Enhancements

This chapter describes the Adabas 8.2 enhancements.

Category	**Enhancements**
General 8.2 Enhancements	*Extended Address Volume (EAV) Support*
	New and Changed Utility Functions
	New and Changed Operator Commands
	ADARUN Parameter Enhancements
	Direct Call Command Enhancements
	New LGBLSET Macro Parameters
	New Sample COBOL Programs
Adabas 8.2 SP3 Enhancements	*License Refresh and Verification Support*
	Improved ADASAV Volume Information on z/OS
	Session Autorestart Progress Messages
	Unique Descriptor Pool Minimum Size Increased
Adabas 8.2 SP2 Enhancements	*SMGT Response Code Handling*
	ADALNK Access to Multiple ID Tables on BS2000
	System Field Support
	Partial LOB Field Processing
	Shared Hold Status
	SMF Support (z/OS only)
	Command and Protection Log Alert Messages
	Printing Multifetch Statistics to a File
	LORECX DSECT Updates
	Logically Deleting and Undeleting Descriptors
	Maximum Length of Fixed Point Fields Increased
	Date-Time Edit Mask Enhancements
	Time Zone Support
	Daylight Savings Time Support
	RI Commands: Change Possibly Affecting Your Application Programs

Category	Enhancements
Adabas 8.2 SP1 Enhancements	*Logically Deleting and Undeleting File Fields*
	New Date-Time Edit Masks
	Lowercase Field Name Enhancement
	Expanded File Chain Enhancements
	Putting Files in Read-Only State
	Database Conversion Improvements
	Unique Descriptor (UQ) Option Removal
	Work Area Monitoring
	PLOG and CLOG Data Set Enhancements
	Spanned Record Logging
	New Adabas Event Log Support
	Response Code 145 (ADARSP145 - Hold Queue Entry) Event Enhancements
	Multiple CICS Task-Related User Exit (TRUE) Support
	CICS Installation Changes
	Performance Enhancements
	Nucleus 64-Bit Processing Support
	Adabas Buffer Description (ABD) Structure Changes
	Large Page Support
	z/OS SVC Dump Enhancements
	z/OS SVC Authorization Enhancements
	z/OS SVC Prelinked to SAF
	Reporting on Replication-Related Records in Work Part 1
	Printing Adabas Maintenance and SVC Status Information
	Replication of Security Definitions
	Address Converter Extent Handling Enhancement
	Adabas Delta Save Update: ADASAV MERGE Limitation Lifted
	Terminology and Name Changes

Extended Address Volume (EAV) Support

Adabas supports the use of extended address volumes (EAV) under z/OS. For a description of the Software AG requirements for this support, review the documentation at the following link: *http://www.softwareag.com/corporate/images/Software_AG_ETS_Product_Support_for_Extended_Address_Volumes__EAV_07222010_tcm16-43365.pdf.*

License Refresh and Verification Support

You can now refresh the license file currently used by the nucleus. This allows you to overwrite the current license with a new one or force the system to reread and reload its existing license. To support this functionality, Adabas 8.2 SP3 introduces a new operator command: LICREFRESH. For complete information, read *LICREFRESH Command*.

Improved ADASAV Volume Information on z/OS

Adabas 8.2 SP3 includes an enhancement to ADASAV SAVE that provides more accurate volume information in z/OS environments. In previous releases, due to the asynchronous nature of ADASAV I/O processing, a number of blocks written on one volume might have been reported by ADASAV as written on the preceding volume.

Important:
To enable this improved ADASAV volume information, apply optional Adabas zap AY823135 and IBM APAR OA33465. If you choose to apply AY823135, make sure you also apply IBM APAR OA33465. Installing AY823135 without IBM APAR OA33465 can cause the ADASAV SAVE backup to become corrupted. An ADASAV RESTORE would then fail (with system abend S002) and the database might then be unusable.

Session Autorestart Progress Messages

While performing a session autorestart after a nucleus failure, Adabas 8.2 SP3 now prints progress messages that indicate how much repair and recovery work has already been done and how much remains to be done. By default, these progress messages occur at the end of each phase of autorestart processing (backward repair, forward repair, autobackout). A new ADARUN parameter, INTAUTO, has been provided that allows you to specify the time interval (in seconds) between autorestart progress messages, thus allowing you to request that additional progress messages be printed for each phase of the autorestart process.

For complete information about the INTAUTO ADARUN parameter, read *INTAUTO Parameter: Time Interval Between Autorestart Progress Messages*.

New Sample COBOL Programs

Adabas 8.2 SP3 includes two new sample COBOL programs, COBADA8 and COBACI8. These programs are provided to illustrate how to make Adabas ACBX direct calls in a batch/TSO environment (COBADA8) and under CICS/TS (COBACI8). Their source modules should be compiled using the Enterprise COBOL compiler Version 3 Release 3 or later.

The COBADA8 member is provided in the ADA*vrs*.MVSSRCE library and the COBACI8 member is provided in the ACI*vrs*.MVSSRCE library.

Note:
These programs are provided "as is" and will not be supported by Software AG.

For more information about these sample programs, read *ACBX Examples*.

Unique Descriptor Pool Minimum Size Increased

Starting with Adabas 8.2 SP3, the minimum setting of the unique descriptor pool will be increased from zero to 5000 bytes. This size is established by the ADARUN LDEUQP parameter.

Note:
This new functionality is also available in Adabas 8.2 SP2 if you apply zap AN822111.

For complete information, read about the LDEUQP parameter in *LDEUQP Parameter: Length of Unique Descriptor Pool* .

SMGT Response Code Handling

The following changes, made in Adabas 8.2 SP2, affect how SMGT response codes are handled:

- PINRSP will now be loaded automatically at startup if the ADARUN SMGT parameter is set to "YES", similarly to ADAMXY.
- The output from the PIN routines can be directed to DDTRACE1 rather than DDPRINT if ADASMXIT has been updated by the user to specify this and if DDTRACE1 is provided in the nucleus startup JCL.

To activate this support, be sure to apply fix AN822020 and use the ADASMXIT from the ADA822.S001 data set.

Note:
If you are using Adabas 8.2 SP3 or a later version of Adabas, applying this fix is unnecessary as the fix has already been applied to the source for you.

Information about this processing and DDTRACE1 are provided for z/OS sessions in *Adabas Session Execution* and in *Installing The Error Handling And Message Buffering Feature*.

ADALNK Access to Multiple ID Tables on BS2000

In prior versions of Adabas, routing of data from a client to a database on BS2000 systems could only take place over a single ID table defined by the IDTNAME parameter in the following ways:

- As a parameter in the file attached to the link name DDCARD (ADARUN statements);
- As a parameter in the file attached to the link name DDLNKPAR (ADALNK statements); or

- By modifying the parameter ENVNAME in the macro B2CONFIG and assembling the SSFB2C parameter repository module.

In Version 8.2 SP2, it is now possible to route data to specific database IDs to other ID tables, and optionally, to change the database ID to another in the target ID table. For more information, read *ADALNK Access to Multiple ID Tables*.

System Field Support

Support is added in Adabas 8.2 SP2 for *system fields*. A system field is a field in an Adabas file whose value is automatically set by the Adabas nucleus when records are inserted or updated on the file. Optionally, you can specify that some system field values only be set when records are inserted. The following types of system fields are currently supported:

- Job name: The job name of the user inserting or updating a record.
- ETID: The eight-byte user ID of the user inserting or updating a record. This is the user ID set in the Additions 1 field of an OP (open) command for the user session.
- Session ID: The 28-byte user ID of the user inserting or updating a record.
- Session user: The last eight bytes of the 28-byte session ID or the user inserting or updating a record.
- Time: The date or date and time at which a record is inserted or updated.

Two new field options have been added to this release to be used when defining your system fields with the ADACMP COMPRESS utility function: SY and CR. The SY option should be used to identify the contents that should be stored in the system field; the CR option should be used to indicate that the system field value should only be set when the record is inserted into the Adabas file (it will not be modified when an update operation occurs). For more information about the SY and CR field options, read *Field Options*.

In addition, the ADALOD LOAD utility function can now specify the maximum number of occurrences for MU system fields allowed in a file using a new SYFMAXUV parameter. For more information, read *LOAD: Load a File* .

Finally, a new SYFMAXUV parameter is supported by the ADADBS MODFCB utility function that allows you to modify the maximum number of occurrences for MU system fields allowed in a file. For more information, read *MODFCB: Modify File Parameters* .

For complete information about system fields, read *System Fields*.

Partial LOB Field Processing

Adabas 8.2 SP2 introduces new partial large object field (LOB field) processing. Your application programs can now read or write parts of LOB values, or *LOB segments*. In addition, in a series or read or write requests for a LOB field, your applications can control the position of the next LOB segment within the LOB value. This support introduces the following new features:

- New *LOB segment notation* is provided for you to use in the format buffer of a command to select the specific portion (segment) of the LOB value you want to read or write in the call. For more information about this notation, read *Field Selection Criteria* .
- A new command option, L, can be specified in Command Option 2 of ACB or ACBX calls for L1/L4 or A1 commands. This command option requests that the current position in the LOB value be tracked in the ISN Lower Limit (ISL) field of the ACBX. In conjunction with this, a new current positioning notation (an asterisk) can now be used in the *LOB segment notation* of a call to indicate that the call should start at the current position (the position where the last call left off) of the LOB value.

  These features allow you to read or write the LOB value in segments using multiple calls. The current positioning notation (an asterisk) allows you to easily indicate where the next call should start, without changing the format buffer for every call.

  For more information, read *A1 Command: Update Record* and *L1 and L4 Commands: Read / Read and Hold Record* .

For complete information about processing partial LOB fields and all of these new features, read *Processing LOB Segments*.

Shared Hold Status

Adabas 8.2 SP2 introduces a new kind of competitive updating control called *shared hold status*. You can use shared hold status to lock data records in shared mode, rather than in the existing exclusive hold status. This allows your database users to read the same record in parallel transactions, but ensures that no one can update the record concurrently.

