Adabas for UNIX, Windows and OpenVMS

Adabas Version 6.3 Release Notes

Version 6.3.1

April 2013

This document applies to Adabas for UNIX, Windows and OpenVMS Version 6.3.1.

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

Document ID: ADAOS-RELNOTES-631-20130422

Table of Contents

1 Adabas Version 6.3 Release Notes

This document gives an overview of the features of Adabas Version 6.3 that have been introduced or modified since the previous release (Version 6.2). It also describes how to convert Version 3, Version 5.1, Version 6.1 and Version 6.2 databases to and from Version 6.3.

The document contains the following sections:

2 General Information

This chapter provides information which you should be aware of before you install and use Adabas Version 6.3.

Installation of Adabas

To install Adabas Version 6.3 on your machine, please follow the instructions given in the *Adabas Installation Guide*.

Please refer to the section ***Converting a Version 3, Version 5, Version 6.1 or Version 6.2 Database to and from Version 6.3*** for information about how to convert older versions of Adabas to Version 6.3.

The installation documentation is in the doc directory of the distribution CD as a PDF file (*install.pdf*), it can also be accessed in HTML format from the documentation overview page *overview.htm* in the doc directory of the distribution CD. In order to access the installation documentation directly, the distribution CD must be mounted: if an automount daemon is running, insert the CD into the CD-ROM drive, otherwise you must mount it using an appropriate command from the following table. Modify the mount directory and the CD-ROM device name to the values used at your site.

Platform	Mount Command and Options
AIX	/etc/mount -v-cdrfs -o ro /dev/cd0/mnt
HP-UX	/usr/sbin/mount -F cdfs -o cdcase/dev/dsk/c1t2d0 /mnt
LINUX IA-32	/bin/mount -t iso9960 /dev/cdrom /mnt -o ro,
Solaris	/sbin/mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /mnt

On Windows platforms, insert the distribution CD into a CD-ROM drive, and navigate to the doc directory.

Compatibility with Previous Versions

If you are upgrading from an earlier version of Adabas, for example from Version 3.3, Version 5.1, Version 6.1 or Version 6.2 to Version 6.3, you should first read the release notes of any intermediate or skipped versions in order to find out about any compatibility issues that might exist.

Relationship between Version 6.2 and Version 6.3

The reason for calling the current release version 6.3 is that this is the first release that supports Adabas in 64-bit mode on Windows platforms - this marks a big improvement for Windows users, as it removes the limitation of the buffer pool size to about 1 GB. Apart from this, the changes between Version 6.2 and version 6.3 are just the typical changes for a new service pack:

- Bug fixes;
- Limitations of the Version 6.2 functionality that still existed in version 6.2 SP1 have been removed.

As far as the maintenance policy is concerned, this means that Version 6.3 will be treated like a service pack on UNIX platforms; according to our maintenance policy, bugs are only fixed in the newest service pack, and the customer has to upgrade to this service pack. For Adabas Versions 6.2 and 6.3 this means that bugs in version 6.2 are only fixed in version 6.3 and that the customer must upgrade to version 6.3.

Support for Adabas Version 5.1, Version 6.1 and Version 6.2

Adabas Version 6.3 is the official replacement for Adabas Version 5.1, Version 6.1 and Version 6.2. Details of when support for earlier versions will cease can be found in Empower.

Adabas and Entire Net-Work

In order to access an Adabas Version 6.3 database remotely you will require Entire Net-Work 7.4 or above (Windows 7 and Windows Server 2008, 64 bit, and OpenVMS), or Net-Work 7.3.3 and above (all other platforms) and Entire Net-Work Client 1.3.3 or above.

3 New and Modified Features

This chapter describes the new and modifed features in Adabas Version 6.3.

General New Features

Adabas in 64-bit Mode on Windows

Starting with Version 6.3, Adabas is delivered in 64-bit mode on the new 64-bit Windows operating systems Windows 7 and Windows Server 2008. This lets you use buffer pool sizes > 4 GB.

Caution: If you use user exits or hyperexits, you must recompile them in 64-bit mode. Please ensure that the user exit/hyperexit code is portable and still runs in 64-bit mode.

ADAOPR: IO_TIME Parameter

The utility ADAOPR now has a new parameter IO_TIME. This is used to enable/disable and reset the logging of I/O times for the ASSO, DATA, WORK, NUCSRT and NUCTMP containers.

The keyword IO_TIMES has been added to the DISPLAY parameter, and is used to display the container IO times.

Date/Time Fields

Although the [NO]DST parameter for ADACMP and ADADCU has already been documented, it is only available with Version 6.3 and later.

A daylight saving time indicator for a date/time field with the TZ option may also be specified in the search buffer for an L3 and L9 command.

