

Database Administration CONTROL (BC)



HELP.BCDBADADBA

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®], and Motif[®] are registered trademarks of the Open Group.







HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT[®] is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

Database Administration CONTROL (BC)	6
Introduction.....	7
Overview	8
Using CONTROL.....	9
CONTROL Menu Structure and User Guidance	10
Terminal and Color Settings in CONTROL	13
Installing a New Serverdb.....	15
Step-by-step Installation of the Serverdb	17
Installation of the Serverdb using a Backup	19
Update Installation	21
The Main Screen	22
Files used by CONTROL.....	25
Backup Concepts and Strategies	27
Backup Media	28
Medium Label	29
Examples of a Backup Scheme	31
Backup/Restore - a Practical Guide	33
Saving to a Single Medium	34
Recovering from a Single Medium	35
Saving to a Medium with Continuation Media.....	36
Recovering from a Medium with Continuation Media.	37
Saving to Parallel Media without Continuation Medium.....	38
Recovering from Parallel Media (Without Continuation Media).....	39
Saving to Parallel Media with Continuation Media.....	40
Recovering from Parallel Media (With Continuation Media).....	41
Examples of Backup/Restore.....	43
External Backup Tools.....	47
Autoloader under Windows NT.....	49
Other Autoloaders	50
Supported Recovery	51
Batch Mode: xbackup/xrestore	52
Batch Mode: xcontrol.....	53
Menu Functions	54
Operating.....	55
Operating=>Colors	56
Operating=>Refresh.....	57
Operating=>Restart.....	58
Operating=>Shutdown	59
Operating=>Update Statistics	60
Operating=>Exit	62
Info	63
Info=>Activity.....	65
Info=>Configuration.....	66
Info=>Kernel Parameters	68

Info=>Users.....	69
Info=>Caches.....	70
Info=>I/O Accesses.....	72
Info=>Locks.....	73
Info=>Log.....	75
Info=>Last UpdStatistics.....	77
Info=>Tasks.....	78
Info=>Version.....	79
Info=>Help.....	80
Options.....	81
Options=>Autorefresh.....	82
Options=>Remote SQL-Server.....	83
Options=>Accounting.....	84
Options=>Kernel Vtrace.....	85
Backup.....	86
Backup=>Save.....	87
Backup=>Restore.....	89
Backup=>Show History.....	90
Backup=>Show Protocol.....	91
Backup=>Media Manager.....	92
Diagnose.....	94
Diagnose=>Op Messages.....	95
Diagnose=>Command History.....	96
Diagnose=>Inst Protocol.....	97
Diagnose=>Diagnose Monitor.....	98
Configuration.....	100
Configuration=>Alter Parameters.....	101
Configuration=>Alter Config.....	102
Configuration=>Load System Tables.....	103
Configuration=>Demo Support.....	104
Configuration=>National Language.....	105
Configuration=>Install Serverdb.....	107
Configuration=>Clear Serverdb.....	108
Troubleshooting when Problems Occur.....	109
System Crash.....	110
Log Space is Full.....	111
Database is full.....	112
Log Disk is Defective.....	113
System Error.....	114
Performance Analysis and Tuning.....	115
ADABAS Bottlenecks.....	116
The Course of Measured Values (x_wiztrc).....	117
Table Statistics and Structural Checks (xpu).....	118
Settings for NetTerm.....	120

Database Administration CONTROL (BC)

Purpose

You can use the **CONTROL** program to manage ADABAS for R/3 databases (up to Version 6.2.10).

Introduction

The program **CONTROL** is available to manage the database ADABAS for R/3 (in brief: ADABAS).

The ADABAS program **CONTROL** is used to control and monitor the ADABAS server and to execute the backup procedures.

CONTROL supports the following operations:

- Installing the database server
- System Table Loading
- Starting and shutting down the database server
- Starting and shutting down the remote SQL server
- Monitoring the database server
- Saving the database and log
- Restoring the database and log
- Expanding the disk capacities of the database server
- Diagnostic tool where support is required

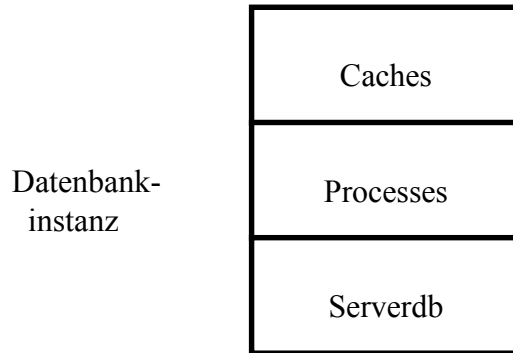
Remote access to non-local database servers by CONTROL is not yet supported. Note, however, that there are numerous terminal emulations that can easily be used to access UNIX servers from Windows PCs. For example, the [NetTerm \[Page 120\]](#) terminal emulation of InterSoft International Inc. is an easy-to-use tool for this purpose.

Overview

Overview

One or more instances of the database can be installed and operated on a computer.

Each database instance consists of processes, main memory structures (caches), and a disk-based database.



[Serverdb structure \[Ext.\]](#)

[Logging \[Ext.\]](#)

[Backup \[Ext.\]](#)

[Restart \[Ext.\]](#)

[Restore \[Ext.\]](#)

[Process structure \[Ext.\]](#)

[Caches \[Ext.\]](#)

[Multiprocessor Configurations \[Ext.\]](#)

[Client Server \[Ext.\]](#)

[Availability \[Ext.\]](#)

Using CONTROL

Prerequisites

You are connected to operating system level (UNIX, Windows NT).

Procedure

Enter the command `xcontrol`.

Result

The connect screen is displayed for entry of the following:

	Entry	Explanation
CONTROL USER	<controluser_name>	CONTROL user identification
PASSWORD	<password>	CONTROL user password
SERVERDB	<serverdb_name>	Name of serverdb

If the specified server database is not yet installed, the installation screen appears [Installing a New Serverdb \[Page 15\]](#).

Optional Procedure

Enter the command `xcontrol` with the option `-d <serverdb_name>` and / or `-u <controluser_name>, <password>`.

```
xcontrol [-d <serverdb_name>] [-u <controluser_name>, <password>]
```

CONTROL Menu Structure and User Guidance

CONTROL Menu Structure and User Guidance**Prerequisites**

CONTROL has been CALLED. A menu bar appears in which the following menu options are possible:

Operating..	Info..	Options..	Backup..	Diagnose..	Configuration..
<u>C</u> olors	<u>A</u> ctivity	<u>A</u> utorefresh..	<u>S</u> ave..	<u>O</u> p Messages	<u>A</u> lter Parameters..
<u>R</u> efresh..	<u>C</u> onfiguration	<u>R</u> emoteSQL Server..	<u>R</u> estore..	<u>C</u> ommand History	<u>A</u> lter Config..
<u>R</u> estart..	<u>K</u> ernel Param	<u>A</u> ccounting..	<u>S</u> how History	<u>I</u> nst Protocol	<u>L</u> oad System Tables
<u>S</u> hutdown..	<u>U</u> sers	<u>K</u> ernel Vtrace..	<u>S</u> how Protocol	<u>D</u> iagnose Monitor..	<u>D</u> emo Support
<u>U</u> ppdate Statistics	<u>C</u> aches		<u>M</u> edia Manager		<u>N</u> ational Language
<u>E</u> xit	<u>I</u> /O Accesses				<u>I</u> nstall Serverdb
	<u>L</u> ocks				<u>C</u> lear Serverdb
	<u>L</u> og				
	<u>L</u> ast UpdStatistics				
	<u>T</u> asks				
	<u>V</u> ersion				
	<u>H</u> elp				

Procedure

The menu bar can be activated in different ways.

CONTROL Menu Structure and User Guidance

- A menu option can be selected directly by pressing the highlighted letter (indicated by an underscore in the illustrations) and the **Ctrl** key at the same time. In most cases, selecting a menu option will display a pulldown menu.
- If the cursor is placed on an input field within the input screen, **F12** can be used to enable the menu bar. To reach the adjacent pulldown menu, use the **left / right** cursor keys.

A function of a displayed pulldown menu is activated either by positioning the cursor and pressing **Enter** or by selecting the highlighted letter (in this case, it is not necessary to press **Ctrl** at the same time).

If a help function is available, you can call it by pressing **F1**. 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**. Pressing **F3** returns to the previous screen.

If the release characters are not highlighted, the CONTROL presentation is not adapted to the terminal definition. In this case, **F2** can be used to change from within the Main Screen and all Installation screens to another presentation. ([color \[Page 13\]](#)).

In an input field, the rest of the line can be deleted starting from cursor position by pressing either **Ctrl+F3** or **Alt+F3**, depending on the terminfo file used. **End** moves the cursor to the end of the input field; **Home** moves the cursor to the beginning of the input field.

Result

Function key settings in the Main Screen

Function key	Setting
F1	Help
F2	Colors
F3	Exit
Enter	Ok
F9	Refresh

Function key settings in the Installation screens

Function key	Setting
F1	Explain
F2	Colors
F3	Cancel
F4	Print
F5, Enter	Next, Ok
F6	ReadConf
F7, Pgup	Prev
F8, Pgdn	Next

CONTROL Menu Structure and User Guidance

F9 , F10	Left (Devspace screen)
F9 , F11	Right (Devspace screen)

Function key settings in the Info screens

Function key	Setting
F2	In Pages / In KB
F3	Return
F4	Print
F6	Edit
F7 , Pgup	Prev
F8 , Pgdn	Next
F9	Refresh
F9 , F10	Left
F9 , F11	Right

Procedure: Print

CONTROL provides a print function in many screens. When the *print* function is used, the display is immediately output to the printer defined with the SET parameters ([Configuration → Alter Parameters → Set Defaults \[Ext.\]](#)).

Terminal and Color Settings in CONTROL

CONTROL provides a series of predefined presentations. A presentation defines how the attributes of screen elements used by CONTROL are to be displayed. For example, the cursor can be represented either with an inverse attribute, foreground and background colors or with the exchanged foreground/background colors and without inverse attribute. With the bold attribute, different combinations are also possible which do not look good on all terminals or are not visible at all.

As each presentation attempts to use another subset of possible properties, the supplied presentations cannot be distinguished from each other on all terminals. The differences in the presentations mainly concern the CONTROL function menus. With the selected presentation, the cursor should appear as a bar and the release characters of a menu function should be highlighted.

Prerequisites

Predefined presentations in CONTROL

DEFAULT	White on blue
HP9-SUN	White on blue
NETTERM	White on black
WHITE	Black on white
XTERM	Black on white
AIX	White on blue
BLACK	White on black

The presentation can be changed if the DEFAULT setting turns out to be not the appropriate one during installation.

Procedure when changing to another predefined presentation

- `color` (or `ctrl+0`) in the first Installation screen.
- additionally with `F2` in all other Installation screens and in the Main Screen.

If none of the predefined presentations meet the requirements, you can define your own new presentation.

Result

If you change the CONTROL presentation ([Configuration](#) → [Alter Parameters](#) → [Set Defaults \[Ext.\]](#), *SQL-PL Presentation*) or if you insert a new presentation, this presentation becomes the active presentation in CONTROL.

UNIX only: further options for setting the presentation

When using different terminals (or terminal emulations or terminal types) for the work with CONTROL, each terminal may require another presentation for best display. CONTROL remembers the last presentation used for the defined environment (`$DBTERM` or `$TERM`). This

Terminal and Color Settings in CONTROL

allows you to work with the appropriate presentation straight away, without having to set it manually, by defining the environment variable **\$DBTERM**.

You may also wish to work on different terminals but with the same terminal emulation (**\$DBTERM**) and activate different presentations. In this case, you can copy the terminal description file (e.g., `$DBROOT/terminfo/a/ansi`) with a new name (e.g., `$DBROOT/terminfo/m/my_ansi`) and set **\$DBTERM** to this name (**DBTERM=my_ansi**).

Installing a New Serverdb

Prerequisites

When CONTROL was called, a serverdb which does not yet exist was specified (that is, there is no parameter file with the database name).

Procedure

Install Serverdb <serverdb> on <servernode>

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 "Next"

Next
Prev
ReadConf
Color
Print
Cancel

- <serverdb>: Serverdb name is taken from the CONTROL call option -d or from the connect screen.
- <servernode>: Computer name within the network If the computer has no net card, the servernode name is local.
- CONTROL USER NAME: [CONTROL USER \[Ext.\]](#) is taken from the CONTROL call option -d or from the connect screen
PASSWORD: <password> for CONTROL USER
- SYSDBA NAME: <[SYSDBA \[Ext.\] Name](#)>
PASSWORD: <password> for SYSDBA
- DOMAIN USER NAME: <[DOMAIN_USER \[Ext.\] Name](#)>
PASSWORD: <password> for DOMAIN USER

The [OPERATOR \[Ext.\]](#) user is not defined within the Installation screen. The OPERATOR user initially has the password OPERATOR. You can subsequently modify the password by using [Configuration → Alter Parameters → Sysusers \[Ext.\]](#).



The names and passwords have a maximum length of 18 characters. Passwords must be entered twice to recognize input errors. Make sure that the SYSDBA name does not begin with **sys**, because this is a character combination reserved for SAP-internal use.

