

Adabas for Linux, UNIX and Windows

Code Page Support, Code Page User Exit

Version 7.0

October 2022

This document applies to Adabas for Linux, UNIX and Windows Version 7.0 and all subsequent releases.

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

Copyright © 1987-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: ADAOS-CPS-70-20220622

Table of Contents

Code Page Support	v
1 About this Documentation	1
Document Conventions	2
Online Information and Support	2
Data Protection	3
2 Code Page Support	5
3 Code Page User Exit	7
Function Interface	9
Default Translation Tables	11

Code Page Support

This document contains information about code page support and the code page user exit.

The following topics are covered:

- *Code Page Support*
- *Code Page User Exit*

1

About this Documentation

■ Document Conventions	2
■ Online Information and Support	2
■ Data Protection	3

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Monospace font	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	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.software-ag.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 Code Page Support

In heterogeneous client/server architectures, where the client and servers reside on machines with different architectures, Adabas supports a method whereby the client nodes that require different character sets (also called code pages) are provided with the appropriate translation tables for the conversions. An example of this would be an Adabas nucleus running on a UNIX machine, with one application running on a mainframe (EBCDIC code) and another application running on a language-sensitive PC; in this case, the nucleus would require 2 translation tables in order to handle the data from both applications correctly.

Code page support is available for:

- ADANUC (ADABAS nucleus)
- ADACMP (compression utility)
- ADADCU (decompression utility)

Adabas supports the use of plain text files as private code pages. In addition to this, DLLs or shared libraries can be used (see [Code Page User Exit](#) below for further details). The translation text file to be used is attached to Adabas with the environment variable ADATRT, but the file must have the extension `.txt` (or `.TXT`), for example:

```
ADATRT = trt.dll # use code pages with built dll
ADATRT = trt.txt # use code pages with text file
```

The syntax of such a text file is as follows:

- Characters and names are not case sensitive.
- A table declaration must start with the code name, either ASCII or EBCDIC. The ASCII table can be used for remote ASCII clients running on a different code page, the EBCDIC table can be used for mainframe clients. One ASCII and/or EBCDIC table may be translated with the text file.

- The character pairs to be translated are specified in hexadecimal notation, separated by a comma. The first character specifies the client value, the second specifies the (ASCII) value on the server side. Only one pair per line is permitted.
- Only the characters that are different need to be supplied, e.g. the translation from EBCDIC “A” to ASCII “A” can be omitted.
- Empty lines are permitted and comments can be made starting with a semicolon.

The following is an example of how to define a private EBCDIC translation file:

```
trt.txt:
EBCDIC
4A,C4    ; translate EBCDIC Ae (0x4A) to ASCII C4
E0,D6    ; translate EBCDIC 0e (0xE0) to ASCII D6
```



Note: The use of text files and shared libraries is mutually exclusive.

