Multiple Platform Support

Prior to Adabas version 7, converting data for Adabas buffers between different machine architectures (ASCII, EBCDIC) was handled by Entire Net-Work. With the increasing use of applications where clients and servers (that is, the databases) have different encodings, it has become necessary to expand the data transfer and conversion capabilities of Adabas itself. To this end, Entire Net-Work determines whether the target database has translation capabilities, and if so, passes the unconverted data on to the database for conversion there.

An additional advantage of translating data within Adabas is that other transport mechanisms can now be supported. For UES-enabled databases, Adabas version 7 supports Entire Net-Work access to the z/OS mainframe database through the TCP/IP protocol from web-based applications or from PC-based applications such as Software AG's Jadabas. See the ADARUN parameters TCP and TCPURL in the Adabas Operations documentation for more information.

Adabas data translation occurs as follows:

- The client application can specify a special encoding and communicate it to the Adabas nucleus at session open (OP command).

- The LNKUES/ADALNK converts Adabas buffer data depending on the architecture of the caller.

- A number of utilities provide for special encoding and architecture settings.

EBCDIC to ASCII and ASCII to EBCDIC translation tables are located in the Adabas Installation documentation. A table listing the encoding keys provided with Adabas version 7 is located in the Adabas Command Reference documentation and the Adabas Utilities documentation.

This chapter covers the following topics:

- Encodings

- ADACOX Conversion Exit

- Conversion of High Value in Value Buffer

- Data Translation Restrictions

- Platform Considerations

Encodings

Adabas recognizes four types of encodings that can be specified in parallel:

Encoding	Character string encoding ...
File	is stored and processed internally
Default User	is used as the default for Adabas local call interface requests and for ADACMP DDEBAND.
User	overrides the default user encoding for a user session or an ADACMP execution. This is used to adapt to the special needs of a client program.
Collation	is in acceptable sort order. The collation can be defined by language and country standards commonly identified with a local definition.

Since user data does not require conversion, Adabas equates the local default user and file encoding to increase processing speed. Remote requests with ASCII architecture are converted using the database default ASCII user encoding.

Double-byte character sets are converted using the native mainframe EBCDIC architecture encoding: host DBCS from IBM, Fujitsu Technology Solutions, or Hitachi.

Special applications or remote clients select a specific user encoding that fits their processing environment at session open.

To ensure round-trip compatibility between architectures and encodings, Adabas uses a file encoding that holds the superset of all characters defined in the default user and any specific user encodings. For wide-character fields, such a file encoding defaults to the universal character set encoding Unicode.

Collation encoding is defined for a descriptor field. Values for this encoding are obtained algorithmically by calling a collation exit programmed to produce culturally correct sorted keys; that is, a character dictionary. Collation encoding may be defined for both alphanumeric and wide-character fields; the collation encoding/exit is defined on the file level for alpha and/or wide descriptor fields.

ADACOX Conversion Exit

A conversion exit, ADACOX, is available for your use. ADACOX supports *context-sensitive* conversion between Windows-1256 and EBCDIC Arabic or EBCDIC Farsi code page with UES enabled databases.

While Arabic characters are *unshaped* in Windows-1256, the supported EBCDIC encodings use *shaped* forms depending on previous or following characters. In addition, for certain consecutive characters the combined form is used e.g. LAM-ALEF ligature.

Currently, no support is included for conversion from logical to visual order and vice versa, or symmetric character conversion.

The conversion exit will always be loaded for UES enabled databases.

When a new conversion between two encodings is first used, the exit is queried whether it supports the conversion. If it does, the exit will be called for any such conversion; if it does not, Adabas and/or Entire Conversion Services will do the conversion.

However, the conversions defined in ADACOX need to be backed by corresponding ECS objects; for example, for the conversion 420 to 1256 the character set properties are determined by the ECS objects.

Conversion of High Value in Value Buffer

When performing searches using the S operator, the high value is usually a sequence of X'FF' bytes.

With UES=YES, the source and target code pages control the conversion of data. The conversion of the character X'FF' depends on its mapping to the target code page. It is therefore possible that the X'FF' will not remain the X'FF' in the converted value.

For example, when converting from 819 (ISO 8859-1 Latin1) to 37 (EBCDIC), the Latin small letter 'y' with diaeresis is mapped from X'FF' to X'DF'. As a result, searches find fewer or even no records.

This problem is solved as follows. With UES=YES and Alpha (or Wide) conversion, all FROM-TO Search/Logical Read Criteria are handled in such a way that in the TO criterion the high-value characters at the value end are preserved when converted into the internal search value and excluded from value conversion.

Note:
This solution is not implemented for the value operators (EQ, GT, GE, LE, and LT). It is limited to the TO value of FROM-TO search criteria (S operator). This applies to alpha and wide-format fields and to the Alpha/Wide format parts of Super and Sub Descriptors.

Data Translation Restrictions

The following restrictions of Entire Net-Work are continued with Adabas translation:

- compressed records (FB=C) are not converted.

- text literals are not converted and are passed as is. When reading records, a literal is returned unchanged (for example, FB AA, '-do not convert-',BB).

- prefetch option P is not supported in conversion.

- ET data is not converted. When reading, ET data is padded with EBCDIC blanks.

Additional restrictions imposed by Adabas include the following:

- for all C'Xn' command codes used by CSCI, only the control block is converted; not the buffers. This applies only for Adabas version 7 servers.

- Entire System Server (NPR) / XCOM applications are not included in the scope of Adabas translation. Those applications need to do their own translation.

- OS/2 unpacked numbers sign X'Dn' is not used and therefore, is not supported.

- Adabas does not provide user translation exits for field-level translation. Such exits are provided by Entire Net-Work.

Platform Considerations

Although differences between Adabas versions running on various platforms are gradually being reduced, the following considerations apply when porting applications:

	Mainframe	Open Systems	OpenVMS
Fixpoint field length	2 and 4 only	1,2,4,8	1,2,4,8
Binary superdescriptor format default is U (unpacked)	No	Yes	Yes
Signed binary superdescriptor	Yes	No	No
Binary superdescriptor format conversion	Yes	No	No
Superdescriptor with MU and PE fields	Yes	No	No
Superfields and subfields	Yes	No	No
Superdescriptor with floating-point format parents	No	Yes	Yes
Maximum length of unpacked fields	29	29	27
Maximum length of packed fields	15	15	14
Prefetch option for read and ET/BT commands	Yes	No	No
Long alpha (LA) field option	Yes	No	No
Field arithmetic update option in format buffer	No	Yes	Yes
MC command	No	Yes	Yes
Hyperfield value generation from value buffer	No	Yes	Yes

Additionally, user data provided in mainframe ADALNK user exits is not sent to ASCII machines.