

# Enabling Universal Encoding Support (UES) for Your Adabas Nucleus

Prior to Adabas Version 7, Entire Net-Work converted all data for mainframe Adabas when necessary from ASCII to EBCDIC. Starting with Version 7, Adabas is delivered with its own data conversion capability called *universal encoding support (UES)*. Entire Net-Work detects when it is connected to a target database that converts data and passes the data through to Adabas without converting it.

Universal encoding support must be activated in:

- The Adabas nucleus.
- The Adabas link routines. For Adabas Version 7, UES is enabled by default for the link routines ADALNK, ADALNKR, and ADALCO. For Adabas 8, UES is enabled by default for *all* link routines. For information on altering UES enablement in the link routines read appropriate sections of *Installing Adabas With TP Monitors*, starting with the section *UES-Enabled Link Routines*.

UES-enabled databases can be connected to machines with different architectures through Com-plete, Software AG internal product software (APS), Entire Net-Work (WCP), and optionally in a z/OS environment through a direct TCP/IP link to the Adabas nucleus from web-based applications or from PC-based applications such as Software AG's Jadabas. Connections through Com-plete or the Software AG internal product software (APS) use the Adabas Com-plete link routines; connections through Entire Net-Work use the Adabas batch link routines. Connections through a direct TCP/IP link are described in this chapter.

- The Adabas database. A sample startup job for a UES-enabled nucleus is provided in member ADANUCU of the ADAvrs.JOBS data set. For more information, read *JCL Required for UES Support (z/OS)* and *JCL Required for UES and TCP/IP Support (z/OS)*. In addition, read *Universal Encoding Support (UES)* as well as *ADADEF Utility: Define a Database* and *ADACMP Utility: Compress-Decompress Data* for more information.

## Note:

The use of UES-enabled link routines and a UES-enabled nucleus is transparent to applications, including applications that do not require universal encoding translation support. Therefore, it is not necessary to disable UES if it is already enabled.

This chapter covers the following topics:

- Connection Through a Direct TCP/IP Link
- Activating the TCP/IP Link

---

## Connection Through a Direct TCP/IP Link

A TCP/IP link requires in addition that you link a reentrant ADALNKR module with customized and reassembled translation tables and that you make the result available in the Adabas steplib. The Adabas 8.1.3 ADALNKR module is supplied with the LNKUES module and the default translation tables ASC2EBC and EBC2ASC.

UES-enabled databases are connected directly through TCP/IP using the Adabas reentrant batch or TSO link routine ADALNKR. The sample jobstream to link the ADALNKR module with your modified translation tables is ALNKUESR.

- Step 1: Assemble the Two Translation Tables into the Adabas Load Library (SMA Job Number I056)
- Step 2: Link the Translation Tables and LNKUES into ADALNKR
- Step 3: Make ADALNKR Available to the Adabas Nucleus

### **Step 1: Assemble the Two Translation Tables into the Adabas Load Library (SMA Job Number I056)**

Assemble the ASCII to EBCDIC and EBCDIC to ASCII translation tables, either default or customized. Use job ALNKUESR in the ADA*vrs*.JOBS library as a guide for customizing site-specific JCL.

### **Step 2: Link the Translation Tables and LNKUES into ADALNKR**

It is now necessary to (re)link ADALNKR with LNKUES and your customized and reassembled translation tables.

Link the ADALNKR, ASC2EBC, EBC2ASC, LNKUES, and other user exit modules into a final ADALNKR module that is UES-enabled. Place this load module into a “USER.LOAD” library. Sample job LNKLNKR8 is provided as a guide in the ADA*vrs*.JOBS library.

### **Step 3: Make ADALNKR Available to the Adabas Nucleus**

The (re)linked ADALNK must be made available to the Adabas nucleus.

If you are calling Adabas Version 8 directly through a TCP/IP link and the correct ADALNKR is not available to the Adabas nucleus, Adabas produces unexpected results, such as response code 148 and empty buffers.

## **Activating the TCP/IP Link**

 **To activate a direct TCP/IP link to the Adabas nucleus:**

1. Set the ADARUN parameter TCPIP=YES.
2. Specify a universal resource locator (URL).

### **Specifying a URL**

The URL is a 20-byte address that conforms to the RFC specification for URLs.

You can specify the URL required to activate the direct TCP/IP link in the ADARUN parameter TCPURL as follows:

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=api-name://stackid:port-number
```

where:

- *api-name* is a 1-3 character value identifying the application programming interface (API) to use. The APIs for the IBM TCP/IP stack (HPS, OES) are currently supported.
- *stackid* is a 1-8 character value identifying the stack to use: - for the HPS API, this is the name of the TCP/IP started task. - for the OES API, no value is needed. - for the ILK API, this is the subsystem identifier.
- *port-number* is a 1-5 character number in decimal notation.

### Examples

```
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=HPS://STACKNAME:1234
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=OES://:1234
ADARUN PROG=ADANUC,TCPIP=YES,TCPURL=ILK://ILZ5:1234
```

## Managing URLs

Optionally, you can specify the first and additional URLs using the operator command TCPIP:

```
TCPIP={ OPEN=url|CLOSE=url | CLOSE }
```

where *url* is the URL for the TCP/IP link you want to open or close and has the same format as the ADARUN TCPURL parameter:

```
api-name://stackid:port-number
```

The command allows you to open or close a TCP/IP link to the Adabas nucleus or to close all links. It can only be used when ADARUN TCPIP=YES and all conditions for that setting have been met. This command can be used to close the URL set in the ADARUN TCPURL parameter, or to open/close additional TCP/IP links.

### Examples

```
TCPIP=OPEN=ILK://ILZ5:1234
TCPIP=CLOSE=ILK://ILZ5:1234
```

```
To close all open URLs:
TCPIP=CLOSE
```