

# Translation Tables

---

This chapter describes the translation tables available for Cisco IOS for S/390. It includes these sections:

- **Cisco IOS for S/390 Character Translation**  
Describes the character translation facilities of Cisco IOS for S/390
- **Choosing a Telnet Translation Table**  
Details factors to consider when selecting a Telnet translation table
- **National-Use Characters**  
Describes the National-Use characters
- **Modifying or Adding Translate Tables**  
Describes how to add new translate table or edit the existing tables
- **Structure of Translate Table Modules**  
Details information about the structure of translate tables provided with Cisco IOS for S/390
- **Generating Prefixes for National Language Translate Tables**  
Describes the use of the XLTBL macro

## Cisco IOS for S/390 Character Translation

Cisco IOS for S/390 provides character translation for both the data connection and the Telnet connection. Telnet connections are made for these services:

- Telnet
- SMTP
- LPR
- FTP (generally called the control connection).

FTP also makes a data connection when files are transferred. During translation, ASCII-encoded data on the TCP/IP network is converted to EBCDIC-encoded data on the MVS system, and vice-versa. Translation tables for several languages are distributed with Cisco IOS for S/390 in both source and load format. You can specify any of the tables provided, modify them, or define and use your own translation tables.

The Telnet connection for an application uses the translate table specified by the TRANTBL parameter on the appropriate APPCFGxx configuration statement; the data connection uses the translate table specified by the CHARSET parameter. If these parameters are not coded in the configuration table, the ENGLISH translate table is the default.

Data connection translation can be done with both single- and double-byte translate tables. Telnet connection translation requires single-byte tables. The single-byte character set translate tables distributed with the product are ENGLISH (the default), DANISH, FCANADA, FRENCH, GERMAN, GSWISS, ITALIAN, KANJI83, KOREAN, JAPANES, SJIS83, SPANISH, SWEDISH, SW850 SW88591 and THAI. Distributed double-byte tables are HANGUL and KANJI. The source code for these tables is located in the SAMP data set and their load module forms are in the LOAD data set.

## Translate Table Specification

The TRANTBL parameter can be specified on the following APPCFGxx statements: Telnet, APPL, FTP, SMTP and LPR. CHARSET can be specified on the FTP statement. Read the specific application chapter for more information about each of these statements and how the table name should be specified.

## Character Table Usage Notes

Consider these notes when specifying translate tables:

- Telnet translation has no impact on 3270-based Telnet connections because they use binary mode.
- If no TRANTBL or CHARSET information is provided, the ENGLISH table is used for all translation.
- If TRANTBL is specified on the HOST statement only, the table named is used for all other Telnet services.
- If TRANTBL is specified for a service (for example, SMTP), the table named will be used for that service.
- TRANTBL on the Telnet statement becomes the default for all line mode APPL statements, unless it is specified on individual APPL statements.
- If TRANTBL is specified on an APPL statement, the table named will be used for that application.
- Regardless of TRANTBL settings, if CHARSET is not specified on the FTP statement, data translation defaults to ENGLISH.
- CHARSET can be overridden for individual FTP sessions by specifying it in SITE parameters. Read the *Cisco IOS for S/390 User's Guide*.

## Choosing a Telnet Translation Table

Because of translation differences between languages, when using a table other than English for Telnet translation, you should first do the following:

- Determine if any of the characters indicated in the following table will appear in your Telnet input or output stream.
- If there are any, check the translate table source to verify they will translate correctly.

The SMTP service uses the @ character as part of the mailing address. In the Danish table however, an EBCDIC X'7C' does not represent an @. Rather, in that table an ASCII X'40' is translated to an EBCDIC X'40' (which is a blank), and an EBCDIC X'7C' is translated to an ASCII X'5C'. Therefore, the Danish table cannot be used for SMTP.

## National-Use Characters

This table indicates the 14 national-use characters. The first row is the EBCDIC character representation for English, the second row is the EBCDIC hexadecimal equivalent, and the third row is the 7-bit ASCII hexadecimal equivalent. The translation of these characters may vary from language to language.

**Table 19-1 National Use Characters**

CHAR	¢	!		`	\$	#	@	¬	~	{	}	\		“
HEX(E)	4A	5A	6A	79	5B	7B	7C	5F	A1	C0	D0	E0	4F	7F
HEX(A)	no symbol	21	7C	60	24	23	40	5E	7E	7B	7D	5C	ns	22

This information was taken from a table in the *IBM 3270 Information Display System Character Set Reference*, GA27-2837.

## Modifying or Adding Translate Tables

If necessary, the translate tables can be modified to suit customer requirements. Also, other translate tables with similar structure can be added. It is recommended that new tables be developed using an existing table as a model.