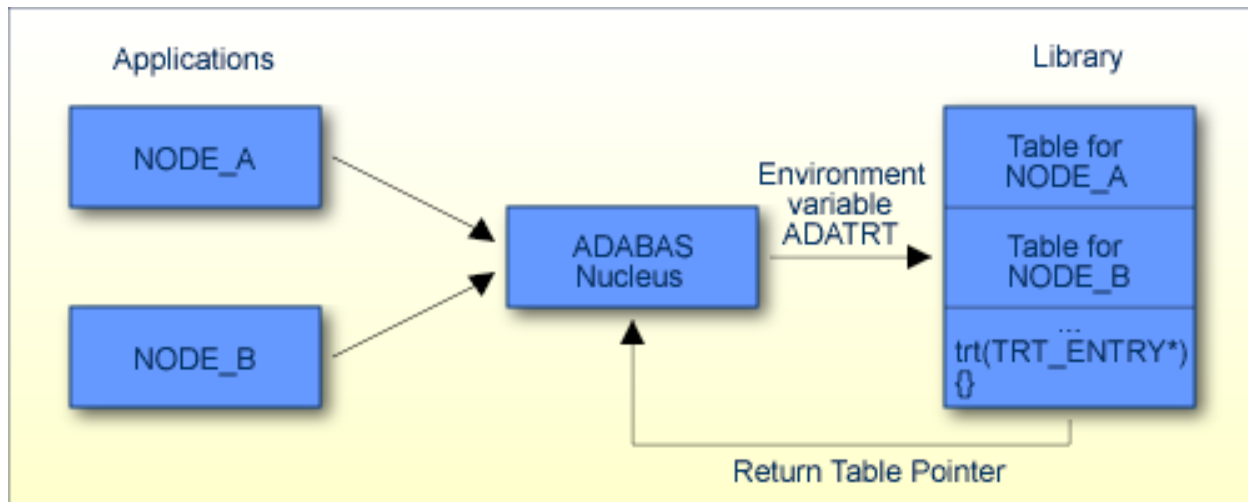
3

Code Page User Exit

■ Function Interface	9
■ Default Translation Tables	11

In the past, ADABAS had a default internal EBCDIC/ASCII translation table. But in order to provide full flexibility with specific translation tables in complex client/server systems, ADABAS also supports a special code-page user exit.

The ADABAS nucleus calls the user exit when it processes a command that was issued from a remote node. The user exit is called with a pointer to a structure that contains the node name (see the following diagram). The user exit now uses a user-written algorithm to determine which translation table to use, and returns a pointer to a table. Adabas then uses this table for the conversions.



The environment variable ADATRT holds the name of the shared library (UNIX) or dynamic link library (Windows) that is created by the user with the appropriate options. Plain text files with the extension .txt can also be used (see *Code Page Support* for further information). The default name for the user function is 'trt', but any name can be selected, for example:

```
setenv ADATRT "$ADADATADIR/db100/adatrt.sl" (UNIX C shell)
set ADATRT= %ADADATADIR%\db100\adatrt.dll (Windows)
```

or

```
setenv ADATRT "adatrt.so my_trt" (UNIX C shell)
set ADATRT=adatrt.dll my_trt (Windows)
```

The environment variable ADATRT must be set before the nucleus or a utility is started. If it is not defined, the default internal translation table will be used if an EBCDIC/ASCII conversion is required. If ADATRT is defined, but the library cannot be loaded successfully, the nucleus will abort during its startup phase and a corresponding error message will be issued.

Function Interface

```
Format: unsigned char *trt (struct trt_entry *)
```

The trt Structure

trt_node

The basic input is the name of the node that issues the current ADABAS call. On the basis of the node name, the function selects the appropriate code page table. If selected, the function returns a pointer to the table. If nothing is selected, a NULL pointer may be specified to indicate that either no or just the default conversions are to be made.

trt_ctype

The function is called with one of two values. A TRT_INIT call will be performed during the nucleus/utility startup: it will return a value of NULL if it is successful. A TRT_CODE_PAGE call has the node name set so that the function can select a translation table. If the function is called from the utilities ADACMP or ADADCU, the TRT_CODE_PAGE call will be performed once, and the node name will be 'adacmp' or 'adadcu'.

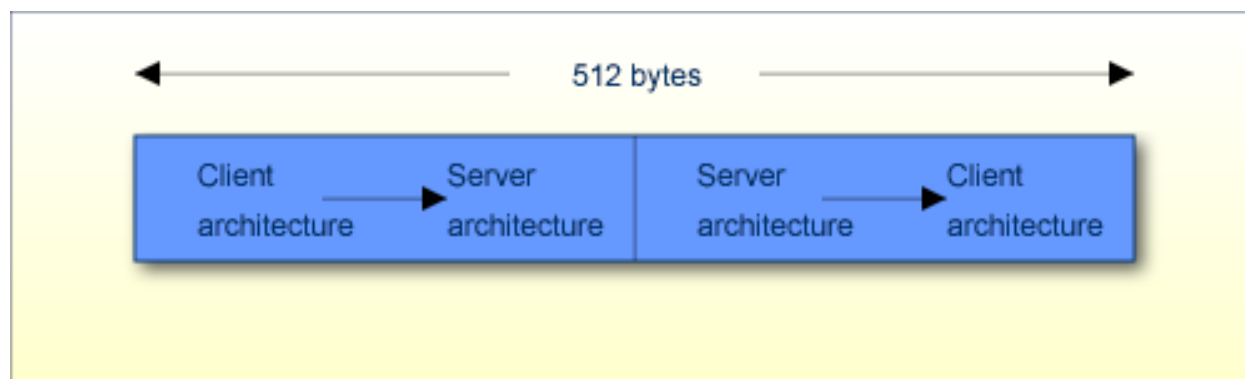
trt_dbid

This specifies the current database number. This is necessary because the translation table function may be shared between several databases.