Installing a New Serverdb



If the screen attributes are not correctly set and the selections cannot be distinguished, you can use [color \[Page 13\]](#) or **F2** or **Ctrl+O** to change to another presentation of the screen.

When all specifications are complete, confirm the entries with *Enter*, *Pgdn* or *Next*.

Setting the database parameters

If necessary, set the [database parameters \[Ext.\]](#).

Creation of the database structure

Create your desired [database structure \[Ext.\]](#).

Starting the Installation

Start the [installation of the serverdb \[Ext.\]](#).

Result

If all installation steps have been performed without error, all lines on the screen end with *OK*. The [main screen of CONTROL \[Page 22\]](#) is displayed.

Step-by-step Installation of the Serverdb

The step-by-step installation is useful if for the time being only a partial installation is to be performed after which for example a data backup of another serverdb is to be loaded into the database or a check is to be made for the reason for an unsuccessful automatic installation (e.g. because of insufficient UNIX kernel parameters).

Prerequisites

The [database parameters \[Ext.\]](#) and the [database structure \[Ext.\]](#) have been created in accordance with your requirements. The entries were confirmed with *OK*.

Procedure

***** START INSTALLATION *****

Press "Install" to run the whole installation without interrupts.
Press "Stepwise" for installation with interactive interrupts.
Press "Restore" to install the SERVERDB from an existing datasave.
Press "Cancel" to go back to parameter changes.

Select *Stepwise*. After each installation step the result of the execution is displayed and the option to cancel or to continue the installation is provided.

INSTALL PARAMETERS

After installing the serverdb kernel parameters, the log file `xparam.prt` is output. Under UNIX, the minimum values of the configuration parameters of the operating system kernel are also displayed as a check.

[Example \[Ext.\]](#)

The operating system kernel may need adapting to these requirements. To do this, interrupt the installation procedure at this point, adapt the operating system kernel, and repeat the installation (*Configuration* → *Install Serverdb*).

START SERVERDB COLD & INIT CONFIGURATION

Please press "Continue" for INIT CONFIGURATION

If you press "Cancel", the installation is not complete -
to complete the installation start Install Serverdb again.

The serverdb is initialized with the selected configuration after it has been switched to the COLD mode.

Step-by-step Installation of the Serverdb

ACTIVATE SERVERDB or RESTORE

```
Please press "Continue" for ACTIVATE SERVERDB

Press "Restore" to install the SERVERDB from an existing
datasave.
If you press "Cancel", the installation is not complete -
to complete the installation, start Configuration/Activate
Serverdb or Backup/Restore from an existing datasave.
```

- Select *Restore* to read the information required for the installation from an existing data backup. Once data backup has been accessed, carry out a restart immediately if the database is consistent. There is now the option to include and restore logs or pages superimposed on the backup.
- Select *Continue* to activate the serverdb without loading data.

Afterwards the serverdb is in WARM mode.

LOAD SYSTEM TABLES

After ACTIVATE SERVERDB , the system tables have to be loaded. The system tables need not be loaded for a restore.

```
Please press "Continue" for LOAD SYSTEM TABLES

If you press "Cancel", the installation is not complete -
to complete the installation start Configuration/Load System
Tables.
```

Result

After the loading of the system tables, the installation is terminated.

Installation of the Serverdb using a Backup

Prerequisites

1. Identify the devspace needed for the data. The size of the data area can be taken from the original database configuration.

To reduce the size of the data area, calculate the minimum space required. To do this, determine the following:

- a) Maximum page number reached (xquery command: `SELECT maxdatapageno FROM SERVERDBSTATISTICS`)
- b) Actual volume of permanent data (xquery command: `SELECT usedperm FROM SERVERDBSTATISTICS`)

Size of data area reserved for the log.

When logs are being restored, the transactions that must be redone are copied to the data area so that the log area is left unchanged for a restart. To enable the data area to accept all the required log files, part of the data area is reserved for them and is therefore not available for permanent data objects. The parameter `RESERVED_REDO_SIZE` specifies the size of the reserved data area in 4 KB pages.

Add the volume of the permanent data and the size of the data area reserved for the log.

Find the maximum value from a) and b).

The size of the database should be at least this maximum value.

2. Ensure that the data devspaces are configured in such a way that the necessary space is available.
3. Start the installation of the serverdb from an existing data backup.

Procedure

The installation dialog provides the following three ways to use an existing data backup:

1. Loading the configuration and the data:

- Select *ReadConf* in one of the screens to set the [database parameters \[Ext.\]](#).
- Check the [database structure \[Ext.\]](#).
- Start screen

Installation of the Serverdb using a Backup



S

elect *Restore*.

The configuration parameters can still be adapted to requirements after loading.

2. Loading the data (without configuration):

Choose *Restore* on the initial screen.

The configuration parameters can no longer be adapted after loading.

3. Importing log or page backups

In special situations it may be necessary to redo log or page backups after restoring a data backup.



To restore the log information from version files, only specify the first version file to initiate the restore operation. CONTROL automatically continues restoring until it finds no more version files. If a version file is not found, CONTROL displays an I/O error. Choose *Continue* and then *End* to conclude the restore procedure. CONTROL puts the database into WARM mode.

Result

In all cases, the Media Manager is displayed to define the medium containing the data backup. The installation of the serverdb then takes place from the existing data backup in accordance with the option which you selected.

Update Installation

Prerequisites

CONTROL performs an update installation when it is started the first time after migrating from one version to the other (from 6.1 to 6.2 and within 6.2).

Procedure

- The kernel parameters are adapted to the new version by CONTROL. CONTROL message: Migration of Param File
- The system tables must be updated. For this purpose, select [Configuration → Load System Tables \[Page 103\]](#).

Result

CONTROL creates a copy of the parameter file (called UNIX: /usr/spool/sql/config/<serverdb>.old, Windows NT/Windows 95: %DBROOT%\CONFIG/<serverdb>.old). Should errors occur while loading the parameters, the operation can be undone by restoring the original parameter file. In this case, copy the file <serverdb>.old (copy of the parameter file) back to <serverdb> (active parameter file).



If an error occurs when restarting at the end of the update installation, inform the relevant Support organization.

The Main Screen

The Main Screen

Purpose

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

```

Operating..  Ino..  Options..  Backup..  Diagnose..  Configuration..
Usage Levels and Serverdb State                               at <Date> <Time>
0      10     20     30     40     50     60     70     80     90    100
|.....|.....|.....|.....|.....|.....|.....|.....|.....|.....|
*****|*****|*****|*****|*****|*****|*****|*****|*****|
Perm+Temp Data  910612 Pages      Data+Redo Space    1000678 Pages  49%
*****
                                           Log                7734 Pages   8%
*****
                                           Sessions           20           13%

-----
State..... WARM
Autosave Log.  ENABLED
Remote SQL.... ENABLED
Background.....NONE
Data Cache Hitrate.....100 %
Converter Cache Hitrate..... 98 %

<Servernode> : <Serverdb>

```

See also [CONTROL Menu Structure and User Guidance \[Page 10\]](#)

Features

Data fields

- <serverdb>: serverdb name
- <servernode>: name of the computer within the network where the database server is installed.
- <Date><Time>: date and time when the displayed data was determined.

Usage Levels (in Bar Form)

Data

describes the current usage level of the database. This measured value informs the administrator whether the physical disk space must be expanded soon. The first bar gives information about the usage level of the data area. The data area always has a reserve area (redo size) which is used when the log is restored. The quantity of permanent or temporary data generated by result tables is displayed (perm+temp data) and the usage level of the data area as a total.

The ADABAS server remains in **warm** mode when the entire data area is 100% full. Operation continues as soon as the disk capacity is increased.

Log

describes the current usage level of the log. The database server remains in **warm** mode if the log devspace and the reserve area in the data area are 100% full (see "Data"). The best log behavior can be obtained with an automatic log backup ([Backup → Save → Autosave Log \[Ext.\]](#)), which saves monitoring the log usage level. With an automatic log backup, the usage level decreases by the size of a log segment each time. Of course the log can also be saved manually ([Backup → Save → Log \[Ext.\]](#)).

Sessions

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 with [Configuration → Alter Parameters → Kernel \[Ext.\]](#) by modifying the [database parameter \[Ext.\]](#) MAXUSERTASKS.

Other Information

State

describes the current operating mode of the database server.

Operating modes of the ADABAS server and their meanings

Mode	Meaning
OFFLINE	The ADABAS server is not running. The database server kernel has not been started yet. Usually, the operator brings the database system directly into warm mode using Operating → Restart [Page 58] .
COLD	The ADABAS server has been started successfully. All system parameters which were changed using Configuration → Alter Parameters [Page 101] are now effective. Database activities are not possible. Some maintenance activities such as recovering the database, application of the log entries or modification of certain system parameters can only be performed in this mode.
WARM	This is the normal operating mode in which users can work with the ADABAS server.

Autosave Log

shows whether the automatic log backup is enabled.

Remote SQL

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

Background

shows whether a backup, a Verify Devspaces, or an Update Statistics is active in the background.

The Main Screen

Data Cache Hit Rate

shows the percentage hit rate of accesses to the data cache since the last serverdb start. The hit rate should be as close to 100% as possible. You should find out the cause of an unfavorable hit rate (under 99%).

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 [database parameter \[Ext.\]](#) CONVERTER_CACHE can be increased with [Configuration → Alter Parameters → Kernel \[Ext.\]](#).

Files used by CONTROL

CONTROL writes the following files to the **rundirectory** of the addressed serverdb. Before a rundirectory is defined during the installation, the files are written to \$DBROOT/wrk or %DBROOT%\WRK.

File	
control.dat	Encrypted user descriptions.
control.par	Information about the terminal presentations, the printer setting, and the defined system editor.
control.log	CONTROL command history
control.med	Defined backup media
control.ins	The log file of the last serverdb installation.
control.mig	The log file of the last migration.
control.bkp	Lock file to synchronize the save and restore requests.
control.bat	The log file of the last batch execution.
control.knl	The history of all backup and restore requests. This file is written by the database kernel.
control.utl	The logs of the previous internal utility requests (save and restore requests, add devspace, update statistics). The file is written by the database kernel.

For security reasons, copies are made of the most important files. These files are written to the **CONFIG directory** (UNIX: /usr/spool/sql/config; Windows NT/Windows 95: %DBROOT%\CONFIG).

The file name is defined from the serverdb name and extension.

File	
<serverdb_name>.dat	Copy of control.dat
<serverdb_name>.knl	Copy of control.knl
<serverdb_name>.utl	Copy of control.utl
<serverdb_name>.med	Copy of control.med

Some utility programs that are called by CONTROL write their files to the **current directory** (this is the directory valid when calling CONTROL). These files only have a temporary meaning and CONTROL removes them after their usage.

File	
load.pr(o)t	Log file of xload
Util*.pr(o)t	Log file of xutil
xparam.pr(o)t	Log file of xparam

Files used by CONTROL

<code>xpcheck.prc</code>	Log file of <code>xparam bcheck</code>
--------------------------	--

Backup Concepts and Strategies

CONTROL supports the backup and recovery procedures in a convenient way. There is always an automatic log backup when the log is filled up to the log segment size or to one third of its capacity. All backups can be activated interactively or with a batch call.

Ad-hoc Backups

In WARM and COLD database mode, CONTROL provides the following kinds of interactive backups under [Backup → Save \[Page 87\]](#):

- **Data migration** (physically complete),
- **Data recovery** (physically complete without checkpoint),
- **Pages migration** (physically incremental),
- **Pages recovery** (physically incremental without checkpoint),
- **Log** (logically incremental).

Automatic Backup of Log Segments

The oldest log segment can be automatically backed up in CONTROL as soon as it is full. For this purpose you can enable or disable the automatic log backup in CONTROL under the menu option [Backup → Save → Autosave Log \[Ext.\]](#). You should use a separate backup device for the automatic log backup. This device must be accessible to the backup process at any time

To protect against handling errors when using **tapes**, CONTROL checks whether any backup on the tape can be overwritten before saving the first log segment.

The automatic backup of log segments can be performed on **Norewind** tapes or version files. **Rewind** tapes, pipes, and external backup tools are not supported for the automatic backup of log segments.

[Backup media \[Page 28\]](#)

[Medium label \[Page 29\]](#)

[Examples of a backup scheme \[Page 31\]](#)

Backup Media

Backup Media

Purpose

One medium is assigned to each backup action. This medium can be a file, a tape, or a pipe.

Process flow

You define a backup medium with [Backup → Media Manager \[Page 92\]](#). It receives a name that can be selected freely under which the medium is used.

You can comprise media to form a group of *parallel media* and name them. Parallel media are simultaneously written or read by the database server. 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 assigned to a backup action like an individual backup medium.

Result

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 a tape is full. For this purpose, the media size must have been specified for the media definition or the tape device must be able to recognize the end of tape. A backup can be done to one backup medium as a minimum.

Backups performed using vendor-specific backup tools

Several external backup tools are supported, for example, ADSTAR Distributed Storage Manager). Special names must be used for a backup medium (see [external backup tools \[Page 47\]](#)).

CONTROL does not require any operator intervention during the backup. The sequence of the tapes for a recovery is defined by the vendor-specific backup tool.