Using shared hold status, your users can protect large object values from concurrent updates without locking out other users who may need to read the same LOB value or other LOB values in the same record. It also allows your users to protect the records they read against concurrent updates for specific periods of time:

- For the duration of the read command;
- When the next record in a sequence is read;
- When the user’s transaction ends;
- Indefinitely.

This new functionality introduces four new command options for the BT, ET, HI, L4 - L6, RI, and S4 commands, although all four new command options do not necessarily apply to all of these commands. These new options are specified in the command option 3 field of ACBX only calls for the associated command. Each command option specifies a different shared hold time period.

- Option C puts the record in shared hold status for the duration of the read operation. It ensures that the version of the record being read has been committed by the last updater. This option is available for the L4, L5, L6 and S4 commands. For the S4 command, the shared hold remains in place for the duration of the read operation.

- Option Q puts the record in shared hold status until the next record in the read sequence is read or the read sequence or transaction is terminated, whichever happens first. It ensures that the record being read cannot be updated concurrently until the next record in the sequence is read (or the transaction is terminated). This option is available for the L4 (when command option 2 is set to "N"), L5, L6 and S4 commands.

- Option S puts the record in shared hold status until the end of the transaction. It ensures that the record being read cannot be updated concurrently until the transaction is terminated. This option is available for the HI, L4, L5, L6, RI and S4 commands.

- Option H keeps a record in shared hold status indefinitely (until the next ET or BT command). This option is available for the BT and ET commands. Records in shared hold status at the time of the BT or ET command are kept in shared hold status beyond the end of the transaction until another ET or BT command is issued (without this H option or the prefetch or multifetch options). Any records in exclusive control are also changed to shared hold status beyond the end of the transaction.

If the same record is placed in shared hold status more than once (using the C or S options or the Q option for different read sequences), it stays in shared hold status until all of the specified hold lifetimes have expired.

For more information about these command options and their detailed functioning in each command, read about the individual commands in *Commands*.

Resource deadlock may occur when users are simultaneously holding records and waiting for records held by other users. Normally, Adabas detects a user deadlock situation and returns a response code to the user who would create the deadlock if placed in wait status. However, the deadlock may not be detected if the users involved are serviced by different nuclei in a cluster. Even in a single nucleus, a deadlock may not be detected if a record involved is being held as a shared resource by more than one user. If a deadlock is not detected, all users involved will wait until the first one of them exceeds the transaction timeout (TT) threshold.

Putting more records in hold status (including shared hold status) may decrease the possible parallelism of transaction processing (and thus, the performance of multiuser applications) and increase the likelihood of deadlocks between transactions (where each transaction holds a record that the other wants).

Using shared hold status affects the ADARUN NH and NISNHQ parameter settings. Each shared hold request with a different command ID (CID), as well as a (shared or exclusive) hold request without a CID, is counted against the NISNHQ and NH limits. This affects application programs that make use of the new Q option for sequential reads. For example, if a program reads records with the Q option and then updates every record, the shared hold operations from the use of the Q option and the exclusive hold operations from the update commands are counted separately. Such a program might need NISNHQ and NH limits set twice as large as when the Q option is not specified.

For complete information about shared hold status, read *Shared Hold Status*.

SMF Support (z/OS only)

For z/OS environments only, support for IBM System Management Facility (SMF) records has been added to Adabas 8.2 SP2. SMF recording is widely used in IBM and ISV software products. Adabas provides support for SMF records in two ways:

1. Adabas nuclei now generate SMF records to record performance and resource consumption. Existing Adabas statistics are the basis for standard Adabas records and a new user exit permits you to include your own data.

2. Adabas nuclei can now be registered with SMF to record CPU consumption in IBM-defined type 89 records.

Seven new ADARUN parameters are introduced as a result of this support:

- The SMF parameter can be used to enable or disable Adabas SMF recording.

- The SMF89 parameter controls whether the Adabas nucleus registers with z/OS for type 89 SMF records.

- The SMFRECNO parameter can be used to define the Adabas SMF record number for user-defined records.

- The SMFINTERVAL parameter can be used to specify whether interval SMF records should be generated and, if so, how often.

- The SMFSUBSYS parameter can be used to associate the SMF records with an IBM or user-defined subsystem.

- The SMFDETAIL parameter can be used to specify the detail sections of Interval and Termination records that should be included (if any).

- The UEXSMF parameter can be used to identify the name of the SMF user exit module that should be used.

For complete information about these ADARUN parameters, read *ADARUN Parameter Directory* .

This enhancement also introduces six new operator commands: SMFRECNO, SMFINTERVAL, SMFSUBSYS, SMFDETAIL, SMFDETAILADD, and SMFDETAILDEL. These commands can be used to alter the comparably named ADARUN parameter settings in a running Adabas session. The SMFDETAILADD and SMFDETAILDEL commands allow you to add and remove specific detail sections of Interval and Termination records from the list of records that should be recorded. These commands can be issued only as operator commands; they cannot be issued in an ADADBS OPERCOM utility function run. For complete information about these commands, read *Operator Commands*.

Finally, this enhancement allows you to provide a user exit that adds a user-specified detail section to the Adabas SMF record. It is a separate load module whose name is provided in the UEXSMF ADARUN parameter. For complete information, read *SMF User Exit*.

For complete information about Adabas SMF records, read *Adabas SMF Records*.

Command and Protection Log Alert Messages

Adabas 8.2 SP2 introduces the ability to optionally have Adabas generate alert messages indicating the status of the command log (CLOG) and the protection log (PLOG). The alert messages are written when Adabas detects that a PLOG or CLOG data set has not been copied within a specified time interval. Thereafter, an alert message is written at regular intervals as long as there is a PLOG or CLOG data set waiting to be copied. These alert messages are issued until the PLOG or CLOG data set is copied or until no logs remain to be copied. Your user exits 2 or 12 can be set up so that the messages appear providing a comprehensive trace of activity so you can resolve and problems with the PLOG or CLOG.

This feature helps you prevent the potential problem of Adabas overwriting an older CLOG or PLOG data set without waiting for direction from the user. Ordinarily, when a PLOG or CLOG data set fills up, the Adabas nucleus examines the return code from user exit 2 or user exit 12 to determine whether it should wait before overwriting an older PLOG or CLOG data set with new information. If there is no user exit 2 or 12, Adabas does not wait, but simply overwrites the older log. In prior releases of Adabas, no alert messages were issued before this happened. This new feature provides you with these alert messages. If this LOGWARN parameter is set to a value other than zero, the statuses of the CLOG and PLOG data sets are analyzed at the frequency specified by the parameter, warning about their status and providing you with the information you need to prevent any overwriting of older logs.

The production of CLOG and PLOG alert messages is optional and is determined by the setting of a new LOGWARN ADARUN parameter. A value of zero (0), the default, indicates that no CLOG and PLOG status checks should occur and that no alert messages should be produced. Any other valid LOGWARN value represents the interval (in seconds) at which the PLOG and CLOG status is checked and resulting alert messages are produced. When a value other than zero (0) is specified for LOGWARN, a valid user exit 2 or user exit 12 must also be specified. For complete information about the LOGWARN parameter, read *LOGWARN : CLOG and PLOG Alert Message Interval*.

In addition, a new LOGWARN command has been added that allows you to alter the setting of the LOGWARN ADARUN parameter for a given Adabas session. This command can be run from the operator console or via the ADADBS OPERCOM utility function. For complete information, read *LOGWARN Command*.

Printing Multifetch Statistics to a File

Effective with Adabas 8.2 SP2, multifetch statistics can be printed when a CL command is issued if both of the following conditions are met:

- The new PREFSTDD ADARUN parameter is specified in the startup JCL. This new parameter identifies the job control statement label (DD name) that defines the location to which the multifetch statistics should be written.
- The nucleus startup JCL includes a valid job control statement with the label name identified by the PREFSTDD ADARUN parameter.

For complete information, read *PREFSTDD: Multifetch Statistics DD Name*.

LORECX DSECT Updates

In Adabas 8.2 SP2, the fields LOX1STYP and LORSTYPE in the LORECX DSECT (distributed in the Adabas source library) have been altered so that the CLOGs reflect accurate information describing which search algorithm was used.

Logically Deleting and Undeleting Descriptors

Adabas 8.2 SP2 allows you to logically delete and undelete a descriptor from a database file. Logically deleting a descriptor from a file removes the descriptor from the file, but retains the field data in the database. A logically deleted descriptor cannot be used as a search descriptor. This functionality is provided using two new ADADBS utility functions: DELDE (logically delete a descriptor) and UNDELDE (undelete a logically deleted descriptor). For more information, read *DELDE: Logically Deleting a Descriptor* and *UNDELDE: Undeleting a Logically Deleted Descriptor*.

Logically Deleting and Undeleting File Fields

Adabas 8.2 SP1 allows you to logically delete and undelete a field from a database file. Logically deleting a field from a file removes the field from the FDT, but retains the field data in the database. Logically undeleting a field that was previously logically deleted reinstates the field in the FDT. This functionality is provided using two new ADADBS utility functions: DELFN (delete a field) and UNDELFN (undelete a field). For more information, read *DELFN: Logically Delete Files* and *UNDELFN: Logically Undelete Files*.

Maximum Length of Fixed Point Fields Increased

The maximum length of fixed point fields (F format fields) was increased to eight bytes in Adabas 8.2 SP2. Fields defined with format F can now be two, four, or eight bytes long.

For more information, read *Field Length and Data Format*.

New Date-Time Edit Masks

New date-time edits masks are introduced in Adabas 8.2 SP1. For a complete reference of these date-time edit masks, read *Date-Time Edit Mask Reference*.

- A new DT field option is introduced that allows you to specify the edit mask for a field definition. For more information, read *Date-Time Edit Mask Option DT*.

  This DT option is supported by ADACMP utility FNDEF field definition statements. For more information, read *DT: Date-Time Edit Mask Field Option*.

