

Migrating From Previous Adabas Versions

When planning your migration from prior versions of Adabas to Adabas 8, please consider the following information. For information on Adabas add-on support, read *Add-On Product Support for Adabas 8*.

1. A new direct call interface has been added in Adabas 8 that is fully compatible with the existing Adabas ACB-based direct call interface. The new direct call interface is based on the new Adabas 8 ACBX and ABD structures.

Existing application programs that use the ACB-based direct call interface can continue to run in the same way, without change. In addition, you can decide whether you want to use the ACBX-based or ACB-based direct call interface in your application programs, on a call-by-call basis. The same program can use both interfaces.

Some of the new features of Adabas 8 require that your application program use the ACBX-based direct call interface. For example, if your application program makes use of the long buffers (larger than 32K) or segmented buffers (multiple format and record buffers) available with Adabas 8, you must use the ACBX-based direct call interface. For more information on these, read *Command Changes*.

2. Existing files must either be reloaded or reorganized if you want to take advantage of the increased limit for MU and PE fields provided with Adabas 8. You may also need to:
 - increase the work pool (ADARUN LWP parameter)
 - increase the protection area on the Work data set (ADARUN LP parameter)
 - increase the internal format pool (ADARUN LFP parameter).
3. If a file has been established with the extended MU or PE limits available in Adabas 8 (for example, using the ADADBS MUPEX function described in *Utility Changes*), and an application program reads the occurrence count of an MU field or PE group using an `xxC` element in the format buffer, verify that the program reads the occurrence count into a record buffer field with two or more bytes (for example, `FB='MUC,2,B.'`) If the program reads the occurrence count into a one-byte field (for example, `FB='MUC.'` or `FB='MUC,1,B.'`), it must be adjusted so that it can deal with two-byte occurrence count values. Adabas returns response code 55, subcode 9 if you only provide a one-byte field in the record buffer for the occurrence count of an MU field or PE group in a file with extended MU/PE limits.
4. If your hyperdescriptors make use of extended MU or PE fields, you will need to reassemble the corresponding hyperexits; with adjustments for the extended MU or PE fields, parameter list, and input parameters. For more information, read *Hyperexit Changes*.
5. Large object fields -- that is, fields defined with the new LB option -- are generally stored in the LOB file associated with the file containing the LB fields (base file). Only small LB field values (up to 253 bytes) are stored directly in the base file. In the LOB file, LB field values are partitioned into segments and stored in one or more segment records, each of which (except the last) occupies a whole Data Storage block.

If you plan to use large object fields, reevaluate your Adabas nucleus (ADARUN) parameters according to the following guidelines:

- For insert, update, and delete operations involving LB field values, Adabas takes all LOB file segment records involved in hold. The hold operations are counted against the ADARUN NH and NISNHQ parameters. These parameters should therefore be increased depending on the expected maximum number of LOB file segment records that may be updated in parallel transactions.
- Furthermore, the before and after images of LB field values being inserted, updated, or deleted are written to the protection area on the Work data set. The LP parameter should be increased by the number of Work blocks equivalent to twice the estimated size of LB field values that may be updated in parallel transactions.
- The LOB file is defined with a unique descriptor (UQ option) that identifies its individual segment records. When updating or deleting LB field values, Adabas needs about 30 bytes of unique-descriptor-pool space per LOB segment record. The LDEUQ parameter should be increased by 30 times the maximum number segment records per large object times the number of LB field values that may be deleted or changed in parallel transactions.
- The NAB parameter should be increased by the size of the largest LOB value that may be read from or written to the database, times the estimated maximum number of parallel Adabas calls that may read or write LOB values from or to the database.
- The LU parameter should be increased by the size of the largest LOB value that may be read from or written to the database. Note that the LU parameter does not describe the size of a memory area allocated by Adabas but rather the maximum length of an area of the attached buffers (defined by the NAB parameter) that can be used by a single Adabas call.

For complete information about ADARUN parameters, read *Adabas Initialization (ADARUN Statement)*.

6. If you intend to use either large object (LB) fields or spanned records, ensure that the ADARUN LWP parameter is at least 50,000 bytes times the number of threads (NT parameter). For complete information about the ADARUN LWP parameter, read *LWP : Length of Adabas Work Pool*.
7. Some restrictions exist in the use of utilities under Adabas 8. Be sure to read about the utility enhancements and restrictions in *Utility Changes* and *Limitations and Restrictions* for a description of how each Adabas utility has been updated for Adabas 8 and what the limitations and restrictions are for their use.