If the group of parallel media is denoted by a predefined name appropriate for the external backup tool and the individual media definitions contain the path of the pipe, then you can also perform a "parallel" backup to vendor-specific backup tools.

Medium Label

Purpose

ADABAS automatically gives each backup an identifying label. This label contains the type of backup (complete or incremental as data or log backup) and a sequence number, whereby the data backups and log backups are numbered separately.

[Example \[Ext.\]](#)

Prerequisites

The medium label uniquely identifies the backups done since the installation of the database. If the backup medium is a tape or a cassette, you should write the medium label on the sticker of the tape or cassette at the end of the backup.

Process flow

The order in which the log backups must be restored can be seen in the sequence number. The sequence number does **not** indicate which log backup must be restored first after a data backup. For this purpose, read the log of the backup containing the log page numbers which determine the suitable log backup for the data backup.

Before you start an immediate (ad hoc) backup to a tape, you should always read the label of the backup that may already be on the tape.

For a backup started immediately (ad hoc), the label is displayed in the protocol file at the end of the backup. When a recovery is performed, the label is displayed for confirmation before the restore procedure starts; it is also included in the log files in which each save and restore action is recorded. Later, after concluding the backup, the label can be read at any time by means of the Media Manager ([Backup → Media Manager \[Page 92\]](#)).



You should design your own backup concept covering several generations. A backup generation is a data backup (*Save → Data*) with subsequent log or page backups.

A tape containing a complete data backup should therefore not be overwritten immediately with the next *Save → Data*. If you retain for example the last four backup generations, it may be possible to use an older backup if a media failure occurs.

You should always use the same tape for each type of save ([example for sticker \[Ext.\]](#)). Of course, you must take care that the tapes and cassettes are not used more times than is recommended by the manufacturer.

Formal Description of the Backup Labels

```
<medium_label>      ::= <save_kind>_<sequence_no>
<save_kind>         ::= DAT | PAG | LOG
<sequence_no>      ::= <number> (DAT and PAG share a sequence number
                                LOG has a separate sequence number)
```

Medium Label

Examples of a Backup Scheme

Example 1

The recommended backup scheme is a complete backup once a week (on Saturday) and one or more log backups each working day. A new generation of backup tapes begins each week with the complete backup. If you are working with four backup generations, the tape with label DAT_1 can be overwritten with backup DAT_5 in the fifth week (with appropriate log backups).

1. Number of backup generations: **four**
2. **One Save** → *Data* per week
3. Regular or automatic log backups (Autosave Log, AUTOON) between the complete backups (*Save* → *Data*). Note that Autosave Log requires a tape device of its own.
4. *Save* → *Pages* in special real-life situations only

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1 st Wk.	DAT_1		LOG_1	LOG_2	LOG_3	LOG_4	LOG_5
2 nd Wk.	DAT_2		LOG_6	LOG_7	LOG_8	LOG_9	LOG_10
3 rd Wk.	DAT_3		LOG_11	LOG_12	LOG_13	LOG_14	LOG_15
4 th Wk.	DAT_4		LOG_16	LOG_17	LOG_18	LOG_19	LOG_20
5 th Wk.	DAT_5		LOG_21	LOG_22	LOG_23	LOG_24	LOG_25

Example 2

The recommended backup scheme consists of one complete backup per week (on Saturdays) and one page backup per work day. The automatic backup of log segments is entered once in this example in order to be complete.

The page backups are recommended in addition to the log segment backups, because, in most cases, they ensure considerably faster recovery times than the log segment backups.

If you are working with four backup generations, the tape with label DAT_1 can be overwritten with backup DAT_25 in the fifth week (with appropriate page backups).

1. Number of backup generations: **four**
2. **One Save** → *Data* per week
3. Automatic log backups (Autosave Log, AUTOON) between the full backups (*Save* → *Data*). Note that Autosave Log requires a tape device of its own.
4. *Save* → *Pages* daily - preferably at the end of the working day.

Examples of a Backup Scheme

	Sat	Sun	Mon	Tue	Wed	Thu	Fri
1 st Wk.	DAT_1	AUTO -ON	PAG_2	PAG_3	PAG_4	PAG_5	PAG_6
2 nd Wk.	DAT_7		PAG_8	PAG_9	PAG_10	PAG_11	PAG_12
3 rd Wk.	DAT_13		PAG_14	PAG_15	PAG_16	PAG_17	PAG_18
4 th Wk.	DAT_19		PAG_20	PAG_21	PAG_22	PAG_23	PAG_24
5 th Wk.	DAT_25		PAG_26	PAG_27	PAG_28	PAG_29	PAG_30

Backup/Restore - a Practical Guide

[Saving to a single medium \[Page 34\]](#)

[Recovering from a single medium \[Page 35\]](#)

[Saving to a medium with continuation media \[Page 36\]](#)

[Recovering from a medium with continuation media \[Page 37\]](#)

[Saving to several parallel media without continuation media \[Page 38\]](#)

[Recovering from several parallel media without continuation media \[Page 39\]](#)

[Saving to several parallel media with continuation media \[Page 40\]](#)

[Recovering from several parallel media with continuation media \[Page 41\]](#)

[Examples of backup/restore \[Page 43\]](#)

Saving to a Single Medium

Saving to a Single Medium

Prerequisites

The simplest backup procedure is where you only require one medium. This is only possible if the capacity of the medium is sufficient.

Interactive Procedure

1. [Select the save action \[Ext.\]](#)
2. [Select medium \[Ext.\]](#)
3. [Check inputs \[Ext.\]](#)
4. [Start backup \[Ext.\]](#)

Batch Call

[Syntax \[Ext.\]](#)



```
xbackup -a SAVEDATA -d mydb -m DAT90
```

Recovering from a Single Medium

Prerequisites

The backed-up data is contained in just one medium.



If the `control.knl` file in the rundirectory of the database is not damaged, CONTROL will support the recovery ([supported recovery \[Page 51\]](#)).

Interactive Procedure

1. [Select the restore action \[Ext.\]](#)
2. [Select medium \[Ext.\]](#)
3. [Check inputs \[Ext.\]](#)
4. [Start restore \[Ext.\]](#)

Batch Call

[Syntax \[Ext.\]](#)



```
xrestore -a SAVEDATA -d mydb -m DAT90
```

Saving to a Medium with Continuation Media

Saving to a Medium with Continuation Media

Prerequisites

The backups *Save → Data* and *Save → Pages* to media and continuation media are only useful if only one output device is available.



Where possible, the backup should always be done to several parallel media rather than to one medium with continuation media. Parallel backups are **not** possible with log backups.

Interactive Procedure

1. [Select the save action \[Ext.\]](#)
2. [Select medium \[Ext.\]](#)
3. [Check inputs \[Ext.\]](#)
4. [Start backup \[Ext.\]](#)
5. [Insert the continuation media and continue the backup operation \[Ext.\]](#)

Batch Call

No batch call is provided for backing up to several media in succession. Where possible, use external backup tools or autoloaders.

Recovering from a Medium with Continuation Media.

Prerequisites

The backed-up data is contained in several media.



If the `control.knl` file in the `rundirectory` of the database is not damaged, CONTROL will support the recovery ([supported recovery \[Page 51\]](#)).

Interactive Procedure

5. [Select the restore action \[Ext.\]](#)
6. [Select medium \[Ext.\]](#)
7. [Check inputs \[Ext.\]](#)
8. [Start restore \[Ext.\]](#)
9. [Insert the continuation medium and continue the restore operation \[Ext.\]](#)

Batch Call

No batch call is provided for recovering from several media in succession. Where possible, use external backup tools or autoloaders.

Saving to Parallel Media without Continuation Medium

Saving to Parallel Media without Continuation Medium

Prerequisites

Saving to parallel media is only possible for data and page backups. No continuation media are needed if the backup fits on to the parallel backup media.

Interactive Procedure

1. [Select the save action \[Ext.\]](#)
2. [Select parallel medium \[Ext.\]](#)
3. [Check entry \[Ext.\]](#)
4. [Start backup \[Ext.\]](#)

Batch Call

[Syntax \[Ext.\]](#)



```
xbackup -a SAVEDATA -d mydb -m TAPES_4
```

Recovering from Parallel Media (Without Continuation Media)

Prerequisites

The backed-up data is contained in several parallel media.



If file `control.knl` in the `rundirectory` of the database is not damaged, CONTROL will support the recovery ([supported recovery \[Page 51\]](#)).

Interactive Procedure

1. [Select the restore action \[Ext.\]](#)
2. [Select parallel medium \[Ext.\]](#)
3. [Check inputs \[Ext.\]](#)
4. [Start restore \[Ext.\]](#)

Batch Call

[Syntax \[Ext.\]](#)



```
xrestore -a SAVEDATA -d mydb -m TAPES_4
```

Saving to Parallel Media with Continuation Media

Saving to Parallel Media with Continuation Media

Prerequisites

Saving to parallel media is only possible for data and page backups. Continuation media are requested if the backup does not fit on to the parallel backup media.

Procedure

5. [Select the save action \[Ext.\]](#)
6. [Select parallel medium \[Ext.\]](#)
7. [Check inputs \[Ext.\]](#)
8. [Start backup \[Ext.\]](#)
9. Insert the continuation media and continue the backup operation.

If the capacity of the parallel media is not sufficient for the backup, **all** media which are defined as parallel media can have a continuation medium. For this case, the "number of media used for the last save" will be explained in greater detail. Three examples are given in each of which the parameter is set differently in step 3. It is assumed for all three examples that six media are sufficient for the backup.

After selecting the medium named `TAPES_4` as the medium to be used for *Save → Data* (step 2), CONTROL displays the [List of parallel media \[Ext.\]](#) for confirmation. Continue as shown in the examples:

[Example 1 \[Ext.\]](#)

[Example 2 \[Ext.\]](#)

[Example 3 \[Ext.\]](#)

Result

No batch call is provided for saving to parallel media with continuation media. Where possible, use external backup tools or autoloaders.

Recovering from Parallel Media (With Continuation Media)

Prerequisites

The backed-up data is contained in several parallel media and continuation media.

The number of media to be loaded is greater than the number of parallel tape devices available.

The number of tapes to be loaded is known when recovering.



If file `control.knl` in the `rundirectory` of the database is not damaged, CONTROL will support the recovery ([supported recovery \[Page 51\]](#)).

Procedure

1. [Select the restore action \[Ext.\]](#)
2. [Select parallel medium \[Ext.\]](#)

After selecting the medium named `TAPES_4` as the medium to be used for *Restore → Data* (step 2), CONTROL displays the [List of parallel media \[Ext.\]](#) for confirmation.

3. [Check inputs \[Ext.\]](#)

Enter the number of media to be loaded as the number of media. The parameter is important now, because the number of parallel tape devices is not sufficient to load all the media simultaneously.

It is not absolutely vital that you specify the correct number of media. However, when tapes are inserted, handling will be easier if you specify the exact number of media.

CONTROL displays the [label information \[Ext.\]](#). These labels should match the texts on the stickers of the tapes, and they should all be the same.

4. [Start restore \[Ext.\]](#)

As in this case the number of parallel tape devices is not sufficient for the number of tapes to be loaded, CONTROL expects [continuation media \[Ext.\]](#).

5. Insert the continuation media and continue the restore operation.

- Continuation medium **tape**: insert the next **tape**. Select *OK* or **Enter**.
- Continuation medium **file**: press **F12** and specify the path name of the next file. Choose *OK*.

You can select another type of medium as the continuation medium.



Under `Device` and `Type`, the screen displays the specifications that were valid for the previous medium. You can change these specifications with **F12**.

CONTROL displays the label information read from the specified continuation medium.

Select *OK* to continue the restore operation.

Recovering from Parallel Media (With Continuation Media)

Repeat this step as often as necessary to complete the restore operation or until it is canceled. In this example, step 5 is performed exactly twice.



Select *Ignore* if the tape device is to be ignored. All the tapes to be inserted are then only expected from the other tape devices. This can be useful, for example, if you only want to load the rest of the tapes on particular tape devices and no longer on all defined parallel devices.

Batch Call

No batch call is provided for recovery from parallel media with continuation media. Where possible, use external backup tools or autoloaders.

Examples of Backup/Restore

The following illustration shows a data backup cycle, in which three points in time for the occurrence of disk failure are marked in the data area for the following recovery examples. Examples of tape labels which CONTROL issues to identify the individual save actions are given in parentheses on the right.

```

          SAVE DATA      (1)                (DAT_00155)
Disk failure A.....
          SAVE LOG (1)  (LOG_00231)
          SAVE LOG (2)  (LOG_00232)
          SAVE LOG (3)  (LOG_00233)
          SAVE PAGES   (1.1)                (PAG_00156)
          SAVE LOG (4)  (LOG_00234)
          SAVE LOG (5)  (LOG_00235)
          SAVE LOG (6)  (LOG_00236)
          SAVE PAGES   (1.2)                (PAG_00157)
          SAVE LOG (7)  (LOG_00237)
          SAVE LOG (8)  (LOG_00238)
Disk failure B.....
          SAVE LOG (9)  (LOG_00239)
          SAVE DATA   (2)                (DAT_00158)
Disk failure C.....
          SAVE LOG (10) (LOG_00240)
          SAVE LOG (11) (LOG_00241)
Disk failure D
    
```

If the log devspace is also damaged, database recovery in the following three examples requires the presence of at least one intact log devspace which can be used in log mode DUAL or NORMAL to recover the defective log devspace ([Backup → Restore → Devspace \[Ext.\]](#)).

