....	8-32; 8-35
timetable, create newly.....	8-35
timetable, form.....	8-18
timetable, name.....	8-21
timetable, show.....	8-34
timetable, store.....	8-35
timetable, store as.....	8-36
timetable,delete.....	8-36
timetable. generate from Schedule Manager.....	8-32
timetables.....	8-15
timetables, actions in.....	8-22
trace, enable/disable.....	6-4
Transaction Log.....	2-1
Transaction Log Devspace.....	2-1

**- U -**

UPDATE STATISTICS.....	2-8; 4-4
UPDATE STATISTICS, scheduled.....	8-2
Users, information.....	5-9

**- V -**

VERIFY.....	2-8
Verify Devspaces.....	7-2
VERIFY, scheduled.....	8-2
Version, information.....	5-19

**- W -**

warning form.....	8-28
WITH REPLICATION.....	2-13





## Table of Contents

<b>1</b>	<b>Introduction.....</b>
<b>2</b>	<b>Concepts.....</b>
2.1	SERVERDB Structure.....
2.2	Logging.....
2.3	Backup.....
2.4	Restore.....
2.5	Consistency Check and Optimizer Support.....
2.6	Caches.....
2.7	Multiprocessor Configurations.....
2.8	Client Server.....
2.9	Distribution.....
2.10	Special Users.....
2.11	Availability.....
2.12	Configuration Parameters.....
<b>3</b>	<b>Call and Installation.....</b>
3.1	Calling CONTROL.....
3.2	Installing a New SERVERDB.....
3.3	The Additional Installation of CONTROL.....
3.4	The Main Screen.....
3.5	CONTROL Menu Structure and Help Texts.....
<b>4</b>	<b>Operating Menu Function.....</b>
4.1	Operating / Refresh.....
4.2	Operating / Restart.....
4.3	Operating / Shutdown.....
4.4	Operating / Update Statistics.....
4.5	Operating / Exit.....
<b>5</b>	<b>Info Menu Function.....</b>
5.1	Info / Activity.....
5.2	Info / Configuration.....

5.3	Info / Users.....
5.4	Info / Caches.....
5.5	Info / I/O Accesses.....
5.6	Info / Locks.....
5.7	Info / Logs.....
5.8	Info / Processes.....
5.9	Info / Regions.....
5.10	Info / Memory.....
5.11	Info / Version.....
<b>6</b>	<b>Options Menu Function.....</b>
6.1	Options / Reset Counter.....
6.2	Options / Remote SQL Server.....
6.3	Options / Accounting.....
6.4	Options / Access Mode.....
6.5	Options / Kernel Trace.....
6.6	Options / Autosave Log.....
6.7	Options / Schedule.....
<b>7</b>	<b>Backup Menu Function.....</b>
7.1	Backup / Save.....
7.1.1	Backup / Save / Verify Devspaces.....
7.1.2	Backup / Save / Data.....
7.1.3	Backup / Save / Updated Pages.....
7.1.4	Backup / Save / Log.....
7.1.5	Backup / Save / Log Segment.....
7.2	Backup / Restore.....
7.2.1	Backup / Restore / Data.....
7.2.2	Backup / Restore / Updated Pages.....
7.2.3	Backup / Restore / Log.....
7.2.4	Backup / Restore / Devspace.....
7.2.5	Backup / Restore / Clear Log.....
7.3	Backup / Show History.....
7.4	Backup / Show Protocol.....
<b>8</b>	<b>Advanced Backup Functions.....</b>
8.1	Concepts.....
8.1.1	Backup Media.....

8.1.2	Backup Generations.....
8.1.3	Media Label.....
8.1.4	A Concrete Backup Scheme.....
8.2	Backup / Media Manager.....
8.2.1	Media List.....
8.2.2	Parallel Media.....
8.2.3	Media Editor.....
8.2.4	Special Media.....
8.3	Backup / Label Manager.....
8.3.1	The Media Label.....
8.3.2	Backup Generation.....
8.4	Backup / Schedule Manager.....
8.4.1	The Information Lines.....
8.4.2	The Action Schedule.....
8.4.3	Backup / Schedule Manager / Week.....
8.4.4	Backup / Schedule Manager / Action.....
8.4.5	Backup / Schedule Manager / Tools.....
8.4.6	Backup / Schedule Manager / Help.....
8.4.7	Backup / Schedule Manager / Timetable.....
8.5	Batch Mode.....
<b>9</b>	<b>Diagnose Menu Function.....</b>
9.1	Diagnose / Op Messages.....
9.2	Diagnose / Log Messages.....
9.3	Diagnose / Inst Protocol.....
<b>10</b>	<b>Configuration Menu Function.....</b>
10.1	Configuration / Alter Parameters.....
10.1.1	Configuration / Alter Parameters / Set Defaults.....
10.1.2	Configuration / Alter Parameters / Termchar Set.....
10.1.3	Configuration / Alter Parameters / Mapchar Set.....
10.1.4	Configuration / Alter Parameters / Session.....
10.1.5	Configuration / Alter Parameters / Kernel.....
10.1.6	Configuration / Alter Parameters / Sysuser.....
10.2	Configuration / Alter Config.....
10.2.1	Configuration / Alter Config / Add Devspace.....
10.2.2	Configuration / Alter Config / <a href="#">Log Segment</a> .....
10.2.3	Configuration / Alter Config / Config Restore.....

10.2.4	Configuration / Alter Config / Data Restore.....
10.2.5	Configuration / Alter Config / Rename Devspace.....
10.2.6	Configuration / Alter Config / Alter Log.....
10.3	Configuration / Load Systables.....
10.4	Configuration / Init Config.....
10.5	Configuration / Clear Serverdb.....
<b>11</b>	<b>Index.....</b>