Translation Table Layout

Translation tables must be 512 bytes long. The first half is used for translations from the client machine architecture to the server architecture; the second 256 bytes are used for the reverse conversions.

Each table must be a real translation table. This means that the value of the input character is the index to the table, and that the character stored in the corresponding cell will be taken.



The user may use as many translation tables as his heterogeneous environment requires.

Calling Conventions

The following rules apply to the use of code pages:

- The user-written function is always called if the current Adabas call comes from a remote node and is a read or update command.
- If the user-written function returns a pointer (to a code page), it will be used for the conversion. Only alphanumeric (format A) data will be converted.
- If the user-written function returns a NULL and both the remote and local architectures are the same, no conversion will be performed.
- If the user-written function returns a NULL and the remote and local architectures are different, the default Adabas conversion table will be used for the conversion.

Creating a Code Page User Exit

The Adabas kit contains a corresponding C header file and an example source file for a code page user exit.

UNIX Platforms

The structure `trt_entry` is defined in the file `adaux.h`, which is located in `$ADAPROGDIR/inc`.

An example source file for the code page user exit (file name `adatrt.c`) and the corresponding makefile are located in `$ADAPROGDIR/examples/server`.

Enter the following in order to build this user exit example:


```
cd $ADAPROGDIR/examples/server
make trt
```

The shared library for the user exit is created in \$ADAPROGDIR/examples/server.

Windows Platforms

The structure `trt_entry` is defined in the file `adaux.h`, which is located in the subdirectory “Adabas\inc” of the installation directory.

An example source file for the code page user exit (file name `adatrt.c`) and the corresponding makefile are located in the subdirectory “Adabas\examples\server” of the installation directory.

Enter the following in order to build this user exit example:

```
cd <install_dir>\Adabas\examples\server
nmake trt
```

The DLL for the user exit created in the subdirectory “Adabas\examples\server” of the installation directory.

Default Translation Tables

The following are the default translation tables used by Adabas.

Conversion ASCII ISO-8-bit (8859-1) to EBCDIC



Note: Undefined characters not contained in the table will be converted to 0xFF.

Decimal Value	ASCII Hex Value	ASCII Character	EBCDIC Hex Value
0	0x00		0x00
1	0x01		0x01
2	0x02		0x02
3	0x03		0x03
4	0x04		0x37
5	0x05		0x2D
6	0x06		0x2E
7	0x07		0x2F
8	0x08		0x16
9	0x09		0x05

Decimal Value	ASCII Hex Value	ASCII Character	EBCDIC Hex Value
10	0x0A		0x25
11	0x0B		0x0B
12	0x0C		0x0C
13	0x0D		0x0D
14	0x0E		0x0E
15	0x0F		0x0F
16	0x10		0x10
17	0x11		0x11
18	0x12		0x12
19	0x13		0x13
20	0x14		0x3C
21	0x15		0x3D
22	0x16		0x32
23	0x17		0x26
24	0x18		0x18
25	0x19		0x19
26	0x1A		0x3F
27	0x1B		0x27
28	0x1C		0x1C
29	0x1D		0x1D
30	0x1E		0x1E
31	0x1F		0x1F
32	0x20	SP	0x40
33	0x21	!	0x4F
34	0x22	"	0x7F
35	0x23	#	0x7B
36	0x24	\$	0x5B
37	0x25	%	0x6C
38	0x26	&	0x50
39	0x27	'	0x7D
40	0x28	(0x4D
41	0x29)	0x5D
42	0x2A	*	0x5C
43	0x2B	+	0x4E
44	0x2C	,	0x6B
45	0x2D	-	0x60

