**Treehouse SOFTWARE**

# ADAREORG V3.03a
## Release Notes

**Revised: 5ᵗʰ April 2012**

## 1. Introduction

ADAREORG V3.03a has been introduced to provide explicit support for ADABAS 8.1x and above. This includes support for spanned records and MUPEX features, it does not however provide support for large object features of ADABAS namely BLOBS.. It consolidates all previous fixes and includes improved support for G and W type fields

**Please note this version of ADAREORG will only work with ADABAS v8.1x or above.**

**Features Supported:**

1.  Ignore counts after PE/MU entries in DDWAN cards.
2.  Add support for lower case field names.
3.  Implement support for MUPEX parameter in ADAREORG parms
4.  Fix various problems related to second input file
5.  Move processing buffers above the line to remove memory shortages
6.  Correction to LRECL processing to prevent truncation of records longer than 32kb
7.  Increase ADADEC buffer to prevent S0C4
8.  Report out of region as correct error, replacing the ubiquitous S0C3.
9.  Correction to increase decompressed supported LRECL to 1mb when spanned records are not used. Previous limit was 30kb
10. Correction to support decompressed records in memory GT 32kb
11. Corrections to CPUID processing and allow multiple product keys in the product zap
12. Corrections to MUPEX support
13. Tidy up display of licence code display
14. Support for ADABAS v8.2 basic features
15. MUPEX support.
16. Spanned record support.
17. Correction to product code checking routines to identify CPU's in 31 bit mode, a new licence key will be required for this version.
18. Method of recognition of LA fields has changed with ADABAS V8.
19. Implementation of the translation of 1 byte alpha containing blank (X'40') to zero (X'F0') when being converted to other field types.
20. CPUIDs are now displayed in the joblog to help diagnose codeword problems.
21. P type fields which are used as the source of information during P to A conversions are no longer corrupted.
22. U type fields which are used as the source of information during U to A conversions are no longer corrupted.
23. The ADAVER parameter is now an optional parameter with a default value of [8]. This means ADAREORG will interpret all input formats as that of ADABAS v8. To support earlier versions of ADABAS the DBA must use a previous version of ADAREORG.
24. Floating point fields (type G) are now treated as real numeric fields, which may be converted to other field types, and vice versa. This means they are no longer simply treated as binary fields. See the V300 manual for a more extensive explanation.

25. Conversion of numeric field types to A type has changed. "A type" is assumed to be legible and when it is on the receiving end of conversion from a numeric field, the content of the A type fields will have a leading + or – sign, as appropriate. When used as the input for a numeric conversion, leading + or – signs are checked for, and taken into account.
26. Enabled expiry date warning system.

**NOTE:** If DDWAN0n (for n=1,2) is PRESENT then no attempt is made to extract FDT information from DDADA0n (matching n=1,2). When DDWAN01(2) is ABSENT, and USERISN is specified for DDWAN03, then USERISN is assumed to have been specified for DDADA01(2), and ISNs are taken from this file (DDADA01 in preference to DDADA02). This ISN may later be overwritten by an internally generated ISN, if the ISN parameter is specified in the PARM file.

DDWAN01 is present in the JCL and set to DUMMY then this will lead to problems in ADAREORG. If DDWAN01 is not needed it should be commented out. There are impacts with USERISN in this regard, ie it is ignored if DDWAN01 is set to DUMMY.

This version implements support for the ADACMP COMPRESS MUPEX parameter. This means that when specified explicitly in the DDWAN03 parameter cards, the output file will have **2 byte index values** i.e. internally it will be a MUPEX file. That in turn means that ADACMP will convert this to a MUPEX compressed file, assuming that one also specifies the MUPEX parameter to ADACMP.

As a consequence of this MUPEX support ADAREORG can also be used to convert non-MUPEX files into MUPEX files.

**Installation Notes for e-MAIL or CDROM Supplied Software**

This release contains a single compressed directory zip file as follows:

**ARv303a-Release.zip**

The contents of the compressed file will be similar to the following:

➢ **ARv303ain.cmp**  - Source Library Members in compressed EBCDIC XMIT format

➢ **ARv303a-Release-Notes.pdf** – Release notes in PDF format.

