

UNISYS e-@ction CLEARPATH ENTERPRISE SERVERS

MultiLingual System (MLS) Administration, Operations, and Programming Guide

ClearPath MCP Release 7.0

November 2001

Printed in USA
8600 0288-305

UNISYS e-@ction CLEARPATH ENTERPRISE SERVERS

MultiLingual System (MLS) Administration, Operations, and Programming Guide

The UNISYS logo is displayed in a bold, black, serif font. The letter 'i' in 'UNISYS' is lowercase and has a dot above it, while all other letters are uppercase.

© 2001 Unisys Corporation.
All rights reserved.

ClearPath MCP Release 7.0

November 2001

Printed in USA
8600 0288-305

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Notice to Government End Users: This is commercial computer software or hardware documentation developed at private expense. Use, reproduction, or disclosure by the Government is subject to the terms of Unisys standard commercial license for the products, and where applicable, the restricted/limited rights provisions of the contract data rights clauses.

Correspondence regarding this publication can be e-mailed to doc@unisys.com.

Unisys e-@ction
ClearPath Enterprise
Servers

MultiLingual System (MLS)

**Administration,
Operations, and
Programming Guide**

**ClearPath MCP
Release 7.0**

Unisys e-@ction
ClearPath
Enterprise
Servers

MultiLingual
System (MLS)

**Administration,
Operations, and
Programming
Guide**

**ClearPath MCP
Release 7.0**

8600 0288-305

8600 0288-305

Bend here, peel upwards and apply to spine.

Contents

Section 1. Understanding the MultiLingual System (MLS)

About This Guide	1-1
How to Use This Guide	1-2
Understanding the MultiLingual System (MLS)	1-3
What Is MLS?	1-3
Understanding MLS Terminology	1-4
What Is an Application System?	1-4
What Is a MultiLingual System?	1-4
How Are Cultures and Localized Environments Supported?	1-5
What Processes Are Performed in the MLS Environment?	1-7
Understanding the Components of the MLS Environment	1-8
Coded Character Sets	1-8
Ccsversions	1-11
Languages	1-15
Convention Definitions	1-15
Templates	1-18
HostCCS	1-18

Section 2. Locating and Interfacing with MLS Components

Where Are the MLS Components Defined?	2-1
SYSTEM/CCSFILE File	2-2
SYSTEM/CONVENTIONS File	2-2
CENTRALSUPPORT Library	2-3
Support for Natural Languages	2-3
Interfaces to the MLS Environment	2-4
Tools for Localizing Multilingual Application Systems	2-5
Compilers That Support the MLS Environment	2-5
Programming Tools and Utilities Used in the MLS Environment	2-6

Contents

Section 3. Understanding Templates in Convention Definitions

Parts of a Template.....	3-2
Control Characters.....	3-2
Literal Text.....	3-2
Null Templates.....	3-2
Understanding Date and Time Templates.....	3-3
Processing Date and Time Templates.....	3-3
Control Characters for Date and Time Templates.....	3-3
Examples of Date and Time Templates.....	3-8
Summary of Date and Time Template Control Characters.....	3-9
Summary of Date Elements.....	3-13
Summary of Time Elements.....	3-15
Understanding Monetary and Numeric Templates.....	3-16
Processing Numeric and Monetary Templates.....	3-16
Control Characters for Numeric and Monetary Templates.....	3-18
Examples of Numeric and Monetary Templates.....	3-22
Summary of Monetary and Numeric Template Control Characters.....	3-22
Translating Literal Text Used in Conventions.....	3-24

Section 4. Establishing an MLS Environment

Installing the MLS Environment on Your System.....	4-1
Installing the ASERIESNATIVE Environment.....	4-2
Installing the SYSTEM/CONVENTIONS File for the First Time.....	4-3
Installing Updates to the SYSTEM/CONVENTIONS File.....	4-3
Conventions File Update Report.....	4-5
Initializing the CENTRALSUPPORT Library.....	4-6
Localizing the Operator Display Terminal (ODT) for Use with a Personal Computer Environment.....	4-7
Setting the Ccsversions, Languages, Conventions, and HostCCS at the System Level.....	4-9
Understanding the Impacts of Setting the MLS Options.....	4-9
Using the SYSTEMOPTIONS Command to Set the System Defaults.....	4-10
Using MARC to Display and Set the System Defaults.....	4-14
Viewing and Customizing Conventions at the System Level with MARC.....	4-17
Displaying the CONVDF—View/Add/Modify/Delete a Convention Definition Screen.....	4-18
Viewing Convention Definitions at the System Level.....	4-18
Adding Custom Convention Definitions.....	4-19
Modifying Custom Convention Definitions.....	4-19
Deleting Custom Convention Definitions.....	4-20