Adabas in 64-bit Mode on Windows

Starting with Version 6.3, Adabas is delivered in 64-bit mode on the new 64-bit Windows operating systems Windows 7 and Windows Server 2008. This lets you use buffer pool sizes > 4 GB.

Caution: If you use user exits or hyperexits, you must recompile them in 64-bit mode. Please ensure that the user exit/hyperexit code is portable and still runs in 64-bit mode.

Command Option 1 = ‘X’ for A1 Command

Using command option 1 = 'X' for an Adabas A1 command resets all fields not specified in the format buffer to their default value. In particular, the MU/PE counts are set according to the values contained in the format and record buffer. This feature was already available in Adabas v6.1, but not documented.

New with Version 6.3 SP1

ADABCK

A new sub-parameter RENUMBER has been introduced for file restore or overlay, this lets you restore the files with a new file number if the NONEW_PLOG option is used.

Caution: The check for the NEW_PLOG option was missing and was only introduced with V6.3 SP1 Fix 13. This error caused ADAREC REGENERATE errors. Therefore, if you still use a fix level < V6.3 SP1 Fix 13, please use ADABCK DUMP files only with the option NONEW_PLOG.

Adabas - Adabas Replication

Starting with Version 6.3 SP 1, the direct replication of Adabas files to other Adabas databases is supported via the Event Replicator for Adabas on Open Systems (separate product). For this purpose, some parameters have been added to the utilities ADABCK, ADADBM and ADAOPR. Partially, these new parameters are only to be used by the replication software - in which case the new parameters are undocumented. For more information on the Adabas-to-Adabas replication please refer to the new section *Adabas-to-Adabas Replication* in the *Administration* manual and the documentation of the Event Replicator for Adabas on Open Systems.

ADAOPR

Normally, obsolete entries in the client queue are released automatically when the client queue is full. With the new function ADAOPR FREE_CLQ,you can enforce the client queue cleanup before the client queue is full.

New Tool showcls (UNIX only)

The new tool showcls displays information about the client queue semaphore. It shows the status of the entries in the client queue:

- Free: The client queue entry can be used for a new client connection.
- Obsolete: The client queue entry can be reused after the next client queue cleanup.
- In use: The client queue entry is currently in use.

Modified Features

Utility Output

In Adabas Version 6.3, some changes have been made in the utility output directed to stdout for ADACLP as compared to Version 6.2. If you use scripts to check this output, please ensure that your scripts still work correctly with Adabas Version 6.3 before using Version 6.3 in your production environment.

Hold Queue Processing

In previous versions, the sequence in which commands, waiting for the same ISN, were executed was random. This has been changed in Adabas Version 6.3: the commands are now selected in the sequence in which they were entered in the hold queue.

Mainframe-Compatible RE Command

In previous versions of Adabas, the record buffer for the RE command was left unchanged following the end of the user data if the user data were smaller than the record buffer. Starting with Version 6.3, the record buffer is padded with blanks.

Improved Error Handling in ADADBM

ADADBM no longer terminates in the interactive mode following syntax errors.

"QUIT" and "EXIT" now recognized in FDT parsing Mode

In previous versions of Adabas, the keywords "QUIT" and "EXIT" did not terminate a utility when it was in the FDT parsing mode, for example, after specifying the FIELDS parameter, "QUIT" and "EXIT" are now recognized in the FDT parsing mode.

DRIVES Subparameter in ADABCK OVERLAY and RESTORE is obsolete

The subparameter DRIVES for ADABCK OVERLAY and RESTORE is now obsolete; the number of physical input devices is always the same as the value of DRIVES that was specified for ADABCK DUMP or EXU_DUMP.

New with Version 6.3 SP 1

ADABCK

While in previous Adabas versions, the ET synchronization for a dump at file level waited for the termination of all open transactions in the database, starting with Adabas Version 6.3 SP 1, the ET synchronization for a dump at file level only waits for the termination of those transactions which affect one of the dumped files.

Examples

The client examples are no longer stored under $ADADIR/$ADAVERS (UNIX) or %ALLUSERSPROFILE%\Software AG\%ADAVERS% (Windows). Starting with v6.3 SP 1 they are stored only under $ACLDIR/$ACLVERS (UNIX) or %ALLUSERSPROFILE%\Software AG\%ACLVERS% (Windows).

ADAM Key Algorithm

The algorithm used to compute the ADAM keys for an ADAM descriptor has been modified. This was done because it was assumed that ADAM descriptors were not used - there was an error in ADAFDU which did not allow an ADAM file to be defined based on an ADAM descriptor. However, if somebody nevertheless uses ADAM files, for example because they defined them with an Adabas version that did not yet contain this ADAFDU error, it is strongly recommend to unload the file with ADAULD, refresh it and load it again with ADAMUP. Otherwise, the performance advantages of ADAM files are lost, on the contrary, the performance will be even worse than the performance of non-ADAM files.