Examples of Backup/Restore

Recovery following Disk Failure A

To recover the database after disk failure A, only the first backup version of the data devspace must be loaded using [Backup → Restore → Data \[Ext.\]](#). The following restart then supplements the data devspace by redoing the transactions recorded in the log devspace.

```
restore data (1) (DAT_00155)
restart
```

Recovery following Disk Failure B

When disk failure B occurs, there are several ways of recovering the serverdb. The quickest method of recovery consists in restoring the database using [Backup → Restore → Data \[Ext.\]](#) and subsequently reloading the modified pages 1.1 and 1.2 using [Backup → Restore → Pages \[Ext.\]](#). Finally, both backup versions of log segments 7 and 8 must be redone using [Backup → Restore → Log \(UNTIL\) \[Ext.\]](#).

In the DUAL log mode, the backup version of a log segment is implicitly copied onto both log devspaces. Restart ends the recovery.

1. Restore variant:

```
restore data (1) (DAT_00155)
restore pages (1.1) (PAG_00156)
restore pages (1.2) (PAG_00157)
restore log (Log 7) (LOG_00237)
restore log (Log 8) (LOG_00238)
restart
```

Older backup versions of the log segments can also be used for the recovery of the serverdb.

2. Restore variant:

```
restore data (1) (DAT_00155)
restore pages (1.1) (PAG_00156)
restore log (Log 4) (LOG_00234)
...
restore log (Log 8) (LOG_00238)
restart
```

3. Restore variant:

```
restore data (1) (DAT_00155)
restore log (Log 1) (LOG_00231)
...
restore log (Log 8) (LOG_00238)
restart
```

Recovery following Disk Failure C

When disk failure C occurs, the serverdb can be recovered in the following way:

Only the last backup version of the data devspace is restored. If this version is not readable for any reason, older data backup versions can be restored (analogous to the last examples) which then require that the corresponding log segments be redone.

1. Restore variant:

```
restore data      (2)          (DAT_00158)
restart
```

2. Restore variant:

```
restore data      (1)          (DAT_00155)
restore pages     (1.1)        (PAG_00156)
restore pages     (1.2)        (PAG_00157)
restore log       (Log 7)      (LOG_00237)
restore log       (Log 8)      (LOG_00238)
restore log       (Log 9)      (LOG_00239)
restart
```

The same procedure must be used if organizational reasons require an older database state to be restored [Backup → Restore → Log \(UNTIL\) \[Ext.\]](#) can be used to select the point in time of the desired database state.

Recovery following Disk Failure D



Example 1: a restored log segment is not suitable for recovering a database.

Load the most recent complete backup for the recovery, and then the log segments. Proceed according to the backup protocol, loading the log segments in the order of their creation.

```
restore data      (2)          (DAT_00158)
restore log       (Log 10)     (LOG_00240)
--> "-8003 Log and Data must be compatible"
restore log       (Log 11)     (LOG_00241)
```

Example 1 shows that the log backup (10) could have been completed prior to the full backup DAT_00158. For this reason, error -8003 is reported when this log segment is recovered after the full backup DAT_0158. The error can be ignored and the next log backup (see backup protocol) LOG_00241 restored.



Example 2: a restored log segment is not suitable for recovering a database.

See "Recovery following Disk Failure B" 1. Restore variant

Examples of Backup/Restore

```
restore data      (1)                (DAT_00155)
restore pages    (1.1)              (PAG_00156)
restore pages    (1.2)              (PAG_00157)
restore log      (Log 1)            (LOG_00231)
--> "-8003 Log and Data must be compatible"
restore log      (Log 7)            (LOG_00237)
restore log      (Log 8)            (LOG_00238)
restart
```

In example 2 the first log backup LOG_00231 is accidentally loaded instead of log backup 7. The history information for the log page numbers shows that LOG_00231 cannot follow PAG_00157. The error message -8003 Log and Data must be compatible appears. In this case you must continue with the correct log backup 7. The backup protocol shows which log backup should be restored next.

External Backup Tools

Prerequisites

External backup tools can be used to save to tapes and continuation tapes without operator intervention. CONTROL provides a connection to several external backup tools.

Currently, the following external backup tools are supported:

[ADSM \(IBM\) \[Ext.\]](#)

[FlashNet \(GymData\) \[Ext.\]](#)

OmniBack (HP)

[Networker \(Legato\) \[Ext.\]](#)

[NetVault \(NCR\) \[Ext.\]](#)

Procedure

An **interactive procedure is not provided for**. With external backup tools only [backup/recovery in batch mode \[Page 52\]](#) is possible.

[Notes on timing \[Ext.\]](#)

External backup tools are addressed by special media names.



```

Save Data
Medium.....: ADSM
Device Type.....: P           Parallel.....:
Overwrite (Y/N/V): N           Media Size in Pages: 0

Device: /pipe/to_adsm

```

Media for the Use of External Backup Tools

The following list of archiver tools may change. To see the list valid for the current state of `xbackup` or `xrestore`, specify the parameters `-h` (Help) or `-v` (Version) when calling `xbackup` or `xrestore`.

The following strings as the start of a medium name will cause special treatment. In all these cases, data transfer will be via named pipes, so the `Device Type` must be "P" and the `Media Size` 0:

"EXTERN"

The external backup tool has control. `xbackup` or `xrestore` is called from an **external backup tool**. Data transfer is via a named pipe which must already be available. The backup tool (or its administrator) is responsible for the proper identification of the backup version and labeling of the

External Backup Tools

tape. This medium name must not be the name of a group of parallel media. The pipe name must be passed as parameter `-f`; it is checked against the medium definition.

Under Windows NT for example the backup tool **NetVault** (NCR) is used to store database backups. This tool calls `xbackup` or `xrestore` using the medium name **EXTERN** ([Notes on the medium name EXTERN \[Ext.\]](#)).

ADSM, FNET, OMNI

`xbackup` or `xrestore` has control. `xbackup` or `xrestore` calls a specific backup tool to load or write the data. Data transfer is via named pipes which must not exist when `xbackup` or `xrestore` is called.

The tool called is [ADSM \(IBM\) \[Ext.\]](#), [FlashNet \(GymData\) \[Ext.\]](#) or **OmniBack** (HP). Details for the connection of these tools may change at any time; more tools can be added (see the current README file).

On backup, a string containing machine name, database name, date, time and type of backup is automatically generated and used to **identify** the backup version in the backup tool. When the recovery is performed, the user is shown (on standard output) a numbered list of all backup versions available in the backup tool, the identifiers of which match the machine name and database name. The user must **select** the desired backup version from this list.

If the medium belongs to a group of parallel media, several instances of the backup tool will be started in parallel. The number of media of a group at recovery must be the same as at backup.

NSR

`xbackup` or `xrestore` has control. `xbackup` or `xrestore` calls the [NetWorker \(Legato\) \[Ext.\]](#) to load or write the data. Data transfer is via named pipes which must not exist when `xbackup` or `xrestore` is called.

As NetWorker only supports file names as identifiers, the backup is only identified by pipe name, date and time. On recovery, selection is done from a numbered list of all backups available using this pipe. Parallel media cannot be used.

Batch Call

[Syntax xbackup \[Ext.\]](#)



```
xbackup -a SAVEDATA -d mydb -m ADSM
```

Recovering with the aid of external backup tools is done like a backup. You use the same medium definition as for the backup and perform the restore operation as from a single medium.

[Syntax xrestore \[Ext.\]](#)

Autoloader under Windows NT

Prerequisites

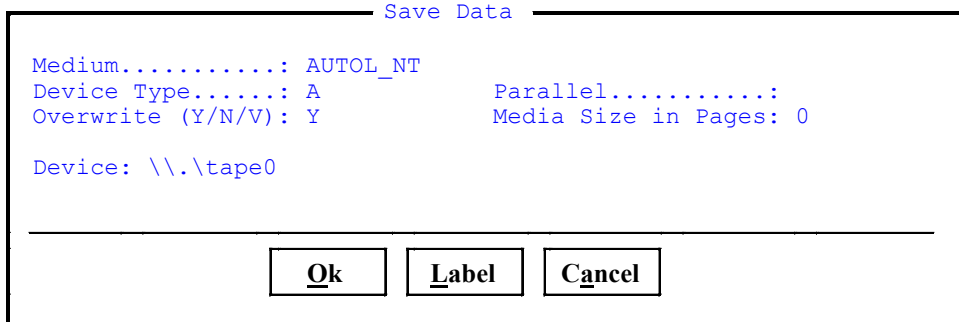
In Windows NT, saving to tapes with continuation media without operator intervention is also possible using an autoloader. To control the autoloader under Windows NT, specify "A" as Device Type ([media editor \[Ext.\]](#)).

When reaching the end of tape, the autoloader is addressed which will use the next tape available. The backup will only be successful if the number of tapes is sufficient for it.

Interactive Procedure

The **interactive procedure** is exactly the same as for [saving to a medium without continuation media \[Page 34\]](#).

Example of medium definition:



```
Save Data
Medium.....: AUTOL_NT
Device Type.....: A           Parallel.....:
Overwrite (Y/N/V): Y           Media Size in Pages: 0

Device: \\.\tape0

Ok Label Cancel
```

At the end of the backup, the tape device shows the number of tapes written. The label displayed and confirmed at the start of the backup (step 3) should be placed on the stickers of the tapes.

Before backing up, it is imperative that you check the write protection of all tapes, because otherwise the entire backup procedure will be aborted.

After inserting the tape cartridge in the autoloader, the first tape must be selected manually.

Batch Call

[Syntax xbackup \[Ext.\]](#)



```
xbackup -a SAVEDATA -d mydb -m AUTOL_NT
```

Recovering with the aid of an autoloader under windows NT is done like a backup. You use the same medium definition as for the backup and perform the restore operation as from a single medium. To facilitate a successful restore, you must ensure that all tapes belonging to the backup have been inserted in the autoloader.

[Syntax xrestore \[Ext.\]](#)

Other Autoloaders

Other Autoloaders

Prerequisites

Saving to tapes and continuation tapes without operator intervention is also possible in non-Windows NT operating systems using an autoloader.

For this purpose, CONTROL provides an option to formulate an explicit operating system command to initiate the change of tape. To do this, specify L as `Device Type` when defining the medium to be used. Then you can use `OsCmd` to define a command that will initiate the change of tape.

Interactive Procedure

The **interactive procedure** is exactly the same as for [saving to a medium without continuation media \[Page 34\]](#).

Example of the medium definition:

```
Save Data
Medium.....: AUTOL
Device Type.....: L           Parallel.....:
Overwrite (Y/N/V): Y         Media Size in Pages: 0

Device: /dev/rmmmt0
OS Cmd: mt -f /dev/rmt0 rewoffl
```

At the end of the backup, the tape device shows the number of tapes written. The label displayed and confirmed at the start of the backup (step 3) should be placed on the stickers of the tapes.

Batch Call

[Syntax xbackup \[Ext.\]](#)



```
xbackup -a SAVEDATA -d mydb -m AUTOL
```

Recovering with the aid of an autoloader is done like a backup. You use the same medium definition as for the backup and perform the restore operation as from a single medium. To facilitate a successful restore, you must ensure that all tapes belonging to the backup have been inserted in the autoloader.

[Syntax xrestore \[Ext.\]](#)

Supported Recovery

Prerequisites

The recovery dialogue supported by CONTROL is usual. It can take place when all the history data of the backups is available, that is when the `control.knl` file is not damaged. If the `control.knl` file is no longer available (for example in the case of disk failure), proceed as described in the [Backup/Restore Guide \[Page 33\]](#).

Interactive Procedure

1. Select the [restore activity \[Ext.\]](#) *Backup* → *Restore* → *Data*. CONTROL then displays all the complete backups known from the past. Select the backup you want.
2. CONTROL now displays the medium information read. Before confirming this information, you must insert the tape in the selected tape device and start [reading the medium \[Ext.\]](#).
3. [Information \[Ext.\]](#) about the selected backup is output, including the backup label. You should check whether the inserted medium is the one you want. Then start the restore procedure.
4. After ending *Restore* → *Data*, CONTROL offers the option to continue with [Restore Pages or Restore Log \[Ext.\]](#) or *Restart UNTIL*.
5. Response of CONTROL
 - a) With [Restore Pages \[Ext.\]](#), CONTROL displays the media definition of the next page backup to be imported. Select *Ok* to start the recovery procedure of the first of these backups. You can then select *Restore Log* or *Restore Pages* again.
 - b) With [Restore Log \[Ext.\]](#), CONTROL displays the media definition of the log backup to be imported first. Choose *Ok* to start the automatic recovery of all log segments.

After choosing *Restore* → *Data* or *Restore Pages*, the restore operation can be terminated by *Cancel*. You can then select *Restore Log* or *Restore Pages* again.

Batch Mode: xbackup/xrestore

Batch Mode: xbackup/xrestore

CONTROL provides an interface for batch operations.

The `xbackup` and `xrestore` functions are provided primarily to combine the CONTROL backup functions with vendor-specific backup tools. Two variants are possible:

- The `xbackup` and `xrestore` functions are called under the user interface and thus under the control of the external backup tool.
- `xbackup` or `xrestore` calls the external backup tool.