Decimal Value	ASCII Hex Value	ASCII Character	EBCDIC Hex Value
46	0x2E	.	0x4B
47	0x2F	/	0x61
48	0x30	0	0xF0
49	0x31	1	0xF1
50	0x32	2	0xF2
51	0x33	3	0xF3
52	0x34	4	0xF4
53	0x35	5	0xF5
54	0x36	6	0xF6
55	0x37	7	0xF7
56	0x38	8	0xF8
57	0x39	9	0xF9
58	0x3A	:	0x7A
59	0x3B	;	0x5E
60	0x3C	<	0x4C
61	0x3D	=	0x7E
62	0x3E	>	0x6E
63	0x3F	?	0x6F
64	0x40	@	0x7C
65	0x41	A	0xC1
66	0x42	B	0xC2
67	0x43	C	0xC3
68	0x44	D	0xC4
69	0x45	E	0xC5
70	0x46	F	0xC6
71	0x47	G	0xC7
72	0x48	H	0xC8
73	0x49	I	0xC9
74	0x4A	J	0xD1
75	0x4B	K	0xD2
76	0x4C	L	0xD3
77	0x4D	M	0xD4
78	0x4E	N	0xD5
79	0x4F	O	0xD6
80	0x50	P	0xD7
81	0x51	Q	0xD8

Decimal Value	ASCII Hex Value	ASCII Character	EBCDIC Hex Value
82	0x52	R	0xD9
83	0x53	S	0xE2
84	0x54	T	0xE3
85	0x55	U	0xE4
86	0x56	V	0xE5
87	0x57	W	0xE6
88	0x58	X	0xE7
89	0x59	Y	0xE8
90	0x5A	Z	0xE9
91	0x5B	[0x4A
92	0x5C	\	0xE0
93	0x5D]	0x5A
94	0x5E	^	0x5F
95	0x5F	_	0x6D
96	0x60	`	0x79
97	0x61	a	0x81
98	0x62	b	0x82
99	0x63	c	0x83
100	0x64	d	0x84
101	0x65	e	0x85
102	0x66	f	0x86
103	0x67	g	0x87
104	0x68	h	0x88
105	0x69	i	0x89
106	0x6A	j	0x91
107	0x6B	k	0x92
108	0x6C	l	0x93
109	0x6D	m	0x94
110	0x6E	n	0x95
111	0x6F	o	0x96
112	0x70	p	0x97
113	0x71	q	0x98
114	0x72	r	0x99
115	0x73	s	0xA2
116	0x74	t	0xA3
117	0x75	u	0xA4

Decimal Value	ASCII Hex Value	ASCII Character	EBCDIC Hex Value
118	0x76	v	0xA5
119	0x77	w	0xA6
120	0x78	x	0xA7
121	0x79	y	0xA8
122	0x7A	z	0x9A
123	0x7B	{	0xC0
124	0x7C		0x6A
125	0x7D	}	0xD0
126	0x7E	~	0xA1
127	0x7F		0x07
196	0xC4	Ä	0x4A
214	0xD6	Ö	0xE0
220	0xDC	Ü	0x5A
223	0xDF	ß	0xA1
228	0xE4	ä	0xC0
246	0xF6	ö	0x6A
252	0xFc	ü	0xD0

Conversion EBCDIC to ASCII ISO-8-bit (8859-1)



Note: Undefined characters not contained in the table will be converted to 0x7F.