- Date-time edit masks are also supported in the format and search buffers of direct calls. You can add date-time edit mask to fields with formats of B, F, P, or U in a loaded file. For more information, read *Date-Time Edit Mask Processing in Format and Search Buffers*.

Date-Time Edit Mask Enhancements

The following date-time edit mask enhancements have been made in Adabas 8.2 SP2:

- The new DT option, introduced in 8.2 SP1, is now supported by the ADADBS NEWFIELD function. In ADADBS NEWFIELD functions, the DT option is supported in FNDEF statements, allowing you to specify a date-time edit mask for a new field.

  For more information, read *NEWFIELD: Add New Field* .

- Changes have been made to the ADADBS CHANGE utility function so you can now change the time zone and the date-time edit mask. For complete information, read *CHANGE: Change Standard Length or Format of a Field* .

- Changes have been made to the ADADBS ONLINVERT utility function so you can now define a date-time field (with edit mask and time zone) as a descriptor field. For complete information, read *ONLINVERT: Start Online Invert Process* .

Time Zone Support

Adabas 8.2 SP2 supports variations in time zones. The following updates have been made:

- A new library, the Adabas time zone library, is provided in your installation.

  A new TZINFO member is provided in your Adabas source library. It lists the time zone support in Adabas and references members of the new time zone library containing the actual code used for time zone conversions. For more information about the Adabas time zone library, refer to the Adabas installation information for the appropriate system platform. For information about the TZINFO library member, read *Supported Time Zones*.

  Adabas uses the time zone data taken from the public-domain tz database, (also called the *zoneinfo* or *Olson* database) to create its own time zone library.

- A new TZ parameter can now be specified in the ADACMP COMPRESS and DECOMPRESS utility functions to specify the local time zone. As records are compressed and loaded into the file, the date-time data is converted from the specified time zone to UTC time (Coordinated Universal Time, also known as Greenwich Mean Time); likewise as records are decompressed and read from a file, the date-time data is converted from UTC time to the specified time zone. Date-time field data is always stored in UTC time in an Adabas file. For more information, read *COMPRESS: Compress an Adabas File* and *DECOMPRESS: Decompress an Adabas File* .

  In addition, review important information about the daylight savings time enhancements provided in this release. If a time zone uses daylight savings time, you must be sure to store and retrieve the daylight savings indicator with your date-time data or there will be no way to distinguish date-time values in the hour before the time is switched back to standard time.

- A new TZ (time zone) record buffer option is available for the OP command. You can use this option to set the time zone of an individual user session. If no time zone is specified for a user session, response 55 (ADARSP055) subcode 30 is issued when a field with the TZ option is accessed or updated. For more information about using the TZ record buffer option in an OP command, read about the OP command *Record Buffer*.

- A time zone field option, TZ, can now be specified to identify date-time fields that should be presented in the user's local time and stored in UTC time, thus allowing for differences in time zones. For more information, read *Time Zone Option TZ.*

- Changes have been made to the ADADBS CHANGE utility so you can now change the time zone and the date-time edit mask for a field. For complete information, read *CHANGE: Change Standard Length or Format of a Field* .

- Changes have been made to the ADADBS NEWFIELD utility so you can now specify the time zone for a new field. For complete information, read *NEWFIELD: Add New Field* .

- Changes have been made to the ADADBS ONLINVERT utility function so you can now define a date-time field (with edit mask and time zone) as a descriptor field. For complete information, read *ONLINVERT: Start Online Invert Process* .

Note:
The default number of time zones that can be used in a nucleus session is limited to 8 and the time zone pool size default to 32K. However, Software AG allows you to increase these numbers (refer to the appropriate zap in member ZAPOPT of the distributed source library) so that all time zones can be loaded.

Daylight Savings Time Support

Adabas 8.2 SP2 provides daylight savings time support. If the current time zone uses daylight savings time, Adabas provides the following support:

- A new DST parameter can now be specified in the ADACMP COMPRESS and DECOMPRESS utility functions to indicate that date-time data includes a daylight savings time indicator. If a time zone uses daylight savings time, you must be sure to store and retrieve the daylight savings indicator with your date-time data or there will be no way to distinguish date-time values in the hour before the time is switched back to standard time.

  The DST parameter requires that the TZ parameter be set in the same ADACMP run.

  For more information, read *COMPRESS: Compress an Adabas File* and *DECOMPRESS: Decompress an Adabas File* .

- A new daylight savings time indicator, D, can be specified in format and search buffers to indicate that a daylight savings adjustment is required for a date-time field with the TZ option specified. For complete information, read *Daylight Savings Indicator (D)*.

RI Commands: Change Possibly Affecting Your Application Programs

Adabas 8.2 SP2 on mainframes no longer releases records unconditionally. If your application issues an RI command for a record that has been updated in the current transaction, Adabas will now return response code 113 (ADARSP113); if ISN=0 was specified, response code 2 (ADARSP002) is returned. This change in behavior may affect how your application programs are coded.

This change was necessary to accommodate Adabas support for shared hold status and partial large object (LOB) field processing. It prevents problems that might occur if updated records are released from hold, the transaction is subsequently backed out, and there are competitive updates to the same records. Also, this change ensures that Adabas on mainframe and open systems behave in the same manner regarding RI commands.

Application programs that issue RI commands may be affected, if they perform updates and if they try to release updated records from hold status. Such application programs should be adjusted to either not issue RI commands for updated records or suppress the resulting response code 113 (ADARSP113). In addition, if your installation has Natural installed, you must apply Natural zap NA76045 for downward compatibility, which is available using Empower.

For complete information about the RI command, read *RI Command: Release Record* .

Terminology and Name Changes

The following terminology and name changes have occurred in Adabas 8.2 SP2:

- UEXITB, also known as user exit B (the before or pre-command user exit), has been renamed LUEXIT1 or link user exit 1 in z/OS and z/VSE environments.

- UEXITA, also known as user exit A (the after or post-command user exit), has been renamed LUEXIT2 or link user exit 2 in z/OS and z/VSE environments.

Lowercase Field Name Enhancement

Adabas 8.2 SP1 introduces the ability to create field names that contain lowercase alphabetic characters. With this change, the maximum number of field definitions for a file has increased from 926 to 3214. Thus, fields WM, wm, wM, and Wm are now all valid field names and now represent four unique fields.

Note:
Lowercase fields will not display correctly (they will be converted to uppercase) if you use the ADARUN parameter settings MSGCONSL=UPPER, MSGDRUCK=UPPER, or MSGPRINT=UPPER.

Expanded File Chain Enhancements

With Adabas 8.2 SP1 you can dynamically insert and remove files from an expanded file change using a new ADADBS function, EXPFILE. For more information on this utility function, read *EXPFILE: Insert or Remove Files in an Expanded File Chain*.

Putting Files in Read-Only State

Adabas 8.2 SP1 allows you place specific database files in read-only state while you maintain them. Your database users will be able to read data from the read-only files while your system administrators perform maintenance on those files. In addition, other files in the database, not in read-only status, can be read and updated as normal.

This support is provided in the form of the following Adabas utility enhancements:

- A new READONLY parameter has been added to the ADASAV RESTONL FILES, ADASAV RESTONL FILES, ADASAV RESTONL GCB, ADASAV RESTORE FILES, ADASAV RESTORE FMOVE, ADASAV RESTORE GCB,ADAORD STORE, and ADALOD LOAD utility functions. READONLY can be used to change the read-only status of a database file.

- A new FILEREADONLY parameter has been added to the ADADBS MODFCB utility function. FILEREADONLY can be used to indicate whether an Adabas database file should be placed in readonly status (where it can be updated only by Adabas utilities) or in normal status (where it can be updated in any normal manner).

  Notes:

  1. If the FILEREADONLY parameter is specified, only the FILE parameter (which is required) and the optional PASSWORD and TEST parameters can be specified in the same ADADBS MODFCB run.

2. The ADADBS DSREUSE, ISNREUSE, NEWFIELD, RELEASE DE, and RENAME utility functions do not need to lock the file for their use; these functions can perform their processing in parallel with active users. This means that you do not need to set a file in read-only status (FILEREADONLY=YES) if you will be performing these other ADADBS functions only.

For complete information, review the descriptions of the READONLY and the FILEREADONLY parameter as they are used by each these utility functions in *Utilities*.

Database Conversion Improvements

You now no longer need to run multiple ADACNV utility runs to convert or revert an Adabas database from one version to another. Adabas 8.2 SP1 includes an updated ADACNV utility that will convert or revert a database in a single step.

For complete information, read *ADACNV Utility: Database Conversion* .

Unique Descriptor (UQ) Option Removal

Adabas 8.2 SP1 introduces the ability to remove the UQ (unique descriptor) option from a descriptor field definition using the ADADBS CHANGE utility function. A new option, NOUQ, is introduced for the OPTION parameter. When you run the utility specifying OPTION=NOUQ for a descriptor field, the UQ setting is removed from the descriptor field.

For general information about unique descriptor fields, read *Descriptor Options DE, UQ, and XI* . For specific information about the new NOUQ option in the ADADBS CHANGE utility function, read *CHANGE: Change Standard Length or Format of a Field*.

Work Area Monitoring

Adabas 8.2 SP1 allows you to monitor the use of parts 1, 2, and 3 of the Work data set. This information allows you to proactively adjust the sizes of these areas and ensure 24-by-7 access to your databases. It also allows you to reduce the risk of receiving response code 9 (ADARSP009), subcode 15 errors while using your Adabas database.

This support is available using the DRES operator command. This support is also available via the ADADBS OPERCOM function. The information is also provided in Adabas Online System and Adabas Manager screens.

Note:
The maximum pool value of Work part 1 is derived from the LP parameter. It corresponds to the maximum number of blocks a transaction can spend on Work Part 1 before Adabas decides to back it out.

For more information about this new information, read *DRES Command*.

