Universal Encoding Support (UES)

Note:
UES support requires that you use a version 7 or above Adabas SVC or router.

The Universal Encoding Support (UES) is a database option that enables Adabas to

- perform data conversions;
- handle wide-character encoding;
- set the basis for internationalization tasks such as collation sequences.

Data conversion needs arise when communicating with different systems, i.e., conversion between different code pages for alphanumeric data or conversion of numerical data due to different machine architectures (see also section Multiple Platform Support).

Wide-character encoding is used in Asian language environments. Due to the need for a large number of different characters, non-single-byte character sets have been defined. In addition, Unicode, a Universal Character Set, is more frequently used (see also section Wide-Character Encodings).

A frequently listed internationalization task is searching and sorting data in a language specific order rather than binary order as defined by the encoding (see also section Collation Descriptor Exits in *User Exits*.

This chapter covers the following topics:

- Wide-Character Encodings
- Wide-Character Data Support

Wide-Character Encodings

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.

Previous versions of Adabas have stored DBCS-encoded data in alphanumeric fields. Problems with this solution include the following:

- the default blank of alphanumeric fields may be different from the blank required for double- or multiple-byte character fields;
- field truncations caused by length overwrites can result in changed or invalid characters because the string is cut off at a byte boundary rather than at a character boundary.
- client/server applications are difficult to implement when client and server use different encodings for their double- or multiple-byte character sets.

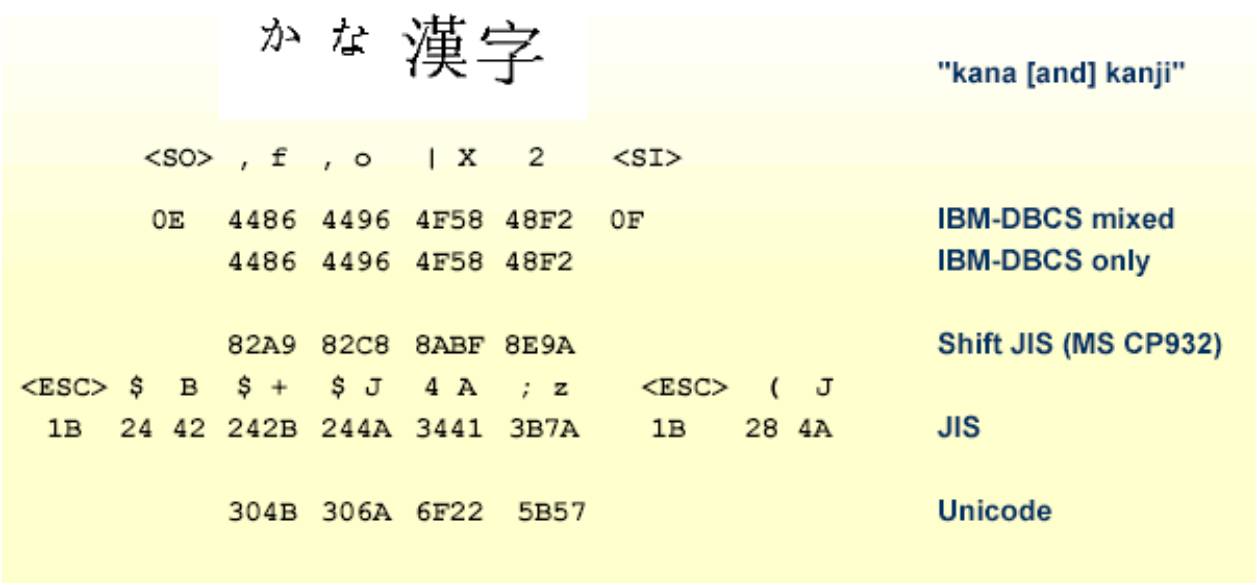
Although version 7 of Adabas continues to support the storage of DBCS-encoded data in alphanumeric fields, it introduces a wide-character (W) field format to store data with a well defined encoding and character set.

The default encoding for Wide format is Unicode for both storage and user. This default can be changed on user and storage level to the encoding appropriate for the intended usage.

In the figure below, the Japanese Kana (first two) and kanji (second two) characters are encoded in mainframe modal (mixed) and non-modal (pure)

- DBCS for use on EBCDIC-based machines
- JIS for use on ASCII-based machines

and in Unicode, a fixed 2-byte encoding that is more universal than the other encodings and is used as the default encoding in Adabas.