## Structure of Translate Table Modules

A translate table module has a special structure containing a set of translate tables as described below. The national language translate table modules used by the Telnet and Telnet-based applications must have this structure:

```
Prefix - XLTBL TYPE=START
Set of translate tables
Suffix - XLTBL TYPE=FINISH
```

For Telnet purposes, there should be four translate tables in the set. An ATOE table for ASCII-to-EBCDIC translation, ETOA for EBCDIC-to-ASCII translation, AUPC for ASCII lower case to upper case folding, and EUPC for EBCDIC lower case to upper case folding, respectively. Each individual table is 256 bytes long and suitable for use with the IBM System/370 TR instruction. Also, the statement labels for these tables must be listed in the XLTBL macro TABLE parameter in the order listed above.

Data translation requires only the first two tables (ATOE and ETOA), as there are no case changes involved in the transfer. Various command line entries require all four translate tables in the set.

## Generating Prefixes for National Language Translate Tables

The XLTBL macro generates the prefix and suffix needed by a national language translate table module.

## XLTL Macro Syntax

The syntax for the XLTL macro is as follows:

## XLTL Macro Description

```
[name] XLTL [TYPE = (START | FINISH)]
[, KIND = (CHAR | NUMBS | DBCS)]
[, TABLE = label_name,...]
[, FLAG = (REFLEX | ASCII8 | ASCII16 | NOSOSI | BKTSUB)]
[, TITLE = 'char_string']
[, TAG = character]
[, ID = character (s)]
[, DSECT = (YES | NO)]
```

### Syntax Description

<b>TYPE</b> = (START   FINISH)	Specifies whether header or trailer data is to be generated. START generates the translate module prefix, FINISH its suffix. Default: START
<b>KIND</b> = (CHAR   NUMS   DBCS)	This keyword defines the translation set to be defined: CHAR: single byte, 4-table set NUMS: single byte, 2-table set Default: CHAR
<b>TABLE</b> = ( <i>label_name</i> , <i>label2</i> , <i>label3</i> ,...)	List of addresses for individual translate tables ( <i>label1</i> , <i>label2</i> ,...) contained in the translate table module. <i>label1</i> is the statement label for the first translate table, <i>label2</i> for the second, etc. The number of items depends upon KIND.  For Telnet translate tables, KIND=CHAR should be coded, and TABLE should list four labels for ATOE, ETOA, AUPC, and EUPC translate tables, in that order.  There is no default.
<b>FLAG</b> = (REFLEX   ASCII8   ASCII16)	List of options to be set in the flag area of the fixed header: REFLEX: ASCII to EBCDIC and EBCDIC to ASCII is symmetric. ASCII8: ASCII data is 8 bit (SBCS). ASCII16: ASCII data is 16 bit.  There is no default.
<b>TITLE</b> = ' <i>char_string</i> '	Specifies a character string used to describe the translation set. Length is limited to 30 characters.  The default is 'name Translation', where name is the label on the XLTL invocation statement.
<b>TAG</b> = <i>character</i>	One-byte character identifier used to group the soft key definitions. Default: \$

---

**ID** = *character(s)*

Three-byte character identifier used as a root for macro generated label names.

This needs to be changed only if the XLTBL macro is issued multiple times in the same TYPE.

Default: XS@

**DSECT** = YES | NO

YES generates a DSECT map of the data area.

NO generates a CSECT map of the data area.

When you create your own translation tables, the DSECT parameter should always be set to NO. Although the default value of YES is used by Cisco IOS for S/390 for internal applications, if YES is selected when you create a translation table, the translation table will be invalid.

When YES is specified the keywords TITLE, FLAG, and TABLE are ignored, so this parameter should always be set to NO.

Default: YES

## XLTBL Macro Example

```
ENGLISH XLTBL TYPE=START,  
KIND=CHAR,  
TABLE= ( ATOE, ETOA, AUPE, EUPA ) ,  
FLAG= ( ASCII8 ) ,  
TITLE='English Single byte table',  
DSECT=NO
```

## Maintaining Translation Tables Using UMODTRAN

The SMP/E usermod JCL stream UMODTRAN in the CNTL library provides for translation table maintenance. It is used for both modifying and adding tables. It will compile and link the table source into a load module in the LOAD library. Cisco IOS for S/390 uses only the load module version when it does translation.

SMP/E maintains all translate table source in the SAMP library. For existing tables, make changes in the SAMP member before executing UMODTRAN. Build new translate tables in a different library, as UMODTRAN will copy them into the SAMP library as part of its SMP/E processing. Once the table has been copied, all subsequent changes should be made in the SAMP library.

---

**Note** You should run an SMP/E APPLY CHECK against any USERMOD that you are trying to install, as there may be additional PREs on your system that are not accounted for. Once you gather this information, add the SYSMOD list(s) to the ++PRE(xxxxxx) statement. Then SMP/E REJECT the USERMOD to remove the invalid entry from the SMP/E CSI. You will then be able to RECEIVE/APPLY the USERMOD with success.