PLOG and CLOG Data Set Enhancements

Adabas 8.2 SP1 introduces several enhancements to its handling of protection log (PLOG) and command log (CLOG) data sets:

- A new NPLOGBUFFERS ADARUN parameter has been introduced that you can use to increase the number of PLOG protection I/O buffers, allowing multiple PLOG blocks to be written in a single write I/O to the PLOG and improving the performance of applications that were formerly constrained by writes to the PLOG. For more information, read *NPLOGBUFFERS : PLOG Protection I/O Buffer Control* .

- The ADARES CLOG and ADARES PLOG utility functions have been expanded to check the PPT at the end of the CLOG or PLOG run to determine whether any additional CLOG or PLOG data sets need to be copied. If so, the ADARES utility invokes user exit 2 or 12, as appropriate, for each uncopied CLOG or PLOG data set it detects. For complete information, read *CLCOPY: Copy Dual Command Log* and *PLCOPY: Copy Protection Log to Sequential Data Set*.

- The ADADBS utility introduces four new functions (ADDCLOG, ADDPLOG, DELCLOG, and DELPLOG) that you can use to dynamically add and delete CLOG and PLOG data sets without terminating your current nucleus session. Running these utility functions is invalid when Adabas is running with dual CLOGs or PLOGs.

  For complete information, read *ADADBS Utility: Database Services* .

- You can now log the extended I/O list to the CLOG data sets, for both CLOGLAYOUT=5 and CLOGLAYOUT=8. A new ADARUN parameter, LOGVOLIO, is provided so you can specify this functionality for an Adabas nucleus.

  You can also use two new operator commands to dynamically control this logging.

  1. Use the new LOGVOLIO operator command on the console (or in an ADADBS OPERCOM utility function run) to dynamically start logging of the extended I/O list. For more information, read *LOGVOLIO Command* .

  2. Use the new NOLOGVOLIO operator command on the console (or in an ADADBS OPERCOM utility function run) to dynamically stop logging the extended I/O list. For more information, read *NOLOGVOLIO Command*.

- A new ADADBS utility function, REACTLOG, now allows you to reactivate command logging in an active nucleus where it had been disabled previously as a result of an I/O error. The cause of the I/O error needs to be corrected before running this utility function or command logging will simply fail again and will not be reactivated.

  For complete information, read *REACTLOG: Reactivating Command Logging*.

- A new SRLOG ADARUN parameter allows you to indicate how spanned records should be logged to the protection logs. Entire records (ALL), modified segments of spanned records (UPD), or entire records up to and including the modified segments (PART) can be logged. The default is UPD. For more information, read *SRLOG : Controlling Spanned Record Logging* .

Spanned Record Logging

Adabas 8.2 SP1 allows you to write complete spanned records to the PLOGs. In the past, spanned records were only partly written to the PLOGs, making them impossible to decompress and use.

- A new SRLOG ADARUN parameter allows you to indicate how spanned records should be logged to the protection logs. Entire records (ALL), modified segments of spanned records (UPD), or entire records up to and including the modified segments (PART) can be logged. The default is UPD. For more information, read *SRLOG : Controlling Spanned Record Logging* .

  The SRLOG parameter replaces the PLOGSEGSDS ADARUN parameter; the PLOGSEGSDS parameter has been dropped in this release.

- The ADASEL utility has been enhanced with three new SET GLOBALS parameters to support the decompression of spanned records from the PLOG: SPANREC, MAXLOGRECLEN, and LWP. Each of these parameters is described in *SET GLOBALS Parameter*.

  In addition, a SPANREC parameter has been added to the ADASEL OUTPUT instruction and can be used in an ADASEL OUTPUT instruction to indicate that new spanned record SELH and SELC output headers should be used for all EXPA*n* output if there is the possibility that the output decompressed spanned records will exceed the physical record length limitation. DSECTs for the SELH and SELC headers can be found in the Adabas source library. For complete information, read *OUTPUT Instruction*.

- The ADACDC utility has been enhanced with two new parameters: SPANREC and MAXLOGRECLEN. Each of these parameters is described in *Running the Utility* .

  The SPANREC parameter indicates that new spanned record CDCH and CDCN output headers should be used for all CDCOUT output if there is the possibility that the output decompressed spanned records will exceed the physical record length limitation. DSECTs for the CDCH and CDCN headers can be found in the Adabas source library.

New Adabas Event Log Support

Adabas 8.2 SP1 introduces a new Adabas event log that can be used to log database events. In this release, only response code 145 (ADARSP145) events are logged to the Adabas event log. Each entry in the log is currently 128 bytes long, although this may change in future releases.

The size of the Adabas event log for a database is controlled by the INFOBUFFERSIZE ADARUN parameter. In addition, the entries in the Adabas event log can be displayed using a new ADADBS DEVENTLOG utility function.

Response Code 145 (ADARSP145 - Hold Queue Entry) Event Enhancements

Several enhancements have been made in Adabas 8.2 SP1 to support response code 145 (ADARSP145) events. Response code 145 (ADARSP145) occurs when a hold queue entry is not available for an N1or N2 command or when a command is issued that attempts to put a record on hold but cannot because the record is held by some other user. In this release of Adabas, events triggering response code 145 (ADARSP145) are now logged in the:

- Adabas event log, if an Adabas event log has been allocated. The Adabas event log size and allocation is controlled by the INFOBUFFERSIZE ADARUN parameter for a database.

- command log, if CLOGLAYOUT=8 is specified. A new record type (x’000D’) is now produced in a CLOGLAYOUT=8 command log to accommodate response code 145 (ADARSP145) events.

In addition, a special L2 command call can now be issued if you receive a response code 145(ADARSP145). When you execute the L2 command with `ISN=0`, `FNR=-4(X’FFFC)`, and an appropriate format buffer, you can obtain information about the user currently holding the ISN when the response code was issued. For complete information on this call, read *Special L2 Command Call for Response 145*.

Multiple CICS Task-Related User Exit (TRUE) Support

Adabas 8.2 SP1 supports the installation of multiple CICS task-related user exits (TRUEs) and Adabas application stubs from a single execution of the ADACIC0 installation program. Multiple TRUEs allow your site to tailor different Adabas CICS execution options in the same CICS region with a centralized installation procedure and software. For complete information, read *Installing Adabas with CICS under Adabas 8*, in the *Adabas z/OS Installation Guide*.

CICS Installation Changes

Additional CICS installation changes have occurred in Adabas 8.2 SP1. We no longer deliver ADACIRQ with Adabas on z/OS and z/VSE systems. The CICS input transient data queue is no longer required and the DDLINK input is no longer required. These have been replaced by the Adabas CICS installation options table (member ACIOPT) and the Adabas CICS names module (member ACINAMES). For more information, read *Installing Adabas with CICS under Adabas 8*, in the *Adabas z/OS Installation Guide*.

Performance Enhancements

Various updates have been made in Adabas 8.2 SP1 to ensure it continues to operate at peak performance:

- The performance (CPU consumption) of opening new Adabas user sessions, when there are already thousands or tens of thousands of concurrent users, has been improved.

- The performance of large object (LOB) updating has been improved by eliminating redundant serialization. All nuclei now reduce their use of the unique descriptor pool for LOB updates and customers using LOBs from Adabas 8.1 may be able to reduce the unique descriptor pool size (ADARUN parameter LDEUQP). Adabas Cluster Services and Adabas Parallel Services configurations no longer acquire external unique descriptor locks for LOB updates.

- Design changes have been made to the nucleus so that the blocks of a new address converter (AC) extent are now formatted asynchronously rather than synchronously. This allows a database to continue processing without waiting for all of the new AC extent blocks to be formatted.

- The new V64BIT ADARUN parameter may be used to improve the performance of your Adabas nucleus sessions. V64BIT allows the nucleus to use virtual storage above the 2 gigabyte bar, freeing virtual storage below the bar for other uses.

- For systems running in 64-bit mode (ADARUN parameter V64BIT is set to YES), the new ADARUN parameter LARGEPAGE allows the nucleus to use large pages, which may particularly improve the performance of long-running, memory-access intensive applications.

- The new NPLOGBUFFERS ADARUN parameter can be used to increase the number of PLOG protection I/O buffers, allowing multiple PLOG blocks to be written in a single write I/O to the PLOG and improving the performance of applications that were formerly constrained by writes to the PLOG.
- The new NWORK1BUFFERS ADARUN parameter can be used to increase the number of Work part 1 protection I/O buffers, allowing multiple Work part 1 blocks to be written in a single write I/O to the Work data set and improving the performance of applications that were formerly constrained by writes to Work part 1.
- An internal Software AG Vector Table (SAGVT) is introduced in this release for z/OS systems. This is an internal z/OS CSA structure anchored in a CVT-based field assigned by IBM to Software AG. The vector table is used internally by Adabas to identify the location of an internal SVC table Adabas now uses to store and maintain SVC numbers and their corresponding IDT addresses.

  In prior releases, this information was placed in the internal SSCT entry created every time an SVC is installed. To locate the IDT in former releases, Adabas was required to sequentially scan all of the SSCTs. With Adabas 8.2, when an SVC is installed, its IDT information is stored in the SVC table identified by the vector table. To locate the IDT associated with the SVC, Adabas need only scan the internal SVC table, thus eliminating the more time-intensive SSCT scan.

  For information on the Adabas SVC, read *Initializing the Adabas Communication Environment in z/OS* .
- Two new LGBLSET parameters have been added, UBTYPE and UBSTIME, that can be used to:
  - obtain a single user block element per CICS task (instead of a user block pool) and
  - set a time interval to clean up locked user blocks.

  These new parameters help eliminate abends that can occur when CICS transactions lock the user blocks but do not release them. For more information about the LGBLSET macro, read *Modifying Source Member Defaults (LGBLSET Macro) in Version 8* .

Nucleus 64-Bit Processing Support

Adabas 8.2 SP1 introduces support for 64-bit processing. If your z/OS operating system supports 64-bit virtual storage, you can now request that an Adabas nucleus use this virtual storage above the 2 gigabyte bar. A new ADARUN parameter, V64BIT, allows you to indicate whether your Adabas nucleus should use virtual storage above the 2 gigabyte bar. For more information, read *V64BIT : 64-Bit Virtual Storage Option*.