Setting a Default Language and Convention at the Usercode Level	4-21
Viewing and Setting the Language and Convention at the Session Level.....	4-21
Viewing and Setting a Natural Language.....	4-22
Viewing Convention Definitions at the MARC Session Level	4-25
Setting Conventions at the Session Level.....	4-25
Setting the Language and Convention at the Task Level	4-28
Using Ccsversions, Languages, or Conventions at the Program Level	4-28

Section 5. Localizing Products in an MLS Environment

Localizing Your Products.....	5-2
Translating Screens for Your Products	5-2
Translating Messages for Your Products	5-3
Translating Help Text for Your Products.....	5-3
Localizing Screens	5-4
Localizing Screens with IMG	5-4
Localizing SDF Plus Screens	5-5
Establishing Languages for SDF Plus	5-6
Localizing Help Text.....	5-8
Translating a Help Text Symbol File	5-8
Translating Commands That Identify the Help Book.....	5-9
Translating Commands That Identify the Help Text.....	5-11
Translating Text and Embedded Keywords.....	5-14
Localizing Unisys Products	5-15
Translating Messages for Unisys Products	5-22
Translating Help Text for Unisys Products	5-23
Identifying Unisys Products Localized with SDF Plus	5-25
Localizing Workstation Products	5-27
Making Translated Files Available.....	5-28

Section 6. Creating Application Systems in an MLS Environment

Designing Application Programs to Use the Internationalization Features.....	6-1
Providing Support for Coded Character Sets and Ccsversions	6-2
Providing Support for Natural Languages.....	6-16
Providing Support for Business and Cultural Conventions	6-20
Designing User Interfaces That Can Be Localized.....	6-32
Providing Support for Coded Character Sets.....	6-32
Providing Support for Ccsversions	6-33
Providing Support for Natural Languages.....	6-33
Providing Support for Business and Cultural Conventions	6-33

Section 7. Configuring Unisys Products for Internationalization

Using the Translation Table and Code Extension Features	7-2
Coded Character Set Mapping with FTP	7-3
Inbound File Transfers	7-3
Outbound File Transfers	7-3
Configuring HLCN for Internationalization	7-4
Using Code Extension	7-4
What Is Code Extension?	7-4
Who Can Use Code Extension?	7-5
What Are the Code Extension Special Characters?	7-5
Altering the Adapter Control to Recognize New Control Characters	7-6
How Does the Code Extension Process Work?	7-7
Using Coded Character Set Mapping Tables	7-12
Examples	7-17
Requesting Code Extension	7-21
Setting Code Extension	7-21
Transporting Kanji and Other Double-Octet Characters	7-22
Configuring Client Access Services for Internationalization	7-22
Which Coded Character Sets Are Used?	7-22
Which Files Are Translated?	7-23
How Are Translations Performed?	7-24

Section 8. Using C Programs in an MLS Environment

Understanding Locales	8-1
Understanding Locale Categories	8-2
Understanding the Predefined Locales	8-4
Creating or Modifying Locales	8-5
Understanding the LOCALES Array	8-5
Using MSGTRANS to Create or Modify a Locale	8-8
Testing Locales	8-14
Making the New or Changed Locales Available	8-15
Effects of Locale Components on C Library Functions	8-16
Character Handling <ctype.h>	8-17
Localization <locale.h>	8-18
Input/Output <stdio.h> and General Utilities <stdlib.h>	8-24
String Handling <string.h>	8-24
Date and Time <time.h>	8-25

Section 9. Coded Character Sets and Coded Character Set Mapping Tables

Identifying Coded Character Sets on the Release Media 9-2
 Coded Character Set Mapping Tables 9-5
 Decimal and Hexadecimal Numbers 9-15
 Obtaining Decimal Numbers for 8-Bit Characters 9-15
 Obtaining Hexadecimal Numbers for 8-Bit Characters 9-15

Section 10. Eight-Bit Coded Character Sets: Arabic20EBCDIC to CANSupplISO

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets 10-2
 ASCII Coded Character Set (8CS1.00) 10-12
 ASERIESEBCDIC Coded Character Set (8CS1.01) 10-17
 CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets 10-26

Section 11. Eight-Bit Coded Character Sets: CODEPAGE437 to CODEPAGE1256

CODEPAGE437 Coded Character Set (8CS49.00) 11-2
 CODEPAGE850 Coded Character Set (8CS49.00) 11-11
 CODEPAGE851 Coded Character Set (8CS43.00) 11-20
 CODEPAGE852 Coded Character Set (8CS27.03) 11-29
 CODEPAGE857 Coded Character Set (8CS00.00) 11-38
 CODEPAGE864 Coded Character Set (8CS00.00) 11-47
 CODEPAGE866 Coded Character Set (8CS56.03) 11-56
 CODEPAGE1250 Coded Character Set (8CS58.03) 11-65
 CODEPAGE1251 Coded Character Set (8CS57.03) 11-74
 CODEPAGE1252 Coded Character Set (8CS60.03) 11-83
 CODEPAGE1253 Coded Character Set (8CS00.00) 11-92
 CODEPAGE1254 Coded Character Set (8CS00.00) 11-101
 CODEPAGE1256 Coded Character Set (8CS00.00) 11-110