Wide-Character Encoding Example

Modal encodings shift back and forth between single- and double-byte character encodings. Mixed DBCS strings always start and end in single-byte mode.

Double-byte character only field lengths must be an even number of bytes.

For EBCDIC encodings, the padding or blank character is X’40’ or X’4040’. On Hitachi machines, the wide space is X’A1A1’ and the single byte space is X’40’. Adabas allows a single byte space to appear in double-byte mode without a mode switch.

Wide-Character Data Support

Adabas supports wide-character data with

- extended alphanumeric format fields; and
- wide-character format fields.

For an existing database or file, the encoding is assigned to alpha or wide fields using the ADADBS utility without an unload/reload. The field-level option NV (pass a field unconverted to/from a caller) is available.

- Extended Alphanumeric Fields
- Wide-Character Fields
- Special DBCS Format Conversion Rules

Extended Alphanumeric Fields

Adabas extends alphanumeric fields to support wide-character data by defining encoding keys on both the database and file levels: the file level encoding takes precedence over the database encoding. The encoding specifies the format in which the data is to be stored. It is also used as the default format in which data is exchanged with a local user.

The encoding must be compatible with EBCDIC; that is, the space character must be X’40’. For internal processing reasons, only one of the following encoding families is supported for a given file:

- EBCDIC (single-byte character set)
- mixed host-DBCS
- host-DBCS with DBCS-only option

Advantages and Disadvantages

The advantages of using extended alphanumeric fields include

- immediate support of existing databases that contain DBCS data;
- applications such as Natural continue running without changes; and
- no logic changes in the Adabas nucleus for calls from the same encoding/architecture since alphanumeric fields do not define an internal coding.

The disadvantage is that DBCS is not a universal encoding and unlike Unicode, it does not support all characters used in the world’s languages.

Limitations

For an application, all alphanumeric fields have the same encoding. It is not possible to use different encodings for different fields in the same session.

Conversion Considerations

When converting from pure single-byte character encodings, the field length of variable fields may change requiring a shift of the converted record.

Wide-Character Fields

Adabas defines a wide-character (W) format for fields. W format fields are similar to alphanumeric (A) format fields in that encoding keys are defined on both the database and file levels: the file encoding takes precedence over the database encoding. It differs from A field encoding in that

- if no encoding is specified, the default Unicode encoding is used.
- the internal encoding specifies the format in which the data is stored.
- the user encoding specifies the default format for data presented to the user.

A descriptor is stored (and sorted) with internal encoding.

Advantages and Disadvantages

The advantages of using wide-character (W) fields include the following:

- round-trip problems are avoided because the character set of the local encoding can be a superset of all character sets of user and special encodings;
- space is saved because internal encodings allow the use of UTF-8 when supported by ECS; and
- native Unicode (the user encoding), the standard Java text encoding, can be directly stored and retrieved.

The disadvantages are that

- Natural and other products do not immediately support the new format; and
- support for W format fields currently has the limitations listed in the next section, some of which may be resolved in future releases of Adabas.

Limitations

- For an application, all wide-character (W) fields have the same encoding. It is not possible to use different encodings for different fields in the same session.
- A W field cannot be the source for a phonetic descriptor or hyperdescriptor.
- Format conversions are not possible from numbers (U, P, B, F, G) to W format.
- A W field cannot be part of a coupled field, physical or soft.
- A W field cannot be part of a format selection criterion (conditional format). This limitation is due primarily to the single-byte character encoding of the criteria input (format buffer, search buffer, and utility).

- A W field cannot be part of a security-by-value criterion.
- A W field cannot be used with an edit mask.
- Format buffer literals are handled as unconvertible single-byte character strings.

Special DBCS Format Conversion Rules

To ensure a smooth transition from existing applications that use mixed-DBCS and DBCS-only data, special format conversion rules have been defined:

1. A modal DBCS encoding comprising the superset of single-byte and double-byte characters is treated as mixed-DBCS encoding for alphanumeric fields and as DBCS-only encoding for wide-character fields.
2. When converting from wide-character DBCS-only to the user's alphanumeric mixed-DBCS encoding, the encoding difference is ignored.

For example, if the user encoding for both alpha and wide formats is defined as DBCS and in the FDT, field AA is defined as alpha and field WW is defined as wide:

Format Buffer	Value in User Buffer
AA[,A]	mixed-DBCS
AA,W	DBCS-only
WW,A	DBCS-only
WW[,W]	DBCS-only