Adabas Buffer Description (ABD) Structure Changes

Updates have been made in Adabas 8.2 SP1 to support 64-bit and ALET-qualified buffers in Adabas calls. The structure of Adabas buffer descriptions (ABDs) has changed to accommodate this support. Two reserved fields (ABDXRSV4 and ABDXRSV5) have been replaced with new DSECT fields ABDXALET and ABDXADRG respectively.

- ABDXALET is now used to store the ALET for the buffer (if ABDXLOC=C’D’).

- ABDXADRG is now used to store the 64-bit indirect address pointer (if ABDXLOC=C’I’ or C’D’).

For complete information, read *Adabas Buffer Descriptions (ABDs)* .

Large Page Support

Adabas 8.2 SP1 introduces large page support for z/OS systems. If your z/OS operating system supports 64-bit virtual storage and large pages (1 MB pages of real storage), you can request that the Adabas nucleus use large pages. A new ADARUN parameter, LARGEPAGE, allows you to indicate whether your Adabas nucleus should use large pages or not. The default is not to use large pages. For more information, read *LARGEPAGE : Large Page Support Option*.

z/OS SVC Dump Enhancements

The z/OS SVC dump support in Adabas 8.2 SP1 has been enhanced to support the changes made by IBM to IEATDUMP in a recent z/OS release. z/OS users running non-authorized should be able to use the transaction dump IEATDUMP service in a similar way to authorized users with SVCDUMP.

For more information, read *Using the z/OS SVC Dump Facility*.

z/OS SVC Authorization Enhancements

Adabas 8.2 SP1 introduces support for the System Authorization Facility (SAF) in providing authorization for the Adabas 8 SVC in z/OS environments. SAF can now be used in some circumstances to ensure that the Adabas nucleus, as well as other MPM servers (such as Entire Net-Work and the Natural Global Buffer Pool) are authorized to prevent inappropriate use of critical ADASVC functions. There are still some circumstances where APF authorization is required instead. For complete information, read *Authorization Requirements*.

z/OS SVC Prelinked to SAF

Adabas 8.2 SP1 no longer requires that you link the Adabas SVC (ADASVC) with SAF (SVCSAF) during installation. In Adabas 8.2, the SVC is prelinked to SAF and the SVCSAF WAL library member is no longer supplied.

Reporting on Replication-Related Records in Work Part 1

Adabas 8.2 SP1 provides ADAWRK utility support for replication-related records written to Work part 1. Prior versions of the ADAWRK utility did not provide this support. You can now use the ADAWRK utility to report on replication-related records so you can determine how to handle the data when the database autostart fails and the database will not come up.

For complete information on replication support in the ADAWRK utility, read *ADAWRK Utility: Work Area Recovery Reports*

Printing Adabas Maintenance and SVC Status Information

A new utility, ADAZIN, is introduced in Adabas 8.2 SP1. ADAZIN can be used to print maintenance information about Adabas load modules and status information about the Adabas SVC on the system in which ADAZIN is run. Names of target load modules and SVC numbers can be specified to the utility to limit the range of the printed report.

Replication of Security Definitions

Adabas 8.2 SP1 allows you to replicate security definitions if you have Event Replicator for Adabas 3.2 or later installed. For more information, refer to your Event Replicator for Adabas documentation.

Address Converter Extent Handling Enhancement

With Adabas 8.2 SP1, contiguous address converter (AC) blocks are added to the last existing AC extent, if possible, rather than creating a new AC extent. However, if contiguous blocks are not available, a new AC extent is created.

This enhancement was made in order to reduce the number extents defined in the File Control Block (FCB). At this time, it affects only your Adabas nuclei. The Adabas utilities do not support this update at this time.

Adabas Delta Save Update: ADASAV MERGE Limitation Lifted

Adabas 8.2 SP1 includes internal enhancements that allow Adabas Delta Save to handle overlapping delta save numbers in ADASAV MERGE processing. In prior releases, the merge of two delta save data sets with overlapping delta save numbers would terminate abnormally with the message:

```
DSF065 Input Full and Delta Save tapes not fitting together
```

This enhancement allows data sets with DSIDs that share a delta save number to be merged. For example, the save tape with DSID `7/1-5/`*`timestamp1`* can now be merged successfully with a save tape with DSID `7/5-9/`*`timestamp2`*. The fact that overlapping delta save number 5 exists on both tapes and in both DSIDs can now be handled successfully during the merge.

For complete information about the ADASAV MERGE utility function, refer to your Adabas Delta Save documentation.

New and Changed Utility Functions

Adabas 8.2 introduces the following utility enhancements:

Utility Function	**New or Changed?**	**Enhancement Description**
ADACDC	Changed	Spanned records can now be fully decompressed and read from the PLOG. In support of this, two new parameters have been added: SPANREC and MAXLOGRECLEN. Each of these parameters is described in *Running the Utility* . The SPANREC parameter indicates that new spanned record CDCH and CDCN output headers should be used for all CDCOUT output if there is the possibility that the output decompressed spanned records will exceed the physical record length limitation. DSECTs for the CDCH and CDCN headers can be found in the Adabas source library.
ADACMP	Changed	The ADACMP utility has been updated to support the new date-time edit masks introduced in this version of Adabas. To support these date-time edit masks in field definitions, a new field option, DT, has been added to FNDEF field and group field definition statements. this field option allows you to specify a date-time edit mask for the field.
ADACMP COMPRESS	Changed	The following new parameters are supported in this release: ● A new TZ parameter allows you to specify the local time zone. As records are compressed and loaded into the file, the date-time data is converted from this time zone to UTC time. Date-time field data is always stored in UTC time in an Adabas file. ● A new DST parameter can now be specified to indicate that date-time data includes a daylight savings time indicator. If a time zone uses daylight savings time, you must be sure to store the daylight savings indicator with your date-time data or there will be no way to distinguish date-time values in the hour before the time is switched back to standard time. The following new field options are supported in this release in FNDEF field definitions: ● A new SY option allows you to indicate that the field is a system field and identify the type of system field being defined. ● A new CR option can be used for system fields (defined with the SY option) to indicate that a system field value should only be set when a record is inserted into a file and not when it is updated. Without the CR option, system field values are set for both record insertions and deletions. The CR option may not be specified for fields defined with the MU option. ● A new TZ option can now be used to indicated that a date-time field that should be presented in the user’s local time and stored in UTC time, allowing for differences in time zones.
ADACMP DECOMPRESS	Changed	The following new parameters are supported in this release: ● A new TZ parameter allows you to specify the local time zone. As records are decompressed and read from the file, the date-time data is converted from UTC to a corresponding time in this time zone. Date-time field data is always stored in UTC time in an Adabas file. ● In addition, a new DST parameter can now be specified to indicate that date-time data includes a daylight savings time indicator. If a time zone uses daylight savings time, you must be sure to retrieve the daylight savings indicator with your date-time data or there will be no way to distinguish date-time values in the hour before the time is switched back to standard time.

Utility Function	New or Changed?	Enhancement Description
ADACNV	Changed	The ADACNV utility has been updated so that you can now convert and revert Adabas databases between releases in *one step*, regardless of how many releases have occurred in between the two versions. The version of the earliest Adabas databases that can be converted using this utility is 6.1. Likewise, the earliest version of Adabas database to which you can revert a more current database is 6.1. In addition, during ADACNV processing, all changed RABNs are logged to the Adabas Delta Save area (in case Adabas Delta Save was enabled before the conversion). Finally, it is no longer necessary to make a full backup before and after a conversion; a backup of only the changed database blocks (an Adabas Delta Save ADASAV SAVE DELTA run) is sufficient.
ADADBS ADDCLOG	New	This new ADADBS utility function allows you to dynamically add command log (CLOG) data sets without terminating your current nucleus session. Running this utility function is invalid when Adabas is running with dual CLOGs.
ADADBS ADDPLOG	New	This new ADADBS utility function allows you to dynamically add protection log (PLOG) data sets without terminating your current nucleus session. Running this utility function is invalid when Adabas is running with dual PLOGs.
ADADBS CHANGE	Changed	The following ADADBS CHANGE updates have been made: ● You can now use this utility function to remove the UQ (unique descriptor) option from a descriptor field definition. A new option, NOUQ, has been added for the OPTION parameter of the ADADBS CHANGE function. ● You can specify an altered Adabas field definition for a field using a new FNDEF parameter. Changes to the time zone and the date-time edit mask are included.
ADADBS DELCLOG	New	This new ADADBS utility function allows you to dynamically delete command log (CLOG) data sets without terminating your current nucleus session. Running this utility function is invalid when Adabas is running with dual CLOGs.
ADADBS DELDE	New	This new ADADBS utility function allows you to logically delete a descriptor from a database file.
ADADBS DELFN	New	This new ADADBS utility function allows you to logically delete a field from a database file.
ADADBS DELPLOG	New	This new ADADBS utility function allows you to dynamically delete protection log (PLOG) data sets without terminating your current nucleus session. Running this utility function is invalid when Adabas is running with dual PLOGs.
ADADBS DSREUSE	Changed	In prior releases, Adabas locked the file associated with ADADBS DSREUSE processing. With this release, enhancements to Adabas have removed the necessity of performing this lock.
ADADBS DEVENTLOG	New	This new ADADBS utility function allows you to display the Adabas event log of a database. In a cluster environment, this function can display the Adabas event log of an individual database or of every database in the cluster.
ADADBS EXPFILE	New	This new utility function can be used to dynamically insert or remove files in expanded file chains.
ADADBS ISNREUSE	Changed	In prior releases, Adabas locked the file associated with ADADBS ISNREUSE processing. With this release, enhancements to Adabas have removed the necessity of performing this lock.