Command Duration displayed by ADACLP

Because many Adabas commands require less than one millisecond, the command duration displayed by ADACLP for command logs created with version 6.3 SP1 is displayed no longer in milliseconds, but in microseconds. The command time displayed for command logs created with older Adabas versions is still displayed in milliseconds.

New with Version 6.3 SP1 Fix 13

ADABCK

The default for [NO]NEW_PLOG for ADABCK DUMP files changed from NEW_PLOG to NONEW_PLOG. For ADABCK DUMP database the default is still NEW_PLOG.

Caution: Before V6.3 SP1 Fix 13, the default for a file dump was NEW_PLOG. In most cases, this change is of no consequence, but if you really need the PLOG switch, you must specify NEW_PLOG explicitly.

4 Limitations and Restrictions

Limitations and Restrictions in Adabas Version 6.3

The following limitations and restrictions exist in Adabas Version 6.3. Enhancements to resolve these limitations will be considered in a future update package.

- The support of the new ACBX interface is still subject to changes. In particular, the error handling, i.e. response codes and subcodes for errors, may be changed.
- The Adabas MC command and Adabas calls that use the multifetch option are not allowed with the new ACBX interface.
- Currently, only the printout of the old CLOG layout (CLOGLAYOUT=5) is supported. All commands are logged, but the information on commands using the new ACBX interface is incomplete.

Operating System and Hardware Restrictions in Version 6.3

Adabas Version 6.3 does not offer support for the following software platforms:

- HP-UX 11i Version 2
- AIX Version 5.3
- Solaris Version 9
- Linux Enterprise Server Version 10
- Windows 2003 Server
- Windows Vista

Operating System and Hardware Restrictions in Future Versions

Future versions of Adabas will not offer support for the following hardware and software platforms:

- Windows XP
- Windows Server 2008 R1
- 32-bit variants of operating systems that are available in both 32-bit and 64-bit mode

5 Converting a Version 3, Version 5, Version 6.1 or Version 6.2 Database to and from Version 6.3

When you want to upgrade a database from Version 3, 5, 6.1 or 6.2 to Version 6.3, you have two options:

- Create a new Version 6.3 database, and copy the data of the old database into the new database using ADABCK RESTORE or ADAORD IMPORT. In this case, the database conversion is done by ADABCK or ADAORD.
- Use the utility ADACVT to convert an existing database in place. The conversion using ADACVT is described in the following.

The conversion utility ADACVT is used to convert the structure of a database from Version 3, Version 5, Version 6.1 or Version 6.2 to Version 6.3. Once the database has been converted, the utility writes a SYNP checkpoint.

When converting a database, the nucleus must not be active or have an AUTORESTART pending. Furthermore, there must be no pending utility entries in the utility control block (UCB).

It is strongly recommended to dump the database using the backup utility ADABCK before the conversion. It is also recommended to check the consistency of the database using ADAVFY'S FIELD and INDEX functions. This can be done either before or after dumping the database.

The utility performs several I/Os and calls some other utilities. The time required for ADACVT to perform the conversion depends on the size of the index for files that have ISNSIZE of 2.

The utility will be able to restart after a crash in most cases. If the utility cannot restart after a crash, restore the original database from the backup and run the utility again.

This utility is a single-function utility.

Converting a Version 6.1 or Version 6.2 Database to and from Version 6.3

Converting to Version 6.3

ADACVT increases the structure level in the General Control Block (GCB).

Converting to Version 6.2

ADACVT decreases the structure level in the General Control Block (GCB).

Converting to Version 6.1

ADACVT checks to see if the new features (field options TS, SY)) are in use. If they are in use, backward conversion to Version 6.1 is not possible; if they are not in use, ADACVT decreases the structure level in the General Control Block (GCB) and changes the FDT back to the Version 6.1 structure (without the FDT update time stamp).

Converting a Version 5 Database to and from Version 6.3

Converting to Version 6.3

ADACVT increases the structure level in the General Control Block (GCB).

Converting to Version 5

ADACVT checks to see if the new features (field options TS, SY) are in use. If they are in use, backward conversion to Version 5.1 is not possible; if they are not in use, ADACVT decreases the structure level in the General Control Block (GCB) and changes the FDT back to the Version 5.1 structure (without the FDT update time stamp)..

