Adabas for Linux, UNIX and Windows

Universal Encoding Support (UES)

Version 7.0.1

October 2022

ADABAS & NATURAL

This document applies to Adabas for Linux, UNIX and Windows Version 7.0.1 and all subsequent releases.

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

Document ID: ADAOS-UES-701-20220622

Table of Contents

Universal Encoding Support

This document contains information about Universal Encoding Support on Adabas.

The following topic is covered:

- *Universal Encoding Support (UES)*

1 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at **https://documentation.softwareag.com**.

In addition, you can also access the cloud product documentation via **https://www.softwareag.cloud**. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at **https://knowledge.softwareag.com**.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at **https://tech-community.softwareag.com**. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at **https://github.com/softwareag** and **https://hub.docker.com/publishers/softwareag** and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at **https://empower.softwareag.com**. Many services on this portal require that you have an account. If you do not yet have one, you can request it at **https://empower.softwareag.com/register**. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

2 Universal Encoding Support (UES)

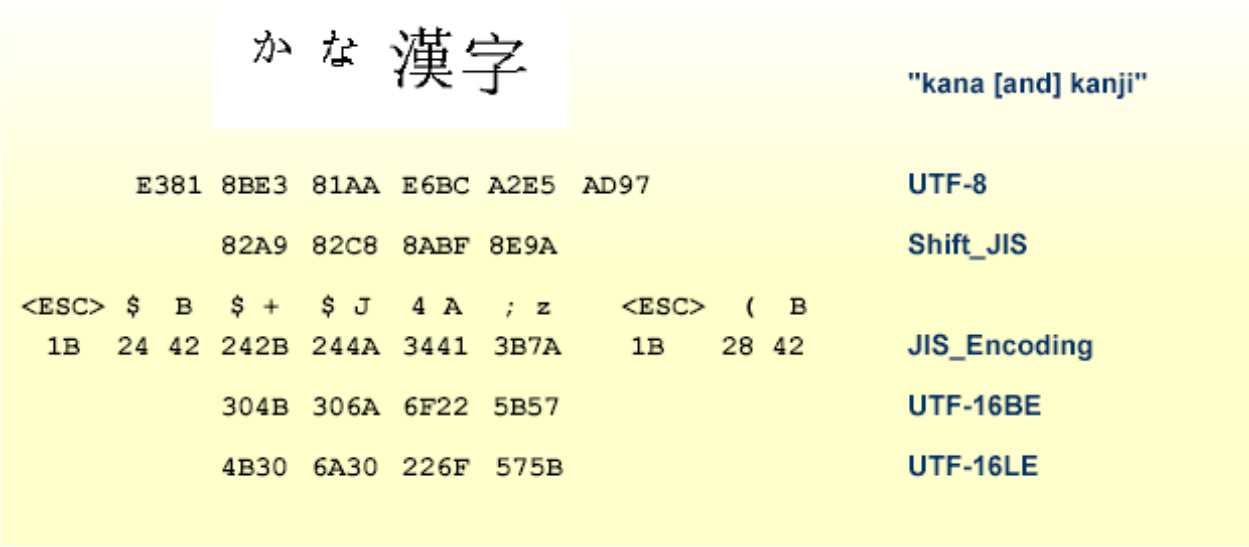
Support for Non-Latin Character Sets

Adabas can now use character sets other than those based on the Latin alphabet and can retrieve data that use these character sets in the correct collating sequence.

In most cases, an Asian text character cannot be encoded using a single byte. For example, Japanese with more than 10,000 characters in its set is encoded using two or more bytes per character. Because of the encoding required, these are called double-byte character sets (DBCS) or multiple-byte character sets (MBCS) as opposed to the single-byte character sets (SBCS) characteristic of most Western languages.

There has been a standardisation of these character sets by the Unicode consortium in the Unicode standard, - please refer to the Unicode homepage at **http://www.unicode.org/** for further information. An example of a DBCS is UCS-2, which contains all characters of Unicode Version 1.1, and an example of an MBCS is UTF-8, which represents all characters of the current Unicode version in 1 to 4 characters.

In the figure below, the Japanese kana (first two) and kanji (second two) characters are shown in a variety of encodings:



Notes:

1. Some character sets are platform-dependent, for example UTF-16. Software AG therefore recommends that you use UTF-16BE (high-order first) or UTF-16LE (low-order first) instead of UTF-16, since it is not possible to ensure which variant will be used if a platform dependent encoding is specified.