➢ **ARv303a-Users-Guide.pdf** – the manual in PDF format.

➢ **ARv303aL.cmp**  - Load Library in compressed EBCDIC XMIT format

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Installation Procedure Overview**

1.     Save the email attachment and/or PC files to disk.
2.     Unzip the supplied release libraries to a directory on a PC. There will be four to six files in total, but z/OS users will only receive the appropriate files:
    - readme.txt; a text file containing last minute information on the release;
    - ARv302-Users-Guide.PDF – The ADAREORG Users Guide;
    - ARv302h-Release-Notes.PDF – The ADAREORG Release Notes;
    - ARv302in – JCL, samples and Macro Library – EBCDIC compressed z/OS only;
    - ARv302inf – JCL, samples/Macro Library – ASCII zip MSP only;
    - ARv302L – Load library – EBCDIC compressed z/OS only
    - ARv302Lf.dat – Load library – EBCDIC format, build job for Fujitsu MSP only.
3.     Load the two mainframe files (ARv302hIN, ARv302hL) to the mainframe-using binary FTP or whatever file transfer mechanism is used onsite – binary mode only, without any translation as the files are already in EBCDIC format.
4.     Unpack the datasets using the information in the following section, Note this procedure will be different for MSP/EX and OS390 [z/OS] users.


**Procedure to Decompress the ADAREORG Install JCL and Load Libraries**

When decompressing the release zip files do not change file extensions on the PC platform, this can lead to problems with CR/LF on binary and ASCII files.

Please refer to the Users Guide for details of the **Installation Procedure**

At this point both the JCL and Load libraries will have been populated and the members are ready for tailoring and testing. There is no longer a requirement to run the build step to create the load modules under z/OS systems [**in fact the installer must not do this..**].

Once the above Source and Load Libraries are in decompressed you should be in a position to run ADAREORG V3.02.

xxxxxx.ARv302.INSTALL install library contains JCL giving examples of user-exits and ADAREORG execution jobs.

Please ensure you modify these to conform to your local standards making appropriate changes to the dataset names in these members to match the ones at your site, especially the JOBLIB/STEPLIB card to point to the V302 Load Library.

**3. Special Notes for Spanned Records and MUPEX processing**

With the advent of spanned records and MUPEX support in ADAREORG has meant some new parameters and some additional steps in processing files with spanned records turned on. One of the key issues is that of record size, the physical maximum BLKSIZE under z/OS is 32kb on disk – with LBI it can be larger on cartridge (64kb), so to cater for records that

maybe gt 32kb in a spanned record the parameter **HEADER=YES** must be used – otherwise ADAREORG may abend with an S0C4.

*The REGION size for ADAREORG jobs should be increased as memory requirements have increased due to the much larger buffers needed to support spanned records We suggest REGION=0M.*

**HEADER=YES** applies to decompressed records and must be supplied to both ADAREORG and ADACMP. This allows ADACMP to understand that it is receiving the spanned records as input from ADAREORG [and ADAREORG understands that it is extracting spanned records from the compressed input file]. See the Users Guide for a more comprehensive discussion on this issue.

When unloading the file to be processed with spanned records it is important to use the ADAULD parameter **SORTSEQ=ISN**, this will put all ISN's in an order – including secondary ISN's - that ADAREORG can process them properly.

## 4. Apply Product Protection Code

The codeword format has changed for this release and above, all customers will need a new codeword to run the software.

ADAREORG requires our Product Authorisation Code, a 22-byte (or more) codeword will need to be supplied so that ADAREORG will run on your system. The code is supplied to ADAREORG as the FIRST parameter in an ADAREORG Parameter dataset (or instream), eg:

CODE=**BIEJTJLJJJJLJHMKIHKHFT**

The code may be permanently zapped into the ADAREORG load module, this zap must be created by CCA and takes the place of the CODE parameter.

Note: A previously zapped codeword will prevent the new zap from applying, either apply the zap to a new/fresh copy of the load module OR comment out the VER's, ensuing no mistakes are made to the zap contents.

**PLEASE CONTACT US FOR FURTHER INFORMATION**