Converting a Version 3 Database to and from Version 6.3

Converting to Version 6.3

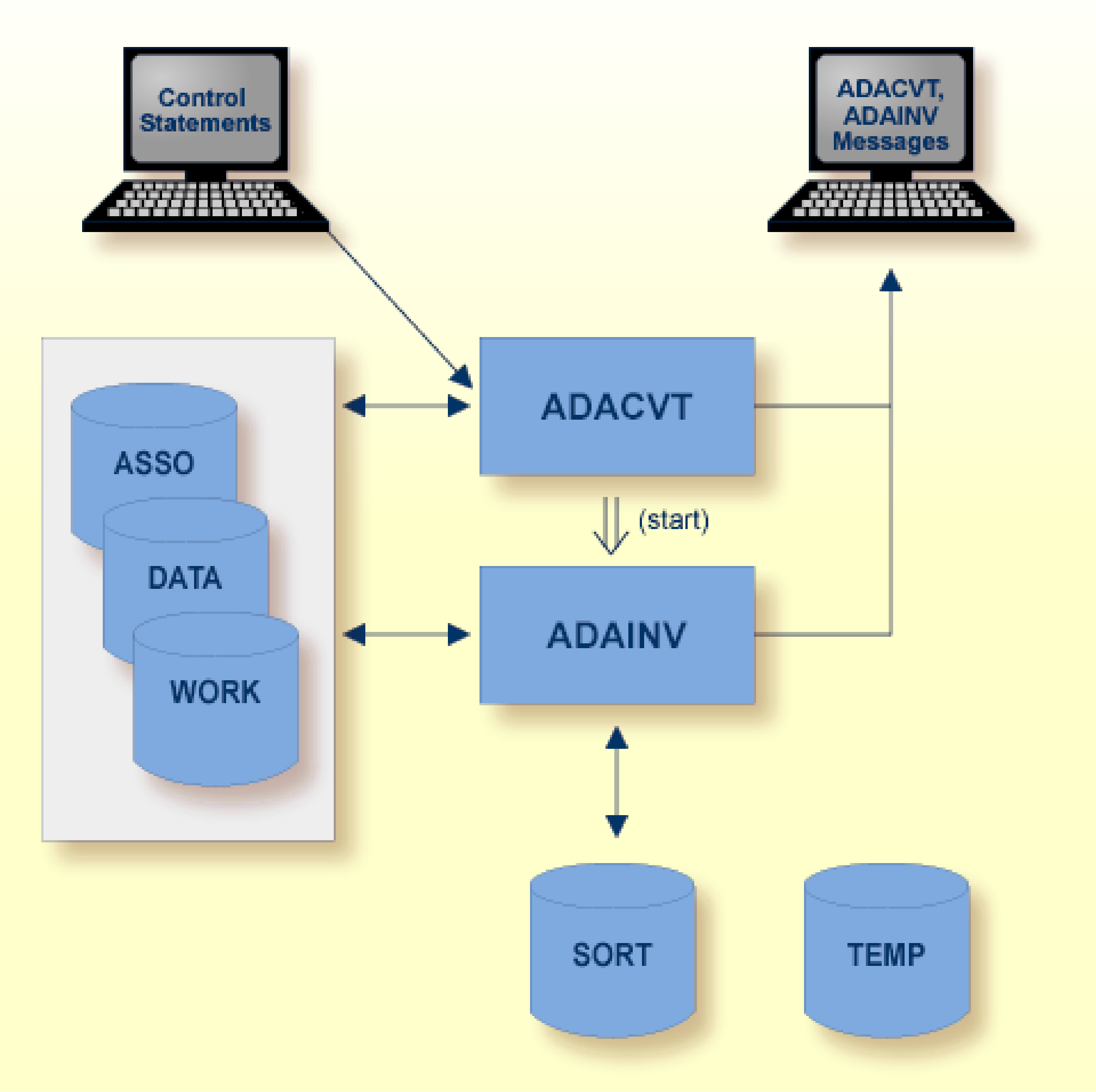
ADACVT increases the structure level in the General Control Block (GCB).

Converting to Version 3

It is not possible to convert a Version 6.3 database back to Version 3 with ADACVT.

If you want to access database files with Version 3.3 again, you must first unload and decompress the files with the Version 6.3 utilities ADAULD and ADADCU, then compress and load them with the Version 3.3 utilities ADAFDU, ADACMP and ADAMUP.

Procedure Flow



Checkpoint Information

During a conversion from Version 3 to Version 6.3, the content of the checkpoint file is deleted.

During a conversion between Version 5 and Version 6.3, a SYNP checkpoint is written.

ADACVT Control Parameters

DBID

```
DBID=number
```

This parameter selects the database to be converted.