Utility Function	New or Changed?	Enhancement Description
ADADBS MODFCB	Changed	The following updates have been made to the ADADBS MODFCB function: ● A new FILEREADONLY parameter has been added that can be used to indicate whether an Adabas database file should be placed in readonly status (where it can be updated only by Adabas utilities) or in normal status (where it can be updated in any normal manner). **Notes:** 1. If the FILEREADONLY parameter is specified, only the FILE parameter (which is required) and the optional PASSWORD and TEST parameters can be specified in the same ADADBS MODFCB run. 2. The ADADBS DSREUSE, ISNREUSE, NEWFIELD, RELEASE DE, and RENAME utility functions do not need to lock the file for their use; these functions can perform their processing in parallel with active users. This means that you do not need to set a file in read-only status (FILEREADONLY=YES) if you will be performing these other ADADBS functions only. ● A new SYFMAXUV parameter is supported to specify the maximum number of occurrences for MU system field values that can be stored in a file. ● In addition in prior releases Adabas locked the file associated with ADADBS MODFCB processing. With this release, enhancements to Adabas have removed the necessity of performing this lock unless the LOBFILE or FILEREADONLY parameters are specified.
ADADBS NEWFIELD	Changed	The DT option is supported in FNDEF statements, allowing you to specify a date-time edit mask and a time zone for a new field. In addition, the system field options SY and CR are supported in FNDEF statements, allowing you to define system fields. In addition, in prior releases Adabas locked the file associated with ADADBS NEWFIELD processing. With this release, enhancements to Adabas have removed the necessity of performing this lock.
ADADBS ONLINVERT	Changed	The FNDEF parameter has been added to the ADADBS ONLINVERT utility function. Using this function you can specify a standard field definition for a new file descriptor. In addition, this new parameter allows you to define a date-time field (with edit mask and time zone) as a descriptor. For complete information, read *ONLINVERT: Start Online Invert Process* .
ADADBS OPERCOM	Changed	You can now use two new operator commands to dynamically control logging of the extended I/O list for both CLOGLAYOUT=5 and CLOGLAYOUT=8. 1. Use the new LOGVOLIO operator command to dynamically start logging of the extended I/O list. 2. Use the new NOLOGVOLIO operator command to dynamically stop logging the extended I/O list. For more information, read *Operator Commands* .

Utility Function	New or Changed?	Enhancement Description
ADADBS REACTLOG	New	This new ADADBS utility function allows you to reactivate command logging in an active nucleus where it had been disabled previously as a result of an I/O error. The cause of the I/O error needs to be corrected before running this utility function or command logging will simply fail again and will not be reactivated.
ADADBS RELEASE DESCRIPTOR	Changed	In prior releases, Adabas locked the file associated with ADADBS RELEASE DESCRIPTOR processing. With this release, enhancements to Adabas have removed the necessity of performing this lock.
ADADBS RENAME	Changed	In prior releases, Adabas locked the file associated with ADADBS RENAME processing. With this release, enhancements to Adabas have removed the necessity of performing this lock.
ADADBS UNDELDE	New	This new ADADBS utility function allows you to undelete a logically deleted descriptor from a database file.
ADADBS UNDELFN	New	This new ADADBS utility function allows you to logically undelete a field that had previously been logically deleted from a database file using the ADADBS DELFN function.
ADALOD LOAD	Changed	The following updates have been made to the ADALOD LOAD function: ● A new READONLY parameter has been added that allows you to change the read-only status of a file. ● A new SYFMAXUV parameter is supported to specify the maximum number of occurrences for MU system field values that can be stored in a file. ● The following new ADALOD LOAD parameters allow you to manage possible ADALOD LOAD replication errors. These parameters should only be used if you also have Event Replicator for Adabas installed. ○ The new RPLINITERROR parameter allows you to indicate whether ADALOD processing should continue if replication initialization errors occur. ○ The new RPLERRORDEACTFILE parameter allows you to indicate whether replication should be deactivated for the file if replication errors occur. ⚠ **Warning:** **If you elect to have ADALOD LOAD continue its processing after a replication error, you are also responsible for recovering your environment appropriately.**
ADAORD STORE	Changed	Four new replication-oriented parameters have been added to the ADAORD STORE function: RPLDSBI, RPLKEY, RPLTARGETID, and RPLUPDATEONLY. These parameters are added to support replication of data during a restore. In addition, a new READONLY parameter has been added that allows you to change the read-only status of the file. **Note:** These parameters can only be specified if you also have Adabas 8.2 SP2 or later installed.

Utility Function	New or Changed?	Enhancement Description
ADAREP	Changed	You can now produce an XML document containing the ADAREP database status report in XML format. This XML document is structured as defined by an XML schema definition (XSD) file provided with Adabas. You can transfer the XML document to any system (probably Windows or UNIX) using automated tools or user-written programs. The new OUTPUT parameter allows you to request this XML document. A new ACODE parameter allows you to specify the code page used for the XML output. In addition, parallel participant table (PPT) information is now included in the ADAREP report by default. PPT information can be very useful when an abnormal outage has occurred for the database and you need to determine the physical state of the database and its existing checkpoint details. You can use a new NOPPT parameter to indicate that the PPT information should *not* be included in the ADAREP report.
ADARES BACKOUT	Changed	The ADARES BACKOUT utility functions have been updated to support the replication of the protection log data set sent to the nucleus. Using the new RPLDATA parameter, you can now activate this replication. In addition, additional statistics that list the number of records backed out for each file in an ADARES BACKOUT run are listed when processing is complete.
ADARES CLCOPY	Changed	The ADARES CLCOPY function has been expanded. It now checks the PPT at the end of the CLCOPY job to determine whether any additional CLOG data sets need to be copied. If so, the ADARES utility invokes user exit 2 or user exit 12, as appropriate, to accommodate the number of data sets that need copying. For example, if NCLOG=8, once the initial CLCOPY job completes, the ADARES utility will issue a call to the nucleus to invoke user exit 2 or 12 for each uncopied CLOG data set it detects.
ADARES PLCOPY	Changed	The ADARES PLCOPY function has been expanded. It now checks the PPT at the end of the PLCOPY job to determine whether any additional PLOG data sets need to be copied. If so, the ADARES utility invokes user exit 2 or user exit 12, as appropriate, to accommodate the number of data sets that need copying. For example, if NPLOG=8, once the initial PLCOPY job completes, the ADARES utility will issue a call to the nucleus to invoke user exit 2 or 12 for each uncopied PLOG data set it detects.
ADARES REGENERATE	Changed	The ADARES REGENERATE utility functions have been updated to support the replication of the protection log data set sent to the nucleus. Using the new RPLDATA parameter, you can now activate this replication. In addition, additional statistics that list the number of records regenerated for each file in an ADARES REGENERATE run are listed when processing is complete.

Utility Function	New or Changed?	Enhancement Description
ADASAV RESTONL FILES	Changed	You can now restore a range of files using the ADASAV functions. As before, you can still restore individual or lists of files. To specify a range of files in the FILES or FMOVE parameters of the various ADASAV functions, use a dash (-) between the first and last file number in the range, in the format: *`fnfirst-fnlast`*. Six new replication-oriented parameters have been added to various ADASAV functions: RPLACTIVE, RPLDATA, RPLDSBI, RPLKEY, RPLTARGETID, and RPLUPDATEONLY. These parameters are added to support replication of data during a restore. **Note:** These parameters can only be specified if you also have Adabas 8.2 SP2 or later installed. In addition, a new READONLY parameter has been added to these functions that allows you to change the original read-only status of the file.
ADASAV RESTONL FMOVE		
ADASAV RESTONL GCB		
ADASAV RESTORE FILES		
ADASAV RESTORE FMOVE		
ADASAV RESTORE GCB		
ADASAV MERGE	Changed	This release of Adabas includes internal enhancements that allow Adabas Delta Save to handle overlapping delta save numbers in ADASAV MERGE processing. In prior releases, the merge of two delta save data sets with overlapping delta save numbers would terminate abnormally with the message: `DSF065 Input Full and Delta Save tapes not fitting together` This enhancement allows data sets with DSIDs that share a delta save number to be merged. For example, the save tape with DSID `7/1-5/`*`timestamp1`* can now be merged successfully with a save tape with DSID `7/5-9/`*`timestamp2`*. The fact that overlapping delta save number 5 exists on both tapes and in both DSIDs can now be handled successfully during the merge. For complete information about the ADASAV MERGE utility function, refer to your Adabas Delta Save documentation.

Utility Function	New or Changed?	Enhancement Description
ADASEL	Changed	Several new parameters and keywords have been added to the ADASEL utility in this release to enhance its flexibility and functionality: ● A new TEST parameter allows you to test the syntax of your ADASEL utility run without actually processing any PLOG records. ● A new FDTINPUT parameter allows you to specify an alternate FDT source to be used during ADASEL processing. ● A new FROM USER clause has been added to the SELECT parameter to allow you to select PLOG records by user or terminal ID, regardless of file number. This new clause is mutually exclusive with the FROM [FILE] and IN [FILE] clauses. In addition, a new NU ADASEL global parameter has been added that allows you to globally limit the number of user or terminal IDs processed by a SELECT parameter. The default is 20. ADASEL will allocate work storage for a run based on the setting of the NU global parameter. ● A new DISPLAY instruction option, ALL, allows you to display all fields of each selected record, including MU and PE group fields and their occurrence counts. ● Two new DISPLAY instruction options, BEFORE and BOTH, now allow you accurately indicate whether before, after, or before and after images of selected data should be output. BEFORE and BOTH are both optional and are mutually exclusive options; only of of them may be specified in a DISPLAY instruction, but neither of them is required. If BOTH is specified, both the before and after images are output; if BEFORE is specified, only the before images are output; if neither BEFORE nor BOTH is specified, only after images are output. ● A new DISPLAY instruction option, COUNT, now allows you to display the number of occurrences of an MU field or PE group. ● Spanned records can now be fully decompressed and read from the PLOG. In support of this, three new SET GLOBALS parameters have been added: SPANREC, MAXLOGRECLEN, and LWP. Each of these parameters is described in *SET GLOBALS Parameter*. In addition, a SPANREC parameter has been added to the ADASEL OUTPUT instruction and can be used in an ADASEL OUTPUT instruction to indicate that new spanned record SELH and SELC output headers should be used for all EXPA*n* output if there is the possibility that the output decompressed spanned records will exceed the physical record length limitation. DSECTs for the SELH and SELC headers can be found in the Adabas source library. For complete information, read *OUTPUT Instruction*. In addition, ADASEL error messages have been enhanced so they are more self-explanatory. In prior releases of Adabas, the IGNORE instruction (that can be specified in the THEN and ELSE clauses of an IF statement) would not actually ignore the before-image of an item. The before-image was output even though IGNORE was specified. In this release of Adabas, neither the before or after-images are output when IGNORE is specified.