Section 12. Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

FrenchArabicEBCDIC (8CS46.01) and FrenchArabicISO (8CS46.00) Coded Character Sets 12-2
 HungarianBTOS Coded Character Sets (8CS00.00) 12-12
 IBM297 Coded Character Set (8CS62.01) 12-21
 IBMSwedenEBCDIC Coded Character Set (8CS00.00) 12-30
 JapanEBCDIC Coded Character Set (8CS00.00) 12-39
 JapanV24 Coded Character Set (8CS00.00) 12-48
 JISASCII Coded Character Set (8CS00.00) 12-57
 Latin1EBCDIC (8CS26.01) and Latin1ISO (8CS26.01) Coded Character Sets 12-66

Contents

Latin2EBCDIC (8CS27.01) and Latin2ISO (8CS27.00) Coded Character Sets	12-76
Latin5EBCDIC (8CS29.01) and Latin5ISO (8CS29.00) Coded Character Sets	12-86
Latin9EBCDIC (8CS00.00) and Latin9ISO (8CS00.00) Coded Character Sets	12-96
LatinCyrillicEBCDIC (8CS55.01) and LatinCyrillicISO (8CS55.00) Coded Character Sets	12-106
LatinGreekBTOS Coded Character Set (8CS41.00)	12-116
LatinGreekEBCDIC (8CS30.01) and LatinGreekISO (8CS30.00) Coded Character Sets	12-125
LocalEBCDIC Coded Character Set (8500.00)	12-135
MACROMAN Coded Character Set (8CS61.03)	12-144
NorwayBTOS Coded Character Set (8CS50.00)	12-153

Section 13. Double-Byte and Mixed, Multibyte Coded Character Sets

ASTASCIIGB2312 Coded Character Set (816CS00.00)	13-5
ASTASCIIKSC5601 Coded Character Set (816CS00.00)	13-6
ASTASCIUUTL Coded Character Set (816CS00.00)	13-7
ASTUTL Coded Character Set (16CS00.00)	13-8
ASUTL Coded Character Set (16CS00.00)	13-9
CODEPAGE932 Coded Character Set (816CS00.00)	13-10
CODEPAGE936 Coded Character Set (816CS00.00)	13-11
CODEPAGE949 Coded Character Set (816CS00.00)	13-12
CODEPAGE950 Coded Character Set (816CS00.00)	13-13
EBCDICGB2312 Coded Character Set (816CS00.00)	13-14
EBCDICKSC5601 Coded Character Set (816CS00.00)	13-15
EBCDICUTL Coded Character Set (816CS00.00)	13-16
EUCJp Coded Character Set (816CS00.00)	13-17
JapanEBCDICJBIS8 Coded Character Set (816CS00.00)	13-19
JapanV24JBIS8 Coded Character Set (816CS00.00)	13-21
JBIS7 Coded Character Set (16CS00.00)	13-23
JBIS8 Coded Character Set (16CS00.00)	13-24
JISASCIJBIS7 Coded Character Set (816CS00.00)	13-25
LetsJ Coded Character Set (816CS00.00)	13-27
UCS2 Coded Character Set (16CS00.00)	13-29
UCS2NT Coded Character Set (16CS00.00)	13-30

Section 14. Ccsversions

Arabic20 Ccsversion Definitions	14-3
ASeriesNative Ccsversion Definitions	14-5
Brazilian Ccsversion Definitions	14-6
CanadaEBCDIC Ccsversion Definitions	14-9
CanadaGP Ccsversion Definitions	14-12
Czechoslovakia Ccsversion Definitions	14-16
France Ccsversion Definitions	14-19
Hungarian Ccsversion Definitions	14-22
LatinGreek Ccsversion Definitions	14-25

Norway Ccsversion Definitions	14–30
Polish Ccsversion Definitions.....	14–33
Romanian Ccsversion Definitions	14–36
Russian Ccsversion Definitions.....	14–39
Spanish Ccsversion Definitions.....	14–43
Swedish1 Ccsversion Definitions	14–46
Swiss Ccsversion Definitions.....	14–49
Turkish Ccsversion Definitions.....	14–53