CONVERT

```
CONVERT=keyword
```

This parameter specifies the structure of the destination version. 'keyword' defines the destination version and can take the following values:

V510
Converts a database from Version 6.3 to Version 5.1.

V610
Converts a database from Version 6.3 to Version 6.1.

V620
Converts a database from Version 6.3 to Version 6.2.

V630
Converts a database from Version 3, Version 5, Version 6.1 or Version 6.2 to Version 6.3.

ADACVT Error Messages

ACNSP **No space for new address converter, need 'number' blocks ('number'KB)**

Explanation There is no more ASSO space for building the new Address Converter. The message shows the total amount of space needed by the AC_CONVERT function.

Action Calculate free ASSO space in database and see how much missing. Then add a new ASSO container with sufficient space with ADADBM ADD_CONTAINER or check if you can use ADADBM DEALLOCATE. Alternatively, you can increase the ASSO container by using ADADBM with the EXTEND_CONTAINER parameter.

ADA245 **pending utility entries in UCB**

Explanation The UCB contains entries which indicate that one or more files or the database is locked.

Action Check the reason for these UCB entries.

CVDBS **database 'number' at unexpected structure level ('number')**

Explanation The current database is not a Version 3 or Version 5 database.

Action None: ADACVT cannot be used to convert unsupported database versions.

CVINVCV **only conversion to version 'vers' permitted**

Explanation Conversion to the destination version is not permitted. Please refer to the CONVERT parameter for information about the conversions that are supported.

Action Use one of the valid keywords.

CVTFDTTL **Conversion not permitted. FDT too long in file 'fnr'.**

Explanation The size of the FDT is larger than the maximum FDT size allowed in the target version of the conversion. Specifically, the FDT may be too large because more than 926 fields have been defined, which was the maximum up to Adabas Version 6.1.

Action Remove all files where the FDT is too large.

CVTOPTNO **Conversion not permitted. Not supported field definition in file 'fnr', field 'field'.**

Explanation ADACVT conversion to an older version is not allowed because of new features used.

Action Remove all files that use new Adabas features.

CVTREFNO **Conversion not permitted. Referential integrity in the file 'fnr'.**

Explanation ADACVT conversion to an older version is not allowed because referential integrity is not supported in that version.

Action Drop all referential integrity constraints.

NOTPOSS **database 'number' not converted to version 'number'**

Explanation The conversion failed. The reason for this has been explained in a previous message.

Action This depends on the reason why the conversion failed. See the corresponding message.

REQLEV **database 'number' already at structure level for version 'vers'**

Explanation The database is already at the requested level.

Action None.

STARTERR **Starting 'utility name' failed**

Explanation The given utility could not be started.

Action This is an informational message. The reason why the utility could not be started is given in subsequent error messages.

6 Adabas Version 6.3 for OpenVMS Itanium

In addition to the new and modified features mentioned above, Adabas Version 6.3 for OpenVMS Itanium also contains the following features that have been introduced or modified since the previous release.

General New Features

Shared Locks

In previous versions of Adabas, it was only possible to lock records exclusively or perform dirty reads; now shared locks are also available. The shared locks ensure that records are not updated during the current transaction.

Partial LOB Access

It is no longer necessary to read or update a complete LOB field; it is now also possible to read or update only a part of a LOB.

More than 926 Fields in an Adabas File

Adabas now also supports lower-case field names. This means that the maximum number of fields, including derived descriptors, in an Adabas file is increased to 3214. In earlier version, some utilities, for example ADAFDU, automatically converted field names to upper case. In order to be compatible with previous versions of Adabas, you must specify the parameter LOWER_CASE_FIELD_NAMES for these utilities; this prevents this conversion from being made.

Date/Time Fields

If a field is defined as a date/time field (field option DT), you can subsequently access it in several different, frequently-used date/time formats. When the TZ option is also specified, the field values are converted to the caller's local time zone.

System-Generated Fields

System-generated fields (field option SY) allow you to automatically generate information about the user or about the time when a record was created or last updated.

Subtransactions

If errors occur in applications, it can sometimes be helpful if it is possible to roll back only a part of a transaction, rather than the complete transaction. To this end, Adabas now supports subtransactions. Subtransactions have been implemented by making changes in the OP, BT and ET command.

Delayed Uniqueness Checks

When updating unique descriptors in a set of records of a file, temporary uniqueness errors may occur, these errors being corrected by the remaining updates. When subtransactions are enabled, the uniqueness checks are delayed until the end of the next subtransaction - this means that you no longer have to ensure that such temporary uniqueness errors don’t occur.

Referential Integrity

Adabas now supports referential integrity constraints - this means that any field in a file which is defined as a foreign key can only contain values from a primary files's primary key. This prevents you from defining records in a file containing a foreign key value that is not contained in the corresponding primary file as a primary key value.