Utility Function	New or Changed?	Enhancement Description
ADAWRK	Changed	A new REPLICATION parameter has been added to the ADAWRK utility that provides you with information on replication-related records in Work part 1. This parameter determines whether a new Replication report and new Replication Summary report are produced by the utility and the presence of additional replication statistics on the existing reports. In addition to the replication support, the ADAWRK utility now assigns a transaction sequence number to each transaction it reads from Work part 1, to aid in uniquely identifying and matching the transactions in the different ADAWRK reports
ADAZIN	New	A new utility, ADAZIN, is introduced in this release. ADAZIN can be used to print maintenance information about Adabas load modules and status information about the Adabas SVC on the system in which ADAZIN is run. Names of target load modules and SVC numbers can be specified to the utility to limit the range of the printed report.

New and Changed Operator Commands

Starting with Adabas 8.2, display-type commands from the operator console (such as DSTAT and DTH) will be executed directly if all Adabas threads or Adabas internal command queue elements are in use by other commands. This enhancement allows you to better determine the status of Adabas should a problem occur that impedes normal command processing. Note that this direct execution of display-type commands applies only to commands from the operator console, but not to commands issued by the ADADBS OPERCOM utility function, Adabas Online System, or Adabas Manager. Also, Adabas might be unable to respond to operator commands (issued from the operator console) in certain error situations, such as if Adabas is looping, in a hard wait, or when it is unable to obtain the CPU processing cycles it needs.

The following table lists new and changed operator commands in Adabas 8.2.

Command	New or Changed	Enhancement Description
ALOCKF	Changed	Support for this command, provided for non-cluster environments in prior versions, is now provided in cluster environments.
DRES	Changed	This command now displays information on the high-water marks for Work parts 1, 2, and 3, allowing you to monitor the use of the Work data set.
LICREFRESH	New	This new command allows you to refresh the license file currently used by the nucleus. It allows you to overwrite the current license with a new one or to force the system to reread and reload its existing license.
LOGVOLIO	New	This new command allows you to initiate logging of the extended I/O list to the CLOG data sets, for both CLOGLAYOUT=5 and CLOGLAYOUT=8. This command can be entered on the console and in an ADADBS OPERCOM utility function run.

Command	New or Changed	Enhancement Description
LOGWARN	New	This new command allows you to alter the setting of the LOGWARN ADARUN parameter for a given Adabas session.
NOLOGVOLIO	New	This new command allows you to stop logging of the extended I/O list to the CLOG data sets, for both CLOGLAYOUT=5 and CLOGLAYOUT=8. This command can be entered on the console and in an ADADBS OPERCOM utility function run.
NWCONNECT	New	This new command allows you to retry establishing the Entire Net-Work DBID target. During nucleus initialization, certain Entire Net-Work errors may have prevented the target from being established, resulting in message ADAM76. Other errors may prevent Entire Net-Work from defining the target when it is started after the nucleus is initialized. Support for the NWCONNECT command in Adabas Cluster Services and Adabas Parallel Services nuclei is provided by Adabas fix AN822079.
RALOCKF	Changed	Support for this command, provided for non-cluster environments in prior versions, is now provided in cluster environments.
RALOCKFA	Changed	Support for this command, provided for non-cluster environments in prior versions, is now provided in cluster environments.
SMFDETAIL	New	This new command allows you to alter the complete setting of the SMFDETAIL ADARUN parameter for the running Adabas session.
SMFDETAILADD	New	This new command allows you to add specific detail sections of Interval and Termination records to the list of detail records already specified by the SMFDETAIL ADARUN parameter for the running Adabas session.
SMFDETAILDEL	New	This new command allows you to remove specific detail sections of Interval and Termination records from the list of detail records specified by the SMFDETAIL ADARUN parameter for the running Adabas session.
SMFINTERVAL	New	This new command allows you to override the setting of the SMFINTERVAL ADARUN parameter for the running Adabas session.
SMFRECNO	New	This new command allows you to override the setting of the SMFRECNO ADARUN parameter for the running Adabas session.

Command	New or Changed	Enhancement Description
SMFSUBSYS	New	This new command allows you to override the setting of the SMFSUBSYS ADARUN parameter for the running Adabas session.

For more information about any Adabas operator command, read *Operator Commands*.

ADARUN Parameter Changes

This section describes the changes made to ADARUN parameters in this release. For more information about any ADARUN parameter, read *Adabas Initialization (ADARUN Statement)* .

The following table summarizes the ADARUN parameter changes in Adabas 8.2.

Parameter	New or Changed	Enhancement Description
CLUCACHEEXTRA	New	Use the new CLUCACHEEXTRA parameter to specify the additional vector size, in blocks, required to allow for the Adabas Caching Facility in the shared Adabas Cluster Services and Adabas Parallel Services cache.
CLUGROUPNAME	Changed	Support for the CLUGROUPNAME parameter with Adabas Parallel Services is provided in this release. In Adabas Parallel Services, use the CLUGROUPNAME parameter to specify the name of the messaging group. The CLUGROUPNAME parameter is already supported in Adabas Cluster Services; this is the first release in which it is supported for Adabas Parallel Services. The CLUGROUPNAME parameter need not be specified for Adabas Parallel Services except when Adabas Parallel Services is used to recover an Adabas Cluster Services database after a failure that resulted in a remnant DIB entry. In this case, the CLUGROUPNAME parameter for Adabas Parallel Services must be set to the same value as it was for the failed Adabas Cluster Services nuclei.

Parameter	New or Changed	Enhancement Description
FORCE	Changed	In this version a data integrity block (DIB) entry will only be removed once the ID Table initialization has been successful. Therefore, you must set IGNDIB and FORCE both to "YES" if either of the following occur: ● You receive a PARM ERROR 26 after parameter settings IGNDIB=NO and FORCE=YES were applied; ● You receive a PARM ERROR 23 after parameter settings IGNDIB=YES and FORCE=NO were applied.
IGNDIB	Changed	In this version a data integrity block (DIB) entry will only be removed once the ID Table initialization has been successful. Therefore, you must set IGNDIB and FORCE both to "YES" if either of the following occur: ● You receive a PARM ERROR 26 after parameter settings IGNDIB=NO and FORCE=YES were applied; ● You receive a PARM ERROR 23 after parameter settings IGNDIB=YES and FORCE=NO were applied.
INFOBUFFERSIZE	New	This new parameter allows you to specify the size, in bytes, of the information buffer pool that represents the new Adabas event log for a database. The default is "0" (no buffer pool/event log is allocated). Valid values other than "0" range from 1024 through 2147483647.
INTAUTO	New	This new parameter allows you to specify the time interval (in seconds) between autorestart progress messages. Valid values other than "0" range from 1 through 16777215.
LARGEPAGE	New	This new parameter allows you to indicate whether the Adabas nucleus should use large pages (1 MB pages of real storage above the 2 gigabyte bar). If your z/OS operating system supports 64-bit virtual storage and large pages, you can request that the Adabas nucleus use large pages. The default is NO (not to use large pages).

Parameter	New or Changed	Enhancement Description
LDEUQP	Changed	The following updates have been made to the LDEUQP ADARUN parameter in 8.2: ● Adabas makes an entry in the unique descriptor pool every time a record with a unique descriptor is deleted or altered such that an existing descriptor value is *deleted* from the index. Adabas Cluster Services and Adabas Parallel Services nuclei will now also make an entry when a record with a unique descriptor is *inserted* and thus the ADARUN LDEUQP parameter setting may require a larger value. Entries in the unique descriptor pool generally exist for the duration of a transaction. ● Slight performance improvements have been made in updating large objects (LOBs) by eliminating redundant serialization. All nuclei now reduce their use of the unique descriptor pool for LOB updates and customers using LOBs from Adabas 8.1 may be able to reduce the unique descriptor pool size (ADARUN parameter LDEUQP). Adabas Cluster Services and Adabas Parallel Services configurations no longer acquire external unique descriptor locks for LOB updates. ● Starting with Adabas 8.2 SP3, the minimum setting of the ADARUN LDEUQP parameter is increased from zero to 5000 bytes.
LFIOP	Changed	The maximum setting for LFIOP is now 18,446,744,073,709,551,615 -- although this is really limited by the available amount of virtual and real memory. In addition, in environments supporting 64-bit virtual storage, the flush I/O pool is now allocated above the 2 gigabyte bar when the V64BIT ADARUN parameter is set to "YES".
LOGVOlio	New	This new parameter allows you to indicate whether the extended I/O list should be written to the command log for CLOGLAYOUT=5 and CLOGLAYOUT=8.
LOGWARN	New	This new parameter allows you to specify whether CLOG and PLOG status checking should occur, with resulting alert messages, and, if so, the frequency of the status checking (in seconds). Valid values range from zero (0 - no checking) to 2147483647 seconds. When a value other than zero (0) is specified for LOGWARN, a valid user exit 2 or user exit 12 must also be specified.