2. The length of a field will vary with the encoding used. This means there may be cases in which the specified field length for one encoding will not be sufficient with a different encoding.

Field Format W for Wide-Character Encoding

Earlier versions of Adabas stored DBCS-encoded data in alphanumeric fields, but Adabas had no knowledge of the character sets. Applications had to know which character sets were used, and the applications themselves were responsible for conversions if they required the data in another character set. This, of course, is still possible with Adabas Version 5, but it now also accommodates DBCS and MBCS much more suitably by the introduction of the wide-character or Unicode (W) field format.

The new field format W has been created to handle double- and multiple-byte characters. The size of the W-format fields in bytes, like alphanumeric fields, is either a standard length defined in the FDT or a variable length. A W-format field can have the same field options as an alphanumeric field, except for the NV option.

A W-format field can be defined as a parent of a super-/subdescriptor; it cannot be defined as a parent of a hyperdescriptor or a phonetic descriptor.

If a superdescriptor contains wide-character (W) fields, the format of the superdescriptor is A.

All data for wide-character formatted fields is stored internally in UTF-8, but the Adabas user can specify the external encoding for a user session in the record buffer of the OP command or with the compression/decompression utilities ADACMP/ADADCU.

Notes:

1. Adabas supports Unicode on the basis of ICU (International Components of Unicode) on Unix and Windows, please refer to the ICU homepage at **https://www.ibm.com/software/globalization/icu** for further information about ICU. Previously, only one ICU version could be supported in Adabas. For a long time, Adabas supported ICU Versions 3.2 on UNIX and Windows (Version 3.6 on OpenVMS) although newer versions are available, because Adabas stores collation keys persistently for collation descriptors and the collation keys may be different in different versions. With Adabas version 6.5, the support of more than one ICU version was introduced; With Adabas Version 6.5, additionally ICU version 5.4 is supported. For existing collation descriptors, Adabas continues using the old ICU version until the descriptor is recreated with ADAINV REINVERT; new collation descriptors are always created with the new ICU version. For future versions, it is planned to always support two ICU versions: If a new Adabas version supports a new ICU version, it no longer supports the oldest ICU version supported before. The consequence of this is, that you should upgrade the ICU version of collation descriptors using the old ICU version. You can do so by reinverting the collation descriptors before upgrading to the new Adabas version. ADAREP FULL FDT can be used to find out the actual ICU Version of a descriptor.

2. There are some differences in the syntax or semantics of the collation specifications for ICU version 3.2 and ICU version 5.4. For example, with ICU version 3.2 the locale "fr" implied the FRENCH option, while with ICU version 5.4, the FRENCH option must be specified explicitly. Please check, if such changes are relevant for some of your collation descriptors. If yes, the affected collation descriptors must be recreated with the new specification for ICU version 5.4, which is equivalent to the old specification used for ICU version 3.2.

Collation Descriptor to Support Universal Encoding

If you have text fields, you usually do not want to order them according to their byte sequence; usually the required collation sequence is language dependent, for example, in Spanish “LL” is considered to be one character, unlike in other languages. Also, for the same language, many different collations are possible. For example, there may be different collations for phonebooks and for book indices. Uppercase/lowercase characters, hyphens in words (e.g., “coop” versus “co-op”), diacritic marks (e.g. umlauts, accent marks in such words as “résumé” versus “resume”) may affect sequencing, or they may be ignored.

The sequencing rules for each collation are implemented in routines that generate a collation key for each text field value; Adabas uses ICU to do this.

A collation descriptor is a descriptor that is based on an ICU collating key for a Unicode field, where the ICU collating key is a binary string produced from the original character string by applying a Unicode Collation Algorithm and language-specific rules. When you perform a binary comparison between the collating keys produced this way for character strings, you perform a comparison between the strings that is appropriate to your locale.

Multiple Platform Support

Universal encoding support (UES) makes it possible for Adabas to process text data provided in any encoding supported by ICU, and to return text data in this encoding. With earlier versions of Adabas, it was only possible to convert data for Adabas buffers between different machine architectures (ASCII, EBCDIC) with one fixed translation table - it was not possible to convert between more than one ASCII and one EBCDIC derived character set.