Performance Optimization

With previous versions of Adabas, it was possible that short running commands could no longer be scheduled because complex, long-running commands blocked all threads, which forced the DBA to specify a large number of nucleus worker threads (ADANUC NT parameter) in order to avoid a very poor overall performance. In order to avoid this situation, now only 50% of Adabas threads can be used to process such long-running commands in parallel. Note that it is not always possible to estimate the runtime of a command correctly at the start of command processing.

Export Security Definitions

The security utility can now be used to export security data. The export file consists of security definitions that can be used as input for ADASCR for another database.

Timestamp of last FDT Update

Adabas now maintains a timestamp for the last FDT update. This timestamp helps you to check whether generated Adabas calls are still valid for a file - detailed checks are only necessary if the timestamp has changed.

New Options X and F for the LF Command

The new X and F options have been added to the LF command in order to provide addition information; this applies in particular to new features (date/time fields, system-generated fields, referential integrity, timestamp of last FDT update).

Modified Features

Utility Output

Some changes have been made in the utility output directed to SYS$OUTPUT for several utilities (e.g. ADAOPR, ADAREC, ADAREP) as compared to the previous version of Adabas. If you use scripts to check this output, please ensure that your scripts still work correctly with Adabas Version 6.3 before using Version 6.3 in your production environment.

Layout of the Adabas Log File

The length of the PID (process ID) field, which is the first field in an Adabas log file record, has been increased to 8 bytes.

Buffer Pool Manager

In previous Adabas versions, the buffer pool could sometimes overflow when there was a high update rate, with data blocks being modified faster than they could be written to disk - this caused the nucleus to crash in order to avoid situations in which data integrity could no longer be guaranteed. This behaviour no longer occurs.

More Precise Checkpoint Records

If more than one file was processed by a utility, previous Adabas versions generated one checkpoint record for all files, for example:

```
SYNP   ADAORD IMPORT=FILES
```

Starting with this version, a separate checkpoint is generated for each file that is processed, for example:

```
SYNP   ADAORD IMPORT=55
SYNP   ADAORD IMPORT=56
SYNP   ADAORD IMPORT=1056
```

Note: A checkpoint record is also created for LOB files, although they need not be specified but are processed automatically with the corresponding base file.

In addition, a SYNX checkpoint is now written for ADAFDU in online mode.

Utility Behaviour

Terminating a Utility

Starting with this version, the keyword "exit" can be used as a synonym for "quit" to terminate a utility.

Upper Case/Lower Case Field Names

The new parameter [NO]LOWER_CASE_FIELD_NAMES has been added to the utilities ADACMP, ADADBM, ADADCU, ADAFDU and ADAINV.

ADACMP

The new parameter SYFINPUT has been added. This parameter specifies the input used for the compression of system generated fields.

ADADBM

The new parameter CHANGE_FIELDS has been added. This parameter is used to modify one or more field specifications.

The new parameter DEFINE_REFINT has been added. This parameter is used to add a referential constraint to the file specified.

The new parameter DROP_REFINT has been added. This parameter is used to remove a referential constraint from the file specified.

The new parameter SYFMAX has been added. This parameter specifies the maximum number of values generated for a system generated multiple-value field.

ADAFDU

The new parameter SYFMAX has been added. This parameter specifies the maximum number of values generated for a system generated multiple-value field.

ADAMON

The new parameter [NO]DATETIME has been added. If this is set to DATETIME, each monitoring line with non-graphical output will be preceded by the current date and time.

ADAMUP

The new parameter RI_CONFLICT has been added. This parameter is used to indicate which action is to be taken if referential integrity is violated.

ADANUC

The default value for the parameter BFIO_PARALLEL_LIMIT has been changed to 50.

The default value of the ADANUC parameter LWP has been increased to 16 MB.

ADAOPR

The output of the ADAOPR DISPLAY parameter now includes additional or more adequate information.

ADAREC

The syntax of the ADAREC CLOSE parameter has been modified to optionally include the extent number of the Protection Log to be closed.

ADAREP

The new parameter CONSTRAINTS has been added. This parameter lists all referential contraints that are defined for the database.

Specifying ADAREP FILES now also displays the timestamp of the last FDT update.

ADASCR

The new parameter EXPORT has been added. This parameter is used to export the current security settings of the the database to a sequential output file.

7 Converting a Database from Adabas Version 4.1 (AXP) to Adabas Version 6.3 (Itanium)

The conversion package, which is delivered as a backup saveset, is used to convert the structure of a database from Version 4.1 (AXP) to Version 6.3 (Itanium).

Note: The backup saveset is not provided in the installation kit, and must be downloaded from Empower.