It is also possible to use the `xbackup` and `xrestore` functions as genuine batch interface. They also provide further options in addition to the backup and recovery functions:

[Functionality and parameters \[Ext.\]](#)

[Actions \[Ext.\]](#)

[Errors \[Ext.\]](#)

[Standard input and output \[Ext.\]](#)

[xbackup/xrestore exit codes \[Ext.\]](#)

[Files \[Ext.\]](#)

Batch Mode: xcontrol

Syntax

```
xcontrol -d <dbname> -u <cnid>,<cnpw> -b <operation> [<filename>]
```



```
xcontrol -d dbdemo -u CN,CN -b loadsystab
```

Parameters for xcontrol

Parameter	
-d <dbname>	The name of the database. Default setting: environment variable <code>SERVERDB</code>
-h	"Help": only information about usage of the particular interface and the names of the special media are given (no operation).
-b <operation>	The operation to be executed (not case sensitive).
-u <cnid>,<cnpw>	The name and password of the CONTROL user.
-v	Version message of <code>xcontrol</code> . No action is started.

Operations

RESTOREDEV	Backup → Restore → Devspace [Ext.]
LOADSYSTAB	Configuration → Load System Tables [Page 103]
WARM	Operating → Restart → Warm [Ext.]
COLD	Operating → Restart → Cold [Ext.]
	Operating → Shutdown → Cold [Ext.]
OFFLINE	Operating → Shutdown → Offline [Ext.]

[xcontrol exit codes \[Ext.\]](#)

Menu Functions

Menu Functions

Features

[Operating \[Page 55\]](#)

[Info \[Page 63\]](#)

[Options \[Page 81\]](#)

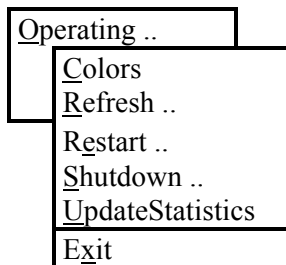
[Backup \[Page 86\]](#)

[Diagnose \[Page 94\]](#)

[Configuration \[Page 100\]](#)

Operating

Features



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

[Operating → Colors \[Page 56\]](#)

[Operating → Refresh \[Page 57\]](#)

[Operating → Restart \[Page 58\]](#)

[Operating → Shutdown \[Page 59\]](#)

[Operating → Update Statistics \[Page 60\]](#)

[Operating → Exit \[Page 62\]](#)

Operating=>Colors

Operating=>Colors

Features

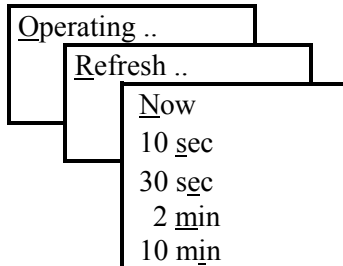
This menu function can be used to modify screen attributes and color settings. It displays several presettings for selection (examples: DEFAULT, AIX, BLACK, NETTERM, HP9_SUN und WHITE). On some terminals, you cannot recognize the release letters of the individual menu items with the "DEFAULT" setting. In this case, you should select the "AIX" setting with inverse representation of the release letters.

In the Main Screen and the Installation Screens, you can also call this function by pressing **F2**.

More color settings can be defined in the warm serverdb mode using the function [Configuration → Alter Parameters → Set Defaults \[Ext.\]](#). Further characteristics are described in [terminal and color settings in control \[Page 13\]](#).

Operating=>Refresh

Features



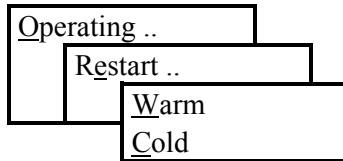
This menu function updates the values of the Main Screen.

An automatic refresh can be started at intervals of 10 sec, 30 sec, 2 min, or 10 min. No other CONTROL actions are possible during the periodic refresh of the Main Screen. The function can be cancelled with **Ctrl+C**.

Operating=>Restart

Operating=>Restart

Features



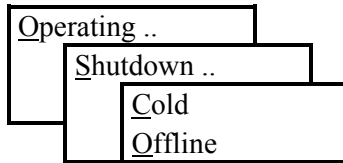
The *Restart* menu function starts the database server from offline mode into the cold or warm operating mode. If the warm operating mode is not reached after activating the *Restart* menu function, you can attempt to find the possible error cause using [Diagnose → Op Messages \[Page 95\]](#). At each start of the severdb with the menu functions *Operating → Restart → Warm* or *Operating → Restart → Cold*, a copy of the console protocol (`knldiag` file in the `rundirectory` of the `serverdb`) is generated under the name `knldiag.old` and the file `knldiag` is deleted.

[Operating → Restart → Warm \[Ext.\]](#)

[Operating → Restart → Cold \[Ext.\]](#)

Operating=>Shutdown

Features



The **Shutdown** menu options switch the database server from warm to cold mode or stop it. The latter case is displayed as offline mode.

A normal *Shutdown* fails when the “Autosave Log” process is active at that time. An error message informs the user that the “Autosave Log” process must be disabled before the *Shutdown* can be performed. *Shutdown Quick*, on the other hand, stops the Autosave process without a message.

[Operating → Shutdown → Cold \[Ext.\]](#)

[Operating → Shutdown → Offline \[Ext.\]](#)

Operating=>Update Statistics

Operating=>Update Statistics

Features

This function updates the statistical information of the database. This includes the number of table entries, the size of tables and indexes, and the value distribution (distinct values) of indexes or columns; it stores the statistical information in the database catalog.

Use

The ADABAS optimizer needs these specifications to determine the best processing strategy for complex SQL statements. If the sizes or value assignments have changed considerably in the database, a new *Update Statistics* is required. We recommend that you perform an *Update Statistics* once a week.

If ADABAS ascertains 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*. As this attempt can be aborted because of conflicting locks, it does not completely replace the explicit *Update Statistics*.

The operation can be applied to particular tables, particular columns or to all base tables of the database server.

Actions

After selecting the *Update Statistics* menu function, the following screen appears:

```
Update Statistics
User Name      : *
Table Name     : *
Column Name    : *

Estimate in Rows : 1000

Ok      Last Upd  Cancel
```

The parameters define the data which the update statistics function is to refer to.

User Name

If * is specified, the function is performed for the complete dataset, which can take some time. It is possible to specify any arbitrary wildcard arguments. If a particular table name is specified, a particular user name must also be defined in the `User Name` field. Likewise, if a particular `Column Name` is used, the specification of `Table Name` and the `User Name` is also required.

Estimate in Rows

You can define the amount of table data for a random sample. The Update Statistics function is only applied to the specified number of rows, which accelerates execution. The quality of the random sample depends on the size of the table and the physical position of the data. Measurements show that good results and fast response times can be achieved for large tables when the Update Statistics function is applied to random samples comprising 1000 to 5000 rows.

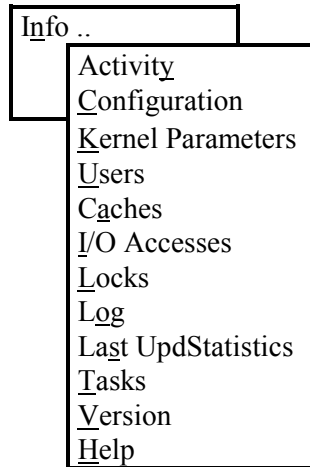
Operating=>Exit

Operating=>Exit

This function is used to leave CONTROL. If the database server is in cold operating mode, you can shut down the database using *STOP* (Offline). All active database processes are terminated.

Info

Features



You can call the following information about the cache with the functions of the Info menu.

- the 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
- the version identification of ADABAS

In the display screen, it is possible to page down using the *Next* function and to page up using the *Previous* function. *Refresh* updates the output. *Return* closes the display screen and returns you to the Main Screen of CONTROL.

In some Info screens, you can display the sizes in the corresponding format by using *In Pages* or *In KB*. This setting is then also valid for the Main Screen until it is reset.

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. However, a detailed interpretation of these outputs is usually only needed for support purposes.

[Info → Activity \[Page 65\]](#)

[Info → Configuration \[Page 66\]](#)

[Info → Kernel Parameters \[Page 68\]](#)

Info

[Info → Users \[Page 69\]](#)

[Info → Caches \[Page 70\]](#)

[Info → I/O Accesses \[Page 72\]](#)

[Info → Locks \[Page 73\]](#)

[Info → Log \[Page 75\]](#)

[Info → Last UpdStatistics \[Page 77\]](#)

[Info → Tasks \[Page 78\]](#)

[Info → Version \[Page 79\]](#)

[Info → Help \[Page 80\]](#)

Info=>Activity

Features

This menu function informs you about the most important events and activities of the ADABAS server. The following information is displayed in summarized form:

- 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
- the activities of the logging process

[ADABAS-Caches \[Ext.\]](#)

[I/O Activities \[Ext.\]](#)

Info=>Configuration

Info=>Configuration

Features

The displayed values are composed of the parameters for the data devspaces, the log devspaces, and the cache areas. If modifications to these values are necessary, they can be made using the [Configuration→Alter Parameters \[Page 101\]](#).



Serverdb State	
Unicode	NO
Accounting	OFF
Monitoring	ON
Serverdb full	NO
Connect possible	YES
Catalog cache pages used	64
Checkpoint wanted	NO
No of bad indexes	0
Vtrace	DEFAULT
Termchar set	IBM437_GER ASCII
Serverdb Configuration	
Default code	ASCII
Date/Time format	INTERNAL
Session timeout	900
Lock timeout	360
Request timeout	180
Log mode	SINGLE
Log segment size in KB	20000
No of Log Devspaces	1
No of Data Devspaces	1
Mirrored Devspaces	NO

Serverdb Usage	
Total data space in KB	1953736
Max persistent data space in KB	1893740
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
Last data page no	378564
Updated perm pages	156224
SYS DEVSPACE	
Devspace name	/u/sys
Total data space in KB.....	14724
LOG NAME 1	
Devspace name	/u/arc1
Total data space in KB	120000
DATA DEVSPACE 1	
Devspace name	/u/dat1
Total data space in KB	1953736

Info=>Kernel Parameters

Info=>Kernel Parameters

Features

This menu function displays the parameters that have been loaded into the database server.



```

Info Kernel Parameters
-----
Restart_shutdown ..... MANUAL
Default_code ..... ASCII
Date_time_format ..... INTERNAL
Maxarchivelogs ..... 1
Maxdatadevspaces ..... 1
Maxbackupdevs ..... 4
Mirrored_data ..... NO
Log_mode ..... SINGLE
Log_segment_size in KB ..... 20000
Log_io_queue ..... 20
Rollback_cache in KB ..... 40
Reserved_redo_size in KB ..... 0
Maxcpu ..... 1
Maxservertasks ..... 13
Maxusertasks ..... 50
Maxlocks ..... 2500
Deadlock_detection ..... 1
Session_timeout in sec..... 900
Lock_timeout in sec ..... 360
Request_timeout in sec ..... 180

```

```

Info Kernel Parameters
-----
Maxdatapages in KB..... 60000
Packet_size in bytes ..... 32768
Cat_cache_supply in KB ..... 212
Data_cache in KB ..... 6000
Sequence_cache in KB ..... 4
Dbproc_data_cache in KB ..... 70
Dbproc_code_cache in KB ..... 350
Converter_cache in KB ..... 46
Rundirectory .. /sqldb/E20/db/wrk/E20
Opmsg1 ..... /dev/syscon
Opmsg2 ..... /dev/null
Kerneldiagsize in KB ..... 400
Kerneltracesize in KB..... 800
Utility_protsize in KB ..... 400

```

Info=>Users

Features

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

In special cases, it may be necessary to stop a user process. You can use *Select* → *Kill* to remove the user and stop the user session. The current transaction is rolled back.

Example:

User ID	Term ID
MAYER	hst1tys3
SCHULZ1	hst1tys5
DEMO	hst3tys1
MAYER	hst1tys2
TEST02	hst2tys5

The following information is displayed:

<i>User ID</i>	contains name of the connected user.
<i>Term ID</i>	contains a terminal identification which depends on the particular operating system.

Info=>Caches

Info=>Caches

Features

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	353	98
File directory.....	1755	1522	233	87
FBM	6	6	0	100
Converter.....	22	22	0	100
USM	0	0	0	0
Catalog.....	5941	4584	1357	77

Information about the ADABAS caches is displayed:

<i>Accesses</i>	Total number of accesses
<i>Successful</i>	Number of successful accesses
<i>Failed</i>	Number of unsuccessful accesses to the ADABAS data cache
<i>Hit rate</i>	Percentage of successful accesses to the database data cache

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

<i>File directory</i>	File Directory Cache 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.
<i>FBM</i>	The FBM cache This area is used for the management of free disk blocks.
<i>Converter Cache</i>	The Converter cache
<i>USM</i>	The USM cache This area is used for the User Storage Management. (User Storage Management)
<i>Catalog</i>	The Catalog cache Information from the SQL catalog of the database provided in the catalog cache is available quickly because no accesses to the disk are required. The hit rate for this area should therefore be more than 80%.