Parameter	New or Changed	Enhancement Description
LRDP	Changed	With this release of Adabas and the upcoming 8.2 releases of Adabas Cluster Services and Adabas Parallel Services, the default value for LRDP has changed: ● For Adabas Parallel Services 8.2 or later, the default is 0 (zero). ● For Adabas Cluster Services 8.2 or later, the default is either the value of the ADARUN LFIOP parameter or 10 megabytes, whichever is smaller.
MLWTO	New	Support for the MLWTO parameter is provided in this release of Adabas, in support of the following Adabas product extensions: ● Adabas Cluster Services ● Adabas Parallel Services ● Event Replicator for Adabas Use this parameter to specify whether multi-line messages that are normally written to the operator console as a series of single-line write-to-operator (WTO) messages should instead appear as multi-line WTO messages. When requested, this parameter will affect the appearance of replication-related and cluster-related messages (messages with prefixes ADAF*, ADAX*, PLI*, and PLX*).
NPLOGBUFFERS	New	This new parameter allows you to specify the number of PLOG protection I/O buffers allocated in the Adabas nucleus session. PLOG protection I/O buffers are only allocated when ADARUN parameter LFIOP is set to a nonzero value. The default is 1.
NT	Changed	The minimum value allowed for the NT parameter was changed from "3" to "4" in this version of Adabas.
NWORK1BUFFERS	New	This new parameter allows you to specify the number of Work part 1 protection I/O buffers allocated in the Adabas nucleus session. Work part 1 protection I/O buffers are only allocated when ADARUN parameter LFIOP is set to a nonzero value. The default is 1.

Parameter	New or Changed	Enhancement Description
PAMREQS	New	This new parameter is available in BS2000 environments only. You can use it to set the maximum number of PAM blocks that can be transferred in one PAM macro request. Valid values range from 1 through 1024, with a default of 80. This parameter replaces an optional zap for Adabas in BS2000 environments.
PLOGSEGSDS	Dropped	This parameter has been dropped from this release. It is replaced by the ADARUN SRLOG parameter.
PREFSTDD	New	This new parameter allows you to specify the job control statement label (DD name) in the job stream that defines the location to which multifetch statistics should be written when a CL command is issued.
SMF	New	This new parameter allows you to enable or disable Adabas SMF recording for the nucleus.
SMF89	New	This new parameter controls whether the Adabas nucleus registers with z/OS for type 89 SMF records.
SMFDETAIL	New	This new parameter allows you to specify the specific Interval and Termination detail record sections you want to include in the Adabas SMF records.
SMFINTERVAL	New	This new parameter allows you to specify whether interval SMF records should be generated and how often. A specific interval can be specified or a source from which the interval can be derived.
SMFRECNO	New	This new parameter allows you to specify the Adabas SMF record number for user-defined records.
SMFSUBSYS	New	This new parameter allows you to associated the SMF records with an IBM or user-defined system.
SRLOG	New	This new parameter allows you to indicate how spanned records should be logged to the protection logs. Entire records (ALL), modified segments of spanned records (UPD), or entire records up to and including the modified segments (PART) can be logged. The default is UPD. The SRLOG parameter replaces the PLOGSEGSDS ADARUN parameter; the PLOGSEGSDS parameter has been dropped in this release.

Parameter	New or Changed	Enhancement Description
SWITCHNR	New	This new parameter allows you to set the job switch number that is set when a nucleus or utility exits with a non-zero response code. Valid values range from 1 through 31, with a default of 10. This parameter replaces an optional zap for Adabas in BS2000 environments.
UEXSMF	New	This new parameter allows you to specify the module name of the SMF user exit that should be invoked during Adabas SMF recording.
V64BIT	New	This new parameter allows you to indicate whether the Adabas nucleus should use virtual storage above the 2 gigabyte bar. If your z/OS operating system supports 64-bit virtual storage, you can request that the Adabas nucleus use it also. The default is NO (not to use 64-bit virtual storage).

Direct Call Command Enhancements

The following direct call command enhancements were made in Adabas 8.2:

- Statistics for the internally used V* and Y* commands are now included on the Adabas Online System Command Usage screen as well as in Adabas shutdown statistics. These commands are used internally by Adabas and Adabas add-on products and should not be used in direct calls in your applications. Should you use them, errors will result.

- A new daylight savings time indicator, D, can be specified in format and search buffers to indicate that a daylight savings adjustment is required for a date-time field with the TZ option specified. For complete information, read *Daylight Savings Indicator (D)*.

- Date-time edit masks are supported in the format and search buffers of direct calls. You can add date-time edit mask to fields with formats of B, F, P, or U in a loaded file. For more information, read *Date-Time Edit Mask Processing in Format and Search Buffers*.

- The following table lists new and changed direct call commands in Adabas 8.2.

Command	New or Changed	Enhancement Description
A1	Changed	A new Command Option 2 value, L, can now be specified in the Adabas (ACBX or ACB) control block to indicate that the position within a large object (LOB) field be tracked in the ISN lower limit field. This option is useful when multiple A1 commands are being issued in a series for a LOB field. This option can only be used if a LOB field is specified in the format buffer using LOB segment notation with asterisk notation in the *bytenum* specification. For more information, read *Processing LOB Segments*
BT	Changed	A new Command Option 3 value, H, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status indefinitely. For more information about shared hold status, read *Shared Hold Status*.
CL	Changed	The following updates have been made: ● In ACBX calls only, user CPU consumption values are now returned in eight bytes (instead of four bytes) and are stored in the ACBXISQG field. The unit of measurement is 1/4096 of a microsecond (instead of 1.048576 seconds). ● You can now request multifetch statistics when a CL command is issued if the PREFSTDD ADARUN parameter is specified in the startup JCL and if the JCL includes a valid job control statement with the label name specified by the PREFSTDD ADARUN parameter.
ET	Changed	A new Command Option 3 value, H, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status indefinitely. For more information about shared hold status, read *Shared Hold Status*.
HI	Changed	A new Command Option 3 value, S, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status until the end of the transaction. For more information about shared hold status, read *Shared Hold Status*.

Command	New or Changed	Enhancement Description
L1	Changed	A new Command Option 2 value, L, can now be specified in the Adabas (ACBX or ACB) control block to indicate that the position within a large object (LOB) field be tracked in the ISN lower limit field. This option is useful when multiple L1 commands are being issued in a series for a LOB field. This option can only be used if a LOB field is specified in the format buffer using LOB segment notation with asterisk notation in the *bytenum* specification. For more information, read *Processing LOB Segments*
L4	Changed	Several new Command Option 3 values, C, Q and S, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status for varying time periods, depending on the option. Option C puts the record in shared hold status for the duration of the read operation; option Q puts the record in shared hold status until the next record in the read sequence is read or the read sequence or transaction is terminated; option S puts the record in shared hold status until the end of the transaction. For more information about shared hold status, read *Shared Hold Status*. A new Command Option 2 value, L, can now be specified in the Adabas (ACBX or ACB) control block to indicate that the position within a large object (LOB) field be tracked in the ISN lower limit field. This option is useful when multiple L1 commands are being issued in a series for a LOB field. This option can only be used if a LOB field is specified in the format buffer using LOB segment notation with asterisk notation in the *bytenum* specification. For more information, read *Processing LOB Segments*
L5	Changed	Several new Command Option 3 values, C, Q and S, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status for varying time periods, depending on the option. Option C puts the record in shared hold status for the duration of the read operation; option Q puts the record in shared hold status until the next record in the read sequence is read or the read sequence or transaction is terminated; option S puts the record in shared hold status until the end of the transaction. For more information about shared hold status, read *Shared Hold Status*.

Command	New or Changed	Enhancement Description
L6	Changed	Several new Command Option 3 values, C, Q and S, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status for varying time periods, depending on the option. Option C puts the record in shared hold status for the duration of the read operation; option Q puts the record in shared hold status until the next record in the read sequence is read or the read sequence or transaction is terminated; option S puts the record in shared hold status until the end of the transaction. For more information about shared hold status, read *Shared Hold Status*.
LF	Changed	A new Command Option 2 setting of "X" is supported for this command. This setting produces a record buffer containing extended field information, including collation descriptor, subfield, superfield, subdescriptor, superdescriptor, hyperdescriptor, and phonetic descriptor information with extended date and time formats.
OP	Changed	A new TZ (time zone) record buffer option is available for the OP command. You can use this option to set the time zone of an individual user session. If no time zone is specified for a user session, response 55 (ADARSP055) subcode 30 is issued when a field with the TZ option is accessed or updated. For more information about using the TZ record buffer option in an OP command, read about the OP command *Record Buffer*.
RI	Changed	Adabas on mainframes no longer releases records unconditionally. If your application issues an RI command for a record that has been updated in the current transaction, Adabas will now return response code 113 (ADARSP113) or, if ISN=0 was specified, response code 2 (ADARSP002) is returned. This change in behavior may affect how your application programs are coded. For more information, read *RI Commands: Change Possibly Affecting Your Application Programs*. In addition a new Command Option 3 value, S, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status until the end of the transaction. For more information about shared hold status, read *Shared Hold Status*.

Command	New or Changed	Enhancement Description
S4	Changed	Several new Command Option 3 values, C, Q and S, can now be specified in the ACBX control block to indicate that a record should be kept in shared hold status for varying time periods, depending on the option. Option C puts the record in shared hold status for the duration of the read operation; option Q puts the record in shared hold status until the next record in the read sequence is read or the read sequence or transaction is terminated; option S puts the record in shared hold status until the end of the transaction. For more information about shared hold status, read *Shared Hold Status*.

For more information about direct call commands, read *Commands*.

New LGBLSET Macro Parameters

The following new LGBLSET parameters were added in Adabas 8.2 SP2.

- LUIDX can be used to indicate whether or not a CICS link routine user ID user exit should be invoked.
- LUIXNAM can be used to specify the name of the CICS link routine user ID user exit program that should be invoked.
- UBTYPE can be used to obtain a single user block element per CICS task (instead of a user block pool). This new parameter helps eliminate abends that can occur when CICS transactions lock the user blocks but do not release them.
- UBSTIME can be used to set a time interval to clean up locked user blocks. This new parameter helps eliminate abends that can occur when CICS transactions lock the user blocks but do not release them.

For more information about the LGBLSET macro, read *Modifying Source Member Defaults (LGBLSET Macro) in Version 8* .