---

Directions for using UMODTRAN are located within the member. When needed, make a copy of the UMODTRAN provided and adapt it to your environment.

Sample JCL for UMODTRAN is shown below:

```
//UMODTRAN JOB
//*
//*      SAMPLE JCL STREAM TO ADD OR MODIFY TRANSLATE TABLES
//*
//*
//* *****
//*      *          *** INSTRUCTIONS ***
//* *****
//* * TO ADD OR MODIFY A TRANSLATE TABLE, DO THE FOLLOWING:
//* *
//* * 1. MAKE THE FOLLOWING GLOBAL CHANGES TO THIS JCL STREAM
//* *
//* * . CHANGE 'UMODNAM' TO A UNIQUE 7 CHARACTER USERMOD NAME
//* *   (IF THIS IS NOT THE FIRST UPDATE TO THE TABLE YOU WILL
//* *     NEED TO ADD A PRE PARAMETER TO THE ++VER STATEMENT
//* *     NAMING THE LAST USERMOD THAT WAS APPLIED.)
//* * . CHANGE 'TABNAME' TO THE TRANSLATE TABLE NAME.
//* * . CHANGE 'SMPINDX' TO THE SMP HIGH LEVEL QUALIFIER NAME.
//* * . CHANGE 'XXX' TO THE APPROPRIATE RELEASE NUMBER.*
//* * . CHANGE 'HOLDCL' TO YOUR SYSOUT HOLD CLASS.
//* *
//* * DO 2.A -OR- 2.B
//* *
//* * 2.A FOR UPDATING AN EXISTING TABLE:
//* *
//* * . VERIFY TABLE UPDATES WERE MADE IN THE SAMP LIBRARY
//* *   MEMBER PRIOR TO RUNNING THIS JOB.
//* *
//* * 2.B FOR ADDING A NEW TABLE:
//* *
//* * . POINT THE TEXTLIB DD STATEMENT TO THE LIBRARY
//* *   CONTAINING THE NEW TABLE SOURCE.
//* *
//* * . UNCOMMENT THE SYSLIB PARAMETER IN THE ++SRC
//* *   STATEMENT. (USERMOD WILL COPY TABLE TO SAMP LIBRARY)
//* *
//* * 3. VERIFY THE JOB STATEMENT IS CORRECT AND SUBMIT THE JOB.
//*
//*****
//*
//*
//SMPE      EXEC PGM=GIMSMP,REGION=4096K,TIME=960,
//          PARM='CSI=SMPINDX.CSI,PROCESS=WAIT'
//SMPHOLD   DD DUMMY
//SMPLOG    DD DSN=SMPINDX.SMPLOG,DISP=MOD
//SMPOUT    DD SYSOUT=HOLDCL
//TEXTLIB   DD DSN=SMPINDX.SAMP,DISP=SHR
//SMPPTFIN  DD *
++ USERMOD ( UMODNAM ) .
++ VER (Z038) FMID(T00XXX) .
++JCLIN .
//JCLINASM JOB ( COMPANY ), 'TRANSLATE',MSGCLASS=X
//ASM      EXEC PGM=ASMBLR,
//          REGION=512K,
//          PARM='OBJ,NODECK,LIST,NORENT'
//*
//SYSIN    DD DSN=IOS390.ATCPSAMP(TABNAME) ,
//          DISP=( SHR,KEEP,KEEP )
//*
//SYSGO    DD DSN=&&PUNCH ( TABNAME ) ,
//          UNIT=3380,
//          SPACE=( 80,( 200,50 ) ) ,
```

---

```
//          DISP=(MOD,PASS)
//JCLINLNK JOB ( COMPANY ), 'UMOD XLATE',MSGCLASS=X
//LKED      EXEC PGM=IEWL,PARM='REUS,OL,LIST,NCAL',REGION=512K
//SYSLMOD   DD DSN=IOS390.TCPLOAD,DISP=SHR
//ATCPLOAD  DD DSN=IOS390.ATCPLOAD,DISP=SHR
//SYSLIN    DD *
INCLUDE ATCPLOAD ( TABNAME )
MODE AMODE ( 31 ),RMODE ( ANY )
ENTRY $TABNAME
NAME TABNAME ( R )
/*
++SRC ( TABNAME ) TXLIB ( TEXTLIB )
DISTMOD ( ATCPLOAD )
/* SYSLIB(TCPSAMP)          */
DISTLIB ( ATCPSAMP ) .
//SMPCNTL  DD *
SET BDY ( GLOBAL ) .
RECEIVE S ( UMODNAM ) .
SET BDY ( TCPTZN ) .
APPLY S ( UMODNAM ) .
/*
```