Info=>I/O Accesses

Info=>I/O Accesses

Features

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



Data pages	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 data devspace, the number of logical and physical read and write accesses is displayed:

<i>Catalog</i>	denotes the data devspace used by the data dictionary of the database system. Frequent writing to this area is a sign of modifications made to the database design.
<i>Permanent</i>	is the actual data devspace for permanent data.
<i>Temporary</i>	is a data devspace on the disk used temporarily; it is required, for example, for the generation of selected datasets.
<i>Leaf, Level 1, Level 2, Level 3</i>	Information on the structure of the data trees. ADABAS organizes the storage of data in the form of B* trees.

Info=>Locks

Features

This menu function informs you 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.LIEFERANTEN    9 00FFFE000000000001
  LOCK ROW excl (360s) 00.00000058.000008 DBADM  qlcptys4 012
    
```

The upper part of the screen displays the following information:

- average number of locks held and requested
- maximum number of locks held and requested
- number of collisions and escalations that have occurred
- number of current row or table locks.

The lower part of the screen displays the following information:

- current locks
- current lock requests described in detail.
- relevant tables and rows
- the type of lock and the
- lock holder and the lock holder's terminal

Share locks	These protect used data against modifications while being accessed. Other users can only read-access data which is locked in this way.
Exclusive locks	These prevent other users from accessing the same data. Not even read-access is possible. Therefore, if exclusive locks are held for a lengthy period, they can be disadvantageous for the other users. Locks are released again by a <i>Commit</i> or <i>Rollback</i> or by a <i>Lock Timeout</i> . The parameters REQUEST_TIMEOUT and LOCK_TIMEOUT are used to resolve blocking and deadlock situations and can be configured (see Info → Configuration [Page 66]).
REQUEST_TIMEOUT	This indicates the time a task may wait for the setting of a requested lock before it will be cancelled.

Info=>Locks

LOCK_TIMEOUT	This indicates the time a task may hold a lock without activities, before it can be rolled back automatically (ROLLBACK).
--------------	---

Info=>Log

Features

This menu function displays information about the state and size of the log devspace as well as the activity of the logging process.

Example:

```

Log
-----
Max. size in KB ..... 159988 Redosize in KB ... 79994
Segment size in KB .... 40000 Log mode ..... SINGLE
Used size in KB ..... 17316 Used size in % ..... 11
Log not saved in KB ... 3215 Log not saved in % ..... 1

Since SERVERDB restart
-----
Physical writes ..... 41 Log since last backup ... 2187
Physical reads ..... 0

Since last save of log
-----
Savepoints written .... 31 Checkpoints written ... 1

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 following information is displayed:

<i>Max. size</i>	Maximum storage space available in the log devspace
<i>Segment size</i>	Size of a segment Segmentation of the log devspace is especially advantageous when the Autosave Log is active. Completed log segments are automatically saved while the database is operational. The saved segment area is then made available again to the logging mechanism for overwriting.
<i>Redo size</i>	Size of area reserved for log in the data area When logs are being restored, the transactions that must be redone are copied to the data area so that the log area is left unchanged for a restart. To enable the data area to accept all the required log files, part of the data area is reserved for them and is therefore not available for permanent data objects. The parameter RESERVED_REDO_SIZE specifies the size of the reserved data area in 4 KB pages.
<i>Log mode</i>	Mode in which the log is operated

Info=>Log

<i>Used size</i>	Size of used space
<i>Used size in %</i>	Percentage represented by the space used The usage level of the log devspace should be monitored carefully. If the usage level reaches 100%, all database activities are automatically stopped. Work can only continue after the log devspace has been saved. If the usage level exceeds 66%, messages are output to the diagnostic file (knldiag) of the database.
<i>Log not saved</i>	Size of log area not yet saved.
<i>Log not saved in %</i>	Percentage represented by log area not yet saved
<i>Physical writes</i>	Write activities in the log devspace since the last restart of the database system
<i>Physical reads</i>	Read activities in the log devspace since the last restart of the database system
<i>Log since last backup</i>	Number of log pages written since the last data backup
<i>Savepoints written</i>	Number of savepoints written
<i>Checkpoints written</i>	Number of checkpoints written
<i>Queue size</i>	Current size of the log queue
<i>Group commits</i>	The number of group commits
<i>Queue allocated</i>	Space currently allocated
<i>Waits for logwriter</i>	Number of wait states for log write operations
<i>Queue entries</i>	Maximum size of log queue
<i>Max. waits</i>	Maximum number of wait states per log page
<i>Overflows</i>	Number of log queue overflows
<i>Avg. waits</i>	Average number of wait states per log page

Info=>Last UpdStatistics

Features

This menu function displays the execution date of the last update statistics performed on an arbitrary set of tables of the serverdb. You set the date using [Operating → Update Statistics \[Page 60\]](#).

It can take a long time to produce the result, depending on the number of existing tables and selected search conditions.

Input screen:

```

Date of last Update Statistics

User Name      : *
Table Name     : *
Column Name    : *
    
```

If you do not require any restriction, enter * as placeholder.

The result of this request is sorted according to the date and time of the last update statistics execution:

Owner	Tablename	Last Update Date	Statistics Time
SQLREISEN00	ORT	1997-10-03	14:11:20
SQLREISEN00	KUNDE	1997-10-03	14:11:20
SQLREISEN00	BUCHUNG	1997-10-03	14:11:19
SQLREISEN00	HOTEL	1997-10-03	14:11:19
SQLREISEN00	RAUM	1997-10-03	14:11:19
SQLREISEN00	KONTO	1997-10-03	14:11:18
SQLREISEN10	HNR_DOM	1997-09-30	08:23:03
SQLREISEN10	RAUM_DOM	1997-09-30	08:23:00

Info=>Tasks

Info=>Tasks

Features

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

Example (for UNIX):

```

ID      UNIX  TYPE  APPL  State  Timeout  Region  Wait  UKPsleep
pid     pid
T1      7888  Timer  -1   Vsleep      0    0    0    0 2670294 (s)
T2      7888  Logwrl -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 tasks can have the following states:

<i>Command wait</i>	Kernel is waiting for a command.
<i>Vsuspend</i>	Wait state, e.g. for system resources.
<i>IOWait (R/W)</i>	Task is waiting for I/O (read/write).
<i>Vsleep</i>	Task is inactive for a short time.
<i>Vbegexcl</i>	Task is waiting for a Critical Region (latch).
<i>Vendexcl</i>	Task is operable after a protected memory access at which a lock collision occurred and UKP control was passed to the waiting task.
<i>Vwait</i>	Task is waiting for an SQL lock.
<i>Wait for LW</i>	Task is waiting for successful log writing.
<i>Running</i>	Task is running.
<i>Runnable</i>	Task is operable, but is waiting for CPU allocation.
<i>Command reply</i>	Task is returning the response.

Info=>Version

Features

This menu function displays the following current versions

- Database kernel
- Runtime environment
- CONTROL

Info=>Help

Info=>Help

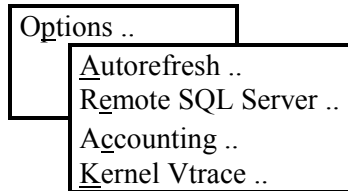
Features

If you select this menu option, the CONTROL help texts will be displayed.

See also [CONTROL Menu Structure and User Guidance \[Page 10\]](#).

Options

Features



[Options → Autorefresh \[Page 82\]](#)

[Options → Remote SQL Server \[Page 83\]](#)

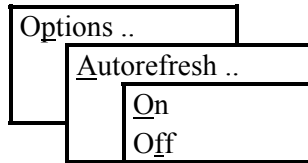
[Options → Accounting \[Page 84\]](#)

[Options → Kernel Vtrace \[Page 85\]](#)

Options=>Autorefresh

Options=>Autorefresh

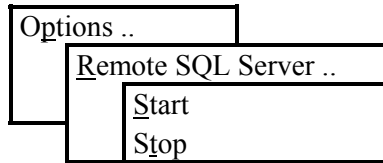
Features



This menu function determines whether or not the values displayed in the Main Menu are to be refreshed automatically after returning from a menu function. You can use *Off* to accelerate the response times of CONTROL.

Options=>Remote SQL-Server

Features



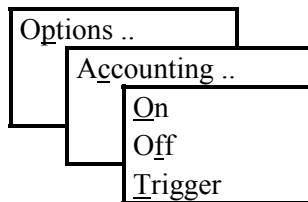
This menu function starts and shuts down the Remote SQL Server which is required for a client/server connection. If the Remote SQL Server is started, application processes running on another computer (client) can connect to the ADABAS serverdb and open database sessions.

<i>Remote SQL Server</i> → <i>Start</i>	Starting the remote SQL server
<i>Remote SQL Server</i> → <i>Stop</i>	Canceling all connections that users have established from other computers to this database serve

Options=>Accounting

Options=>Accounting

Features



Accounting → *On* initiates the recording of session-specific statistics about resources used. This statistical data is collected in the SYSACCOUNT table.

The SYSACCOUNT table has the following structure:

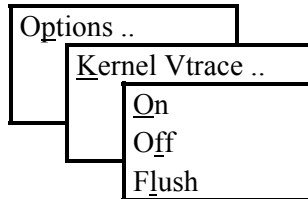
```
CREATE TABLE SYSACCOUNT (
    SERVERDB_NO          FIXED      ( 4),
    SESSION              FIXED      (18),
    NOLOG_SESSION        CHAR       ( 3),
    USERNAME             CHAR       (64),
    GROUPNAME            CHAR       (64),
    TERMID               VARCHAR    (18),
    DBNAME               CHAR       (64),
    CONNECTDATE          DATE,
    CONNECTTIME          TIME,
    RELEASEDATE          DATE,
    RELEASETIME          TIME,
    COMMANDCOUNT       FIXED      (10),
    CPUTIME              FIXED      (10),
    DATA_CACHE_ACCESS  FIXED      (18),
    SESSIONEND           CHAR       ( 8),
    PROGTYPE             VARCHAR    ( 8),
    PROGOWNER            CHAR       (64),
    PROGNAME             CHAR       (64),
    CATALOG_CACHE_SIZE  INTEGER    (10),
    CAT_CACHE_HITRATE   FIXED      ( 4),
    DBPROC_CACHE_SIZE   INTEGER    (10),
    TEMP_SIZE_CACHE     INTEGER    (10),
    TEMP_HITRATE_CACHE  FIXED      ( 4))
```

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

For sessions that do not leave traces in the SYSACCOUNT table although they run for a very long time, there is the *Accounting* → *Trigger* menu function. A signal is given to each user session to enter a row of information into the SYSACCOUNT table. This signal is always checked before a new SQL statement is executed. Once the signal has been processed, it is no longer important and a new entry can only be made using *Accounting* → *Trigger* or at the end of the session.

Options=>Kernel Vtrace

Features



The *Kernel Vtrace* → *On* menu function enables an ADABAS kernel trace for a particular command (knltrace in the rundirectory).

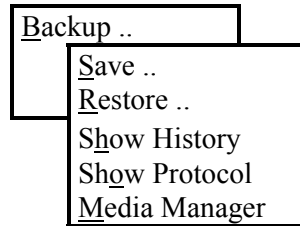
The *Kernel Vtrace* → *Off* menu function disables this logging.

The *Kernel Vtrace* → *Flush* menu function only makes sense if the ADABAS kernel log has already been enabled. You confirm a *Kernel Vtrace* → *Flush* to write the last written remaining buffered entries to the log file. This file cannot be read directly. It can only be interpreted by ADABAS customer support.

Backup

Backup

Features



The *Backup → Save* menu function allows you to save immediately (ad hoc) the complete database contents and log contents, or just the modifications made since the last backup of the database or of the log. The *Backup → Restore* menu function uses the generated backups to restore a corrupt database.

The *Backup → Show History* menu function displays the backup history, and the *Backup → Show Protocol* menu function shows the backup log files. You define the individual backup media by using the *Backup → Media Manager* menu function.

[Backup → Save \[Page 87\]](#)

[Backup → Restore \[Page 89\]](#)

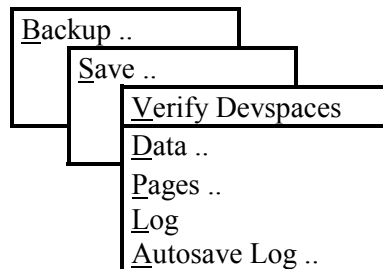
[Backup → Show History \[Page 90\]](#)

[Backup → Show Protocol \[Page 91\]](#)

[Backup → Media Manager \[Page 92\]](#)

Backup=>Save

Use



Objects which can be backed up:

Data	Backup of the entire contents of the database without checkpoint (Backup → Save → Data → Recovery [Ext.]) or with checkpoint (Backup → Save → Data → Migration [Ext.])
Pages	Backup of all changes since the last backup without checkpoint (Backup → Save → Pages → Recovery [Ext.]) or with checkpoint (Backup → Save → Pages → Migration [Ext.])
Log	Backup of the entire log (Backup → Save → Log [Ext.])

Prerequisites

Each backup is written to a backup medium that can be selected from the Media Manager. After selecting the corresponding type of backup, the [Media Manager \[Page 92\]](#) appears for selection or definition of the backup medium. You can then define or change backup media. The backup media can also be defined in advance using the [Backup → Media Manager \[Page 92\]](#) menu function.

The media definitions comprise the path names and properties of the tape device and of the tapes or files under a name that can be selected freely and which should relate to the practical situation. Under these names, the media can be re-used for all possible backup operations. We generally recommend that you use tapes and cassettes as backup media. Only separate media can be kept at other locations as a safeguard in the event of fire or similar hazards. Using files as backup media only makes sense if they are written to tape or cassette afterwards.

Activities

[Procedure for saving \[Page 33\]](#)

After selecting the medium, CONTROL expects that the medium is ready for writing (the tape or cassette must have been inserted in the device or the file must be write enabled). During the backup, the storage device is given a label that contains the type of backup and other specifications. The label used is displayed and must be confirmed.

You should always put the [label \[Page 29\]](#) on the sticker of the storage device.

Backup=>Save**Features**

The backup functions *Data → Migration* or *Pages → Migration* are completed with a checkpoint. This means that all backups that are created are always consistent, that is, if they are used for restoring, you do not need any other log backups.

Since the speed of the backup is important in a periodic backup for a recovery, the checkpoint will be an impediment. The *Data → Recovery* and *Pages → Recovery* backups can only create a consistent database state in conjunction with a further log backup during a recovery.

[Backup → Save → Verify Devspaces \[Ext.\]](#)

[Backup → Save → Data → Recovery \[Ext.\]](#)

[Backup → Save → Data → Migration \[Ext.\]](#)

[Backup → Save → Pages → Recovery \[Ext.\]](#)

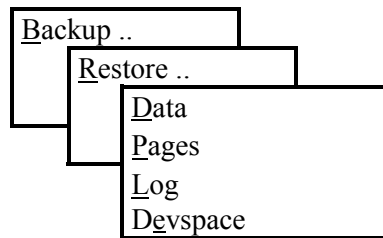
[Backup → Save → Pages → Migration \[Ext.\]](#)

[Backup → Save → Log \[Ext.\]](#)

[Backup → Save → Autosave Log \[Ext.\]](#)

Backup=>Restore

Use



The restore functions can only be used when the database system is in COLD operating mode (exception: *Backup → Restore → Devspaces*).

Backups which can be restored:

Data	Complete data backup (Backup → Restore → Data [Ext.])
Pages	Incremental backup (Backup → Restore → Pages [Ext.])
Log	Log devspace (Backup → Restore → Log (UNTIL) [Ext.])
Devspace	Recovering a disk devspace from the mirrored disk (Backup → Restore → Devspace [Ext.])

Prerequisites

After selecting the desired menu item, the [Media Manager \[Page 92\]](#) appears. Here you can select the restore medium.

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.

Features

[Backup → Restore → Data \[Ext.\]](#)

[Backup → Restore → Pages \[Ext.\]](#)

[Backup → Restore → Log \(UNTIL\) \[Ext.\]](#)

[Backup → Restore → Devspace \[Ext.\]](#)

Backup=>Show History

Backup=>Show History

Features

Select *Backup* → *Show History* to display the backup operations performed so far. The following are displayed: backup identification, backup type, date and time of backup start and backup end, backup medium, and current status.

The backup history is stored in a file named `control.knl` (see [Files used by CONTROL \[Page 25\]](#)) which is stored in the rundirectory of the serverdb.

Backup=>Show Protocol

Features

Use the *Backup* → *Show Protocol* menu function to display the end of the backup protocol. Information about the last save or restore operation is recorded there. You can page up to the beginning of the file.

The backup protocol is stored in a file named `control.utl` (see [Files used by CONTROL \[Page 25\]](#)) which is stored in the rundirectory of the serverdb.

Backup=>Media Manager

Backup=>Media Manager

Use

Select *Backup* → *Media Manager* if you wish to define, change or delete descriptions of backup media.

Media List

If backup media are already defined at the start of the Media Manager, the following list of media available for selection is displayed:

```

Media Manager
EXAMPLE          T  Y      0  /dev/rmt0          ( )
FILE             F  N      0  /backup/dblog.save ( )
VERSFILE        F  V      0  /backup/dblog.s   ( )
DAT90           T  N 400000 /dev/rmt0         ( X )
DAT120          T  N 500000 /dev/rmt1         ( X )
TWOTAPES        T  N      0  PARALLEL          ( )

Parallel-Id : TWOTAPES

Use Buttons or Keys to handle media - otherwise Return.

  Show  Label  Edit  New  Next  Prev  Return

```

Column 1: Medium name

Column 2: Device type

Column 3: Overwrite mode

Column 4: Size

Column 5: Path name of medium

Column 6: X means the medium belongs to the parallel group. No entry means the medium does not belong to the parallel group. The Media Manager only allows one group of parallel media to be defined. In the example, the name of the group of parallel media is TWOTAPES, and the media DAT90 and DAT120 belong to this group.

Features

Use the cursor to mark the desired medium.

Function	Explanation	User Key
<i>Show</i>	Display of the medium	Enter
<i>Label</i>	Display of the label (in the case of a tape if the tape is written, or in the case of a file that is already written)	F6
<i>Edit</i>	Changing the medium with the Media Editor	F10
<i>New</i>	Definition of a new medium. The Media Editor is empty.	

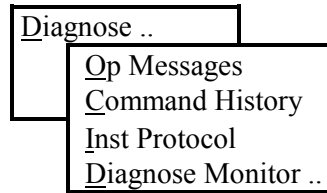
Backup=>Media Manager

<i>Next, Prev</i>	Scrolling through the list of media	
<i>Return</i>	Leaving the list of media	End, F3

[Parallel Media \[Ext.\]](#)

[Media Editor \[Ext.\]](#)

[Special Media \[Ext.\]](#)

Diagnose**Diagnose****Features**

[Diagnose → Op Messages \[Page 95\]](#)

[Diagnose → Command History \[Page 96\]](#)

[Diagnose → Inst Protocol \[Page 97\]](#)

[Diagnose → Diagnose Monitor \[Page 98\]](#)

Diagnose=>Op Messages

Features

This menu function shows the console protocol file or diagfile of the ADABAS server (see also the `knldiag` file in the rundirectory of the serverdb). In the display screen, you can page up and down using the *Previous* and *Next* buttons. *Return* closes the display screen and returns you to the main screen of CONTROL.

The console protocol file helps you to interpret the events that occurred during ADABAS server operation. The entries are made in chronological order. When the console protocol file reaches a certain size, it is overwritten cyclically. The entries are always done to the end of the protocol file. At each start of the serverdb with the menu functions *Operating* → *Restart* → *Warm* or *Operating* → *Restart* → *Cold*, a copy of the console protocol is generated under the name `knldiag.old` and the file `knldiag` is created afresh.

The following events are documented for example:

- Starting and shutting down the ADABAS server
- Information about the physical storage areas (devspaces) of the ADABAS server
- Information about user processes that have established database sessions with the ADABAS server
- Error messages of the ADABAS server system embedding which may be the result because of wrongly selected system parameter sizes
- System error messages, e.g., after device failures
- Error messages and warnings from the ADABAS kernel, for example after an "Emergency Shutdown" (see the "Usage Levels" in [the main screen \[Page 22\]](#)) or in case of internal inconsistencies.

Diagnose=>Command History

Diagnose=>Command History

Features

This menu function shows all actions executed so far in CONTROL. In addition, the results and any errors relating to the activities performed are displayed. The cursor is always positioned initially at the end of the Control Command History, because the most recent information is recorded there. Each entry contains the date, time, and process-ID.

Diagnose=>Inst Protocol

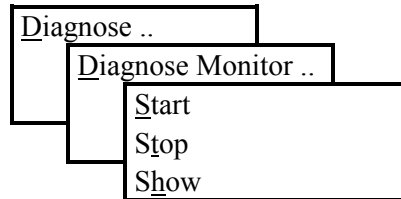
Features

This menu function displays the result protocol of the last installation or from the last loading of the system tables. The cursor is always positioned initially at the end of the installation protocol file, because the result of the installation and a summary of the errors which have occurred is recorded there.

Diagnose=>Diagnose Monitor

Diagnose=>Diagnose Monitor

Features



The *Diagnose* → *Diagnose Monitor* → *Start* menu function takes you to the next input screen:

```

Entries in monitor history:    255
Selectivity (in permill)    :    20
Virtual reads                 : 10000
Elapsed time (in msec)      :   1000
  
```

<i>Entries in monitor history</i>	Number of the monitor events to be stored in the target table
<i>Selectivity</i>	Relation between ROWS_QUAL and ROWS_READ in tenths of a percent
<i>Virtual reads</i>	maximum number of virtual reads
<i>Elapsed time</i>	Maximum time for a select and subsequent fetch in milli-seconds.

Enter the desired values and confirm to activate the Diagnose Monitor.

Use the *Diagnose* → *Diagnose Monitor* → *Show* menu function to display the result of long-running commands in the form of a table report. The result table is sorted according to the RUNTIME of the command and contains the following information:

- RUNTIME
- ROWS_READ
- ROWS_QUAL
- STRATEGY
- VIRTUAL_READS
- PHYSICAL_IO
- DATETIME
- TERMID
- USERNAME

Diagnose=>Diagnose Monitor

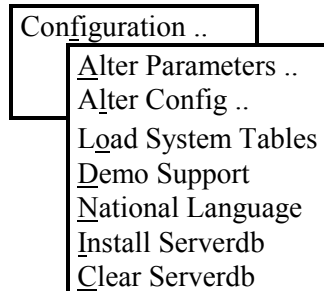
- SQL_COMMAND

Use the *Diagnose* → *Diagnose Monitor* → *Stop* menu function to deactivate the Diagnose Monitor.

Configuration

Configuration

Features



[Configuration → Alter Parameters \[Page 101\]](#)

[Configuration → Alter Config \[Page 102\]](#)

[Configuration → Load System Tables \[Page 103\]](#)

[Configuration → Demo Support \[Page 104\]](#)

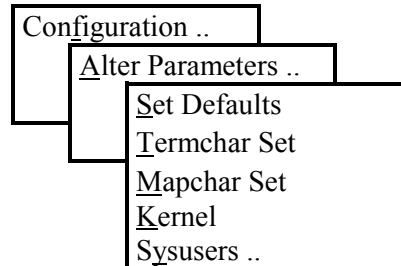
[Configuration → National Language \[Page 105\]](#)

[Configuration → Install Serverdb \[Page 107\]](#)

[Configuration → Clear Serverdb \[Page 108\]](#)

Configuration=>Alter Parameters

Features



[Configuration → Alter Parameters → Set Defaults \[Ext.\]](#)

[Configuration → Alter Parameters → Termchar Set \[Ext.\]](#)

[Configuration → Alter Parameters → Mapchar Set \[Ext.\]](#)

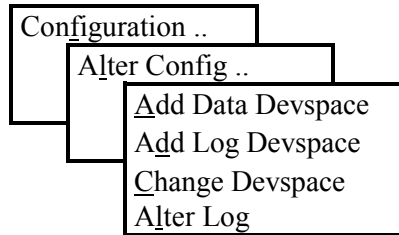
[Configuration → Alter Parameters → Kernel \[Ext.\]](#)

[Configuration → Alter Parameters → Sysusers \[Ext.\]](#)

Configuration=>Alter Config

Configuration=>Alter Config

Features



You can use *Alter Config* to modify the configuration of the serverdb at a later point in time.

[Configuration → Alter Config → Add Data Devspace \[Ext.\]](#)

[Configuration → Alter Config → Add Log Devspace \[Ext.\]](#)

[Configuration → Alter Config → Change Devspace \[Ext.\]](#)

[Configuration → Alter Config → Alter Log \[Ext.\]](#)

Configuration=>Load System Tables

Features

When changing from one version to another, you can use this function to update the system tables.