When converting a database, the nucleus must not be active or have an AUTORESTART pending. Furthermore, there must be no pending utility entries in the utility control block (UCB).

Caution: Only databases with block sizes that are multiples of 1K for all containers can be converted.

It is strongly recommended to dump the database using the backup utility before the conversion.

The format of the sequential files has changed since Adabas version 6.1. All sequential files are now in the "Stream LF" format, which is not compatible with the sequential files from Adabas version 4. In order to prevent incorrect file headers for newly-created files, Software AG strongly recommends that you remove old sequential files from the target directory before creating new ones.

Note: Existing applications must be recompiled on Itanium. The application does not have to be modified, since the Adabas control block and the call interface have not been changed, it just has to be linked with the shared library libadalnkx.

The following topics are covered:

Installation

The conversion package is delivered as a backup saveset. In order to use the conversion utilities, you must first restore the saveset to SAG$ROOT.

Contents of the Conversion Package

The conversion package contains the following files:

File Name	Purpose
ADA_AXP_CONVERT.COM	Procedure to set up the necessary environment and perform the conversion
ADACVT_V3.EXE	Conversion utility to convert a Version 4 AXP database into an intermediate state on Itanium
ADADBM_V3.EXE	Procedure to convert the Address Converter from 3-byte to 4-byte RABNs
ADAINV_V3.EXE	Procedure to reinvert all of the files in the database
LIBADAUTL.EXE, LIBADACOM.EXE, LIBSAGIPC.EXE, LIBSAGRMSIO.EXE	Libraries required by the conversion utility
ADAMSG.ENG	Message file

Prerequisites and Restrictions

The following prerequisites and restrictions apply to using the conversion package:

- You must be running Adabas Version 6.3 or above.
- The Adabas environment must be correctly set up - logical names such as ADADIR and ADAVERS and all symbols must be set.
- Before you start the conversion, you should use the procedure SAG$ROOT:[adabas]database to set up the environment of the database to be converted - this sets the logical names for all of the ASSO, DATA and WORK containers for the conversion procedure.
- You should ensure that there is sufficient free space in the ASSO containers before starting the conversion procedure:
  - The conversion procedure converts the Free Space Table from 3-byte RABNs to 4-byte RABNs, thus requiring additional space in the Free Space Table block.
  - The Address Converter for each file is also converted from 3-byte RABNs to 4-byte RABNs. Because the conversion procedure converts all of the AC blocks at once, the ASSO must have one third more free blocks than the total number of AC blocks used.
- The conversion procedure does not convert floating point data. Floating point data in VAX/AXP databases are stored in the so-called VAX floating point format, whereas Adabas stores floating point data on Itanium in the IEEE format. Any files in the database to be converted that contain floating point data must be unloaded, converted and then reloaded manually.
- As a result of a new format, system files will first be deleted by the conversion procedure and then created again later. If you want to reuse the data in the security and/or checkpoint file, you must unload it before running the conversion procedure, and then reload the appropriate field following the conversion.

- The conversion procedure has no restart capability. If an error occurs during the conversion, you must first restore the database in question before restarting the conversion. One possible source of errors might be insufficient free blocks in the ASSO containers.
- The conversion procedure will reinvert all files with the utility ADAINV. Depending on the number of file loaded, this may take a long time - consider using the optional parameter "noinvert" (see below) as an alternative to speed up the conversion.

Usage

To convert a database

1. Change your working directory to the directory in which the saveset was restored.
2. Call the conversion procedure by entering *@ada_axp_convert <dbid>[noinvert]*, where <dbid> is the number of the database to be converted. "noinvert" is an optional parameter which forces the conversion procedure to skip the reinvert step and which creates a command procedure (invert_all_<dbid>), which can then subsequently perform the reinvert in batch and in parallel.

Using the "noinvert" parameter

Call the conversion procedure by entering *@ada_axp_convert <dbid> noinvert*.

In order to customize the creation of the command procedures, some logical names are set by ada_axp_convert - you can edit ada_axp_convert.com and invert_all_<dbid>.com to set your own values.

You should start the database before running invert_all_<dbid>.com in order to avoid a 'checkpoint block overflow' error. invert_all_<dbid>.com submits a reinvert job (run_inv-batch.com) for each file in the database into a batch queue - the default name is ADACVT$BATCH, and this queue can be deleted once all conversions have been carried out.

Manual Migration

It is also possible to migrate a database manually. The migration process can be divided into two parts - steps to be performed on the Alpha side, and steps to be performed on the OpenVMS/Itanium side. These steps are described below.