Decimal Value	EBCDIC Hex Value	EBCDIC Character	ASCII Hex Value
0	0x00		0x00
1	0x01		0x01
2	0x02		0x02
3	0x03		0x03
5	0x05		0x09
7	0x07		0x7F
11	0x0B		0x0B
12	0x0C		0x0C
13	0x0D		0x0D
14	0x0E		0x0E
15	0x0F		0x0F
16	0x10		0x10
17	0x11		0x11

Decimal Value	EBCDIC Hex Value	EBCDIC Character	ASCII Hex Value
18	0x12		0x12
19	0x13		0x13
22	0x16		0x08
24	0x18		0x18
25	0x19		0x19
28	0x1C		0x1C
29	0x1D		0x1D
30	0x1E		0x1E
31	0x1F		0x1F
37	0x25		0x0A
38	0x26		0x17
39	0x27		0x1B
45	0x2D		0x05
46	0x2E		0x06
47	0x2F		0x07
50	0x32		0x16
55	0x37		0x04
60	0x3C		0x14
61	0x3D		0x15
63	0x3F		0x1A
64	0x40	SP	0x20
74	0x4A	Ä	0xC4
75	0x4B	.	0x2E
76	0x4C	<	0x3C
77	0x4D	(0x28
78	0x4E	+	0x2B
79	0x4F	!	0x21
80	0x50	&	0x26
90	0x5A	Ü	0xDC
91	0x5B	\$	0x24
92	0x5C	*	0x2A
93	0x5D)	0x29
94	0x5E	;	0x3B
95	0x5F	^	0x5E
96	0x60		0x2D
97	0x61	/	0x2F

Decimal Value	EBCDIC Hex Value	EBCDIC Character	ASCII Hex Value
106	0x6A	ö	0xF6
107	0x6B	,	0x2C
108	0x6C	%	0x25
109	0x6D	–	0x5F
110	0x6E	>	0x3E
111	0x6F	?	0x3F
121	0x79	`	0x60
122	0x7A	:	0x3A
123	0x7B	#	0x23
124	0x7C	@	0x40
125	0x7D	'	0x27
126	0x7E	=	0x3D
127	0x7F	"	0x22
129	0x81	a	0x61
130	0x82	b	0x62
131	0x83	c	0x63
132	0x84	d	0x64
133	0x85	e	0x65
134	0x86	f	0x66
135	0x87	g	0x67
136	0x88	h	0x68
137	0x89	i	0x69
145	0x91	j	0x6A
146	0x92	k	0x6B
147	0x93	l	0x6C
148	0x94	m	0x6D
149	0x95	n	0x6E
150	0x96	o	0x6F
151	0x97	p	0x70
152	0x98	q	0x71
153	0x99	r	0x72
161	0xA1	ß	0xDF
162	0xA2	s	0x73
163	0xA3	t	0x74
164	0xA4	u	0x75
165	0xA5	v	0x76

Decimal Value	EBCDIC Hex Value	EBCDIC Character	ASCII Hex Value
166	0xA6	w	0x77
167	0xA7	x	0x78
168	0xA8	y	0x79
169	0xA9	z	0x7A
192	0xC0	ä	0xE4
193	0xC1	A	0x41
194	0xC2	B	0x42
195	0xC3	C	0x43
196	0xC4	D	0x44
197	0xC5	E	0x45
198	0xC6	F	0x46
199	0xC7	G	0x47
200	0xC8	H	0x48
201	0xC9	I	0x49
208	0xD0	ü	0xFC
209	0xD1	J	0x4A
210	0xD2	K	0x4B
211	0xD3	L	0x4C
212	0xD4	M	0x4D
213	0xD5	N	0x4E
214	0xD6	O	0x4F
215	0xD7	P	0x50
216	0xD8	Q	0x51
217	0xD9	R	0x52
224	0xE0	Ö	0xFC
226	0xE2	S	0x53
227	0xE3	T	0x54
228	0xE4	U	0x55
229	0xE5	V	0x56
230	0xE6	W	0x57
231	0xE7	X	0x58
232	0xE8	Y	0x59
233	0xE9	Z	0x5A
240	0xF0	0	0x30
241	0xF1	1	0x31
242	0xF2	2	0x32

Decimal Value	EBCDIC Hex Value	EBCDIC Character	ASCII Hex Value
243	0xF3	3	0x33
244	0xF4	4	0x34
245	0xF5	5	0x35
246	0xF6	6	0x36
247	0xF7	7	0x37
248	0xF8	8	0x38
249	0xF9	9	0x39
250	0xFA		0xFA
255	0xFF		0xFF