```

Load System Tables for Update Installation
-----
Create system tables for DBLOAD..... OK
Create general system tables..... OK
---> Load messages and help infos..... ACTIVE
Load SET defaults..... --
Load system tables for ADABAS TOOLS..... --
Load system tables for SQL-PL..... --
Load SQL-PL WORKBENCH..... --
Create ODBC tables..... --
Create system views..... --
Load SQL catalog views..... --
Load system tables for QueryPlus..... --
    
```

If an error occurs, the ERROR status appears. You can then use *Protocol* to view the installation protocol file that contains more information about the error which has occurred.

Configuration=>Demo Support

Configuration=>Demo Support

Features

You can use this menu function to load a complete demonstration database of ADABAS. The demonstration database is currently available in English and German. If errors occur while loading the demo, the load protocol file will be displayed in a CONTROL window.

You will find further details in the corresponding readme file.

(`$DBROOT/demo/deu/README.txt` or `$DBROOT/demo/eng/EREADME.txt`).

Configuration=>National Language

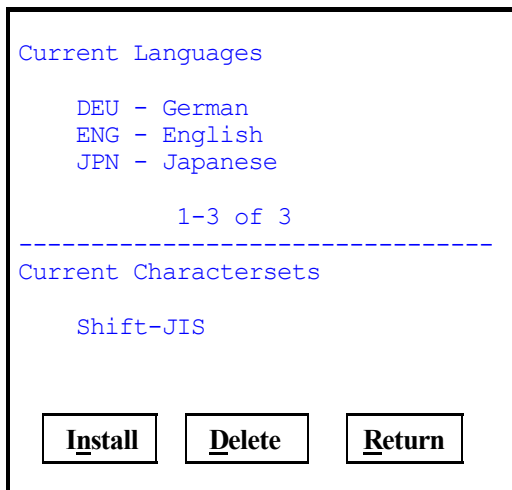
Features

You can use this menu function to load system messages and help files for additional languages. For some languages, character sets are provided that can also be loaded after a first installation. Character sets are needed to input or output Unicode data stored in the database in a special presentation code (which depends on the characters of the language and their representation on the terminal).

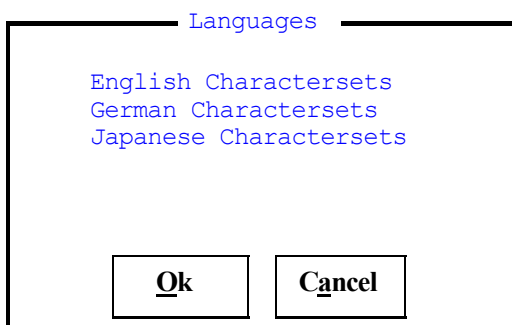
By default, CONTROL loads only the English system messages and help files during the installation of a new serverdb. The same applies when you select *Configuration* → *National Language*. If you wish to have system messages or help information displayed in another language, the data must be first be loaded, and for some languages also the corresponding character set. ADABAS supports a series of additional languages and their corresponding character sets.

After you have selected *National Language*, CONTROL displays all languages and character sets which are currently loaded.

Example:



Install allows you to display a list of all languages that can be loaded. Select the desired language and activate it with *Ok* or *Enter*.



Configuration=>National Language

Both the language and the character set are displayed for selection in one screen for languages that can only be loaded if the corresponding character set has been loaded. Several character sets can be provided for one language.

```
Load Language
(X) Japanese messages
    and help infos
-----
Load Charactersets
(X) Shift-JIS
(X) EUC (U-JIS)
Ok Cancel
```

You must mark all character sets to be loaded with an **x** (an x is entered for each language by default). The selection can be canceled with a blank if the character set has already been loaded.

After loading the marked character sets, you will be asked whether you want to load the system messages and help files for the selected language. If you wish to load only the character sets, use *Cancel* to abort.

If there are more than five languages or character sets, you can scroll through the lists.

ADABAS can hold up to five character sets at the same time. If a further character set is to be activated and all five spaces are occupied, you must first delete one of the five character sets.

Configuration=>Install Serverdb

Features

You can use this menu function to recreate the existing serverdb. When doing so, the current data will be lost. The procedure corresponds to that of a first installation. See [Installation of a New Serverdb \[Page 15\]](#). For a presetting, the values valid so far are offered.

You must always load the system tables after *Install Serverdb*, either by loading a DBEXTRACT using the LOAD tool, by restoring a database backup using the CONTROL menu option ([Backup → Restore → Date \[Ext.\]](#)) or by loading the tables using [Configuration → Load System Tables \[Page 103\]](#).



The old database contents will then be lost.

You can also use this function to install a serverdb from a data backup either with or without the configuration of the backup version. This is described in detail in [Installing a New Serverdb \[Page 15\]](#) and [Installing from an Existing Data Backup \[Page 19\]](#).

A database backup created with [Backup → Save → Data → Migration \[Ext.\]](#) is most suitable for copying an entire database with Restore.

After successful installation of the serverdb, you should perform a data backup.

Configuration=>Clear Serverdb

Configuration=>Clear Serverdb

Features

This menu function removes the current serverdb.

If you confirm *Clear Serverdb*, all data of this database is lost.

All information about this serverdb and its contents (all files, directories, and devices) will be deleted.

Afterwards, the serverdb name is freely available again.

Troubleshooting when Problems Occur

The following information is intended to help database administrators find the right strategy for correcting errors when they occur. In order to do this, the error situations must be precisely analyzed on the basis of the ADABAS error protocol files. When the cause of error is known, the appropriate measures can be taken quickly to return the system to operation.

[System crash \[Page 110\]](#)

[Log devspace is full \[Page 111\]](#)

[Database is full \[Page 112\]](#)

[Log disk is defective \[Page 113\]](#)

[System error \[Page 114\]](#)

System Crash

System Crash

Procedure

[Saving the protocol files \[Ext.\]](#)

[The x lock analysis tool under UNIX \[Ext.\]](#)

[Finding the cause of a system crash \[Ext.\]](#)

Log Space is Full

All modifying transactions are recorded consecutively in the log devspace of the database. The log devspace is released when the log is saved regularly with the database in warm mode.

The log continues to fill up; from the moment it becomes 2/3 full, warnings are recorded in the `knldiag` file. If the log is not saved in time, the log devspace of the database may become full. This results in automatic termination of the database operation.

Messages in the `knldiag` file:

```

11-18 14:55:57 0x20D WRN 52438 LOG      log used 60% (1200 pages
left)
11-18 14:56:03 0x20D WRN 52438 LOG      log used 70% (898 pages left)
11-18 14:56:08 0x20D WRN 52438 LOG      log used 80% (599 pages left)
11-18 14:56:11 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:11 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 14:56:12 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:12 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 14:56:13 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:13 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 14:56:16 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:16 0x20D      18263 CONNECT  Connect req.
(T10,Node:'', PID:0x202)
11-18 14:56:18 0x20D WRN 52438 LOG      log used 90% (299 pages left)
11-18 14:56:20 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:20 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 03:56:20 PM 0x20D      18282 CONNECT  Connection released, T10
11-18 03:56:20 PM 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 03:56:20 PM 0x20D      18282 CONNECT  Connection released, T10
11-18 03:56:20 PM 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 14:56:20 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:20 0x20D      18263 CONNECT  Connect req.
(T10,Node:'', PID:0x202)
11-18 14:56:22 0x20D WRN 52438 LOG      log used 95% (150 pages left)
11-18 14:56:26 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:42 0x20D      18263 CONNECT  Connect req. (T10,
Node:'', ID:0x19F)
11-18 14:56:42 0x20D      18282 CONNECT  Connection released, T10
11-18 14:56:43 0x20D      18263 CONNECT  Connect req. (T7,Node:'',
PID:0x1EC)
----- current write position -----

```

Procedure

[Reaction of the Database Administrator \[Ext.\]](#)

Database is full

Database is full

If there is no longer enough space available on the data devspaces for permanent data or for temporary result sets, the SERVERDB stops in order to permit the administrator to define an additional data devspace. The SERVERDB continues when the additional data devspace has been defined.

If the data devspace of a database is full, this is also displayed in the Main Screen of CONTROL.

Procedure

[Reaction of the Database Administrator \[Ext.\]](#)

Log Disk is Defective

The defective log devspace is displayed in the `knldiag` file: `BAD DEVSPACE`. In addition, CONTROL displays the error message `BAD DEVSPACE` in the Main Menu.

When the DUAL log mode is set, the log consists of two archive logs with the same contents. If one log devspace fails in the DUAL log mode, the database continues to be operable. Database logging is then only done to the intact log devspace.

If both log devspaces fail in the DUAL log mode or if the SINGLE log mode is set, the database must be restored.

Procedure

[Reaction of the Database Administrator \[Ext.\]](#)

System Error**System Error**

ADABAS system error numbers range from -9000 to -9999. These error numbers refer to internal errors that the administrator cannot correct without appropriate support. For this reason, they are not described in the Messages and Codes manual but are recorded in the `knldiag` file.

Procedure

[Reaction of the Database Administrator \[Ext.\]](#)

Performance Analysis and Tuning

In addition to the option to detect long-running commands using the *Diagnose* → *Monitor* menu function, the `x_wizard` tool allows a detailed analysis of the monitoring data. This analysis can be called periodically.

ADABAS Bottlenecks

ADABAS Bottlenecks

Call

`x_wizard [-t <interval>] [-x] [-p|-a] [-d <n>] [-b] [-s] [-D] [-L] [-k|-K] [-l <language>]`

[Options \[Ext.\]](#)

Features

`x_wizard` attempts to analyze the bottlenecks of the current database run. The basis for this analysis is the database monitoring and the database console `x_cons`. Detected bottlenecks are output in text form to rapidly provide database administrators with an overview of the possible causes of performance problems. The analysis can be done either once or at regular time intervals using the option `-t`.

If possible `x_wizard` should be called on the database server because the database console `x_cons` cannot be used in remote operation. If only a remote call is possible, only the monitoring data can be analyzed. In this case, you must not use option `-x`.

An interval of 15 minutes is sufficient (`-t 900`) for routine monitoring of database operation in productive systems. Logging should be enabled with `-p` or `-a` to provide ADABAS Support with an overview of the database activities. For a brief observation of the current database operation, a measuring interval of 10-30 seconds is recommended.

Any bottlenecks detected are classified according to their importance (I : Info, W1 to W3: bottleneck warnings ranging from minimum to serious) The classification of warnings refers to running applications. As a rule, warnings displayed at a system start can be ignored.

Not all `x_wizard` outputs are necessarily caused by actual bottlenecks. For example, table scans can be useful in certain situations, long runtimes of statements can automatically occur with large datasets etc. If bad search strategies (rows read/rows qualified) are suspected, an exact analysis is unavoidable.

[Requirements \[Ext.\]](#)

[x_wizard messages \[Ext.\]](#)

The Course of Measured Values (x_wiztrc)

Call

x_wiztrc [-i <file name>] [-P <lines>] [-o|-c|-t|-O|-C|-g|-s|-S|-l|-r|-R|-T|-d|-p]

[Options \[Ext.\]](#)

Features

x_wiztrc evaluates the data collected by x_wizard and outputs it chronologically in tabular format. The measured values given always refer to the time interval between two measuring times.

x_wiztrc is designed for ADABAS support and development staff, not for the end user. Therefore, no detailed explanation of the output parameters is given.

Prerequisites

Previous start of x_wizard with the options -t sec -b ...

[x_wiztrc Output Tables \[Ext.\]](#)

Table Statistics and Structural Checks (xpu)

Table Statistics and Structural Checks (xpu)

Call

xpu -v|-s [-u] [-e] <count>

[Options \[Ext.\]](#)

Features

xpu allows the following to be done in parallel:

- generating table statistics for the cost-based optimizer.
- checking the B* tree structure of tables.

While checking the B* tree with **xpu -v**, the following two files are generated in the current directory:

- `chtab.prt`, with a list of all tables checked.
- `chtab.err`, with the tables where errors occurred during verification.

Use the recorded error codes to check whether these errors are so serious that a recovery is required.

When table statistics are updated with **xpu -s**, an `updcoll.prt` file is generated in the current directory. This file contains the following information: the tables for which the up-to-dateness of the statistics for the cost-based optimizer was checked and the tables for which an update of statistics was necessary.

If xpu detected no errors in table processing and was completed correctly, the return code 0 (zero) is output to the operating system; otherwise, a value different from 0.

For large databases, checking tree structures and creating exact statistical information requires the physical reading of large amounts of data. xpu allows for parallel read operations by distributing the actions to several tasks. The `count` parameter indicates how many database tasks are to operate in parallel. In any case you must ensure that `count` is less than the ADABAS kernel parameter `MAXUSERTASKS`. Preferably, xpu should be started in low-load times.

The `count` parameter should be equal to approximately twice the number of disks that are physically used for data devspaces (e.g. 5 devspaces on 5 disks: `count = 10`; 1 devspace on RAID 5 with 6 physical disks: `count = 12`).

After special error situations; for example after the reading of inconsistent database pages by a defective disk controller (error code -9026), tables are internally marked as defective and locked for further, modifying usage. However, the data on the hard disk may still be intact. **xpu -v** can be used at a later time, for example after exchanging the controller, to check the internal structure of all tables (but of no indexes). Tables diagnosed as correct are marked as such and can then be updated again.

xpu is started for the database user who is entered in the XUSER file as the default user. For other users, the `SQLOPT` environment variable must be set. Only the tables of the corresponding user are processed.

Table Statistics and Structural Checks (xpu)

The optimizer statistics should be updated weekly (**xpu -s**) for datasets that increase dynamically. The use of a sample (20000 lines per table) can greatly accelerate the calculation of statistics for very large datasets (**xpu -s -e**). A consistency check of the table structures (**xpu -v**) should be performed before each new backup generation, even if no errors occurred in the meantime.

Settings for NetTerm

Settings for NetTerm

The *NetTerm* terminal emulation can be obtained from the World Wide Web or by e-mail:

<http://starbase.neosoft.com/~zkrr01/netterm.htm>

zkrr0@neosoft.com

For optimum use of *NetTerm* from a PC, use the following settings:

- *File* → *Phone Directory*
Emulation = **VT100**
Keys = **CIS**
- Setting the **Enter** key
Options → *Setup* → *Keyboard Keys*
Select **Enter** on the displayed keyboard.
In the text field, select **^M** instead of **^J**.
Select **Change**, **Save** and **Ok**.

After logging onto the UNIX server, you must set the following variables:

```
DBHIF=NETTERM
```

```
DBTERM=vt100
```

```
TERM=vt100
```

```
TERMINFO=$DBROOT/terminfo
```

In CONTROL, use **F2** to select the *NETTERM* color setting. This setting is also available in QUERY and LOAD as one of several presentations that can be selected by means of the SET command.