To start the migration (on theAlpha side)

1 Unload the data using ADAULD and decompress the data using ADADCU (with the DCUFDT parameter).

Example

```
adauld db=<dbid>,file=<fileid>
define dcufdt <name.fdt>
define dcudta ulddta
adadcu dcufdt
```

2 Transfer the file DCUFDT to the OpenVMS/Itanium machine (ASCII format).

3 Transfer the file DCUOUT to the OpenVMS/Itanium machine (binary format) - this will be used as the <in filename> when running cvt_fmt.

To finish the migration (on the OpenVMS/Itanium side)

1 Run ADAFDU with the file DCUFDT as input.

Note: The parameters [NO]AUTOMATIC_ALLOCATION, ERASE, ISNSIZE and [NO]PLOG can no longer be used.

2 Run cvt_fmt (with ADANUC online) by entering the following in the command line:

```
cvt_fmt -n -l <in filename> <out filename> [dbid file]
```

3 Compress the data with ADACMP using the <out filename> produced in the previous step as CMPIN.

Example

```
define CMPIN <filename>
define CMPFDT <filename>
adacmp db=<dbid> update=<fileid> add
```

4 Load the data with ADAMUP, using the output of ADACMP from the previous step as the input.

Example

```
adamup db=<dbid> update=<fileid> add
```

The usage of cvt_fmt is as follows:

```
cvt_fmt [opt] <in filename> <out filename> [<dbid> <file>]
```

Note: <dbid> and <file> are only required in conjunction with the options -f or -n.

The following options are available:

-n
Newline format to exclusive length format

-f
Fixed format to exclusive length format

-h
Record unloaded from high-oder machine (default)

-l
Record unloaded from low-order machine

-e<fmt>
2 byte exclusive length format to fix format, no length, no newline (fmt f), or fix format, no length but newline (fmt n)

-r<order>
RMS format to exclusive length format, record unloaded from high-order machine (order h), or record unloaded from low-order machine (order l)

-i
Records including USERISNs

-p
Print version

-s
Records have NC status field

-c
Check and trace

8 Documentation Updates and Changes

The most recent product documentation, hotfixes and other useful information can be found in Empower. In addition, separate documentation CDs are produced at regular intervals.

Changes made since V6.2 SP1

Administration

A note concerning the behaviour of the keywords *QUIT* and *EXIT* in conjunction with the keyword LOWER_CASE_FIELD_NAMES has been added to *Using Utilities, Terminating a Utility*.

The documentation for the UNIX tool showgdt has been removed.

The description of the keyword SESSIONID for the definition option Sytem Generated (SY) has been revised and augmented.

The default translation tables used by Adabas have been added to *Appendix A*.

Commands

A table that explains the information returned in the 'S' element has been added to the section *Calling Adabas, Format and Record Buffer, Format Buffer Syntax*.

The explanation of the element `name L[mu_pe_index]` in the section *Calling Adabas, Format and Record Buffer, Format Buffer Syntax* has been updated.

The descriptions of the control block fields ACBCID and ACBXCID in the section *Adabas Control Block Structures* have been extended to include details of blank command IDs in both ASCII and EBCDIC.

The descriptions of the search buffers for the L3/L6 and L9 commands have been rewritten and moved to the section *Search and Value Buffers*. The original descriptions of the search buffers have been removed from the individual commands.

The information about the Command ID field in the control block of the RC command has been corrected from "-/A" to "F/U".

A note has been added to the section *Concepts and Facilities, User Types, Access-only* user in order to clarify the restrictions that apply to this type of user.

In the section *Search Buffer Syntax (Using Soft Coupling)*, the description of sfield has been been changed and augmented.

Installation

Starting with Version 6.2, Adabas only supported SUSE 11, the installation documentation has been changed to remove references to SUSE 10.

The software prerequisites for installing Adabas on UNIX and Windows platforms has been updated. HP-UX v2, AIX 5.3 and Solaris 9 are no longer supported.

Messages and Codes

The new message DSTNLS has been added.

The new message ALLERROVFL has been added.

The new message IOT has been added.

The new message IOTRES has been added.

The new message LABLINV has been added.

The new message RAWCORRUP has been added.

The new message SHMNOA has been added.

Subcodes 38, 39, 40, 41and 42 have been added to the nucleus response code 22. The explanation of subcode 37 has been changed.

An new reason (missing last-update-timestamp system generated field) has been added to the nucleus response code 44. Information about subcodes (and their meanings) for response code 44 has also been added to the description.

Utilities

ADACMP

A new note has been added to the description of the ADACMP parameter DST.

ADADCU

A new note has been added to the description of the ADADCU parameter DST.

ADAOPR

New notes explaining the locking of non-existent file numbers and of LOB files have been added to the ADAOPR parameter LOCK.