Section 15. Standard Conventions

Argentina Convention Definition	15–2
ASeriesNative Convention Definition.....	15–3
Australia Convention Definition.....	15–4
Belgium Convention Definition	15–5
Brazil Convention Definition.....	15–6
Chile Convention Definition	15–7
China Convention Definition	15–8
Colombia Convention Definition	15–9
CostaRica Convention Definition	15–10
Croatia Convention Definition	15–11
Cyrillic Convention Definition	15–12
Czechoslovakia Convention Definition	15–13
Denmark Convention Definition.....	15–14
Egypt Convention Definition	15–15
EnglishCanada Convention Definition.....	15–16
EuropeanStandard Convention Definition.....	15–17
FranceBureautique Convention Definition	15–18
FranceListing Convention Definition	15–19
FrenchCanada Convention Definition	15–20
Greece Convention Definition.....	15–21
HongKong Convention Definition	15–22
Hungary Convention Definition	15–23
Italy Convention Definition.....	15–24
Japan1 Convention Definition.....	15–25
Japan2 Convention Definition.....	15–26
Jordan Convention Definition.....	15–27
Kenya Convention Definition.....	15–28
Kuwait Convention Definition	15–29
Lebanon Convention Definition.....	15–30
Malaysia Convention Definition	15–31
Mexico Convention Definition.....	15–32
Netherlands Convention Definition	15–33
NewZealand Convention Definition	15–34
Nigeria Convention Definition	15–35
Norway Convention Definition.....	15–36
Peru Convention Definition.....	15–37
Philippines Convention Definition	15–38
Poland Convention Definition.....	15–39
Romania Convention Definition	15–40
SouthAfrica Convention Definition.....	15–41

Contents

Spain Convention Definition	15-42
StandardTurkey Convention Definition	15-43
Sweden Convention Definition	15-44
Switzerland Convention Definition	15-45
Taiwan Convention Definition	15-46
UAE Convention Definition	15-47
UnitedKingdom1 Convention Definition	15-48
UnitedKingdom2 Convention Definition	15-49
Venezuela Convention Definition	15-50
Yugoslavia Convention Definition	15-51
Zimbabwe Convention Definition	15-52

Section 16. The XlateEBCDIC Server

Major Functions	16-2
IXlateEBCDIC	16-2
IXlateEBC	16-3
IXlateEBCDICCIS	16-3
IEnumXlateCCSes	16-3
Mapping Object Initialization	16-3
The Best Fit Table	16-4
Error Reporting	16-5
User Interfaces	16-9
IXlateEBCDIC Interface	16-9
Interface IXlateEBC	16-14
Interface IXlateEBCDICCIS	16-17
Interface IEnumXlateCCSes	16-19
The XlateEBCDIC Type Library	16-21

Appendix A. CENTRALSUPPORT Library Messages

Result Messages	A-1
Header Messages	A-2
Result Messages by Category	A-4
CENTRALSUPPORT Initialization Messages	A-23

Appendix B. Examples of Using Libraries to Access Output Message Arrays

Example ALGOL Library	B-1
Example C Include File	B-3
Example C Program	B-4
Example COBOL74 Program	B-5
Example COBOL85 Program	B-6
Example Pascal Program	B-8
Example RPG Program	B-9

Appendix C. Using the Complex Procedure

Specification.....	C-2
Zero-Bounded Array version	C-2
Star-Bounded Array Version	C-3
Pointer Version	C-3
Parameters.....	C-4
Interaction of the State and Option Parameters.....	C-7
Implementation Strategies.....	C-8
Discrete Mode	C-8
Continuous Stream Mode	C-8
Reducing Initialization Overhead	C-9
Terminating Calls with Single Bracket Invocation	
Characters	C-10
Example Mappings	C-10

Appendix D. Help Text Available for Localization

Appendix E. Messages Available for Localization

Appendix F. Understanding Railroad Diagrams

Railroad Diagram Concepts.....	F-1
Paths	F-1
Constants and Variables	F-2
Constraints.....	F-3
Following the Paths of a Railroad Diagram	F-6
Railroad Diagram Examples with Sample Input.....	F-7

Appendix G. Related Product Information

Index	1
--------------------	----------

Contents

Figures

1-1.	Coded Character Sets and Ccsversions.....	1-6
6-1.	Logical Comparison Method.....	6-15
6-2.	Sample Output Message Array.....	6-18
7-1.	Example Hardware Configuration for Code Extension.....	7-7
7-2.	Transmitting Data: The Process at the Terminal.....	7-8
7-3.	Receiving Data: The Process at the EDCDLP.....	7-9
7-4.	Transmitting Data: The Process at the EDCDLP.....	7-10
7-5.	Receiving Data: The Process at the Terminal.....	7-11
7-6.	ASCII Mapping Table.....	7-13
7-7.	Mapping Process for Multiple Coded Character Sets.....	7-14
7-8.	Declaring Enumerations.....	7-17
7-9.	Creating Coded Character Set Mapping Tables.....	7-19
7-10.	Recognizing Coded Character Set Mapping Tables.....	7-20
8-1.	MSGTRANS Start Session Screen.....	8-8
8-2.	MSGTRANS Home Screen.....	8-9
8-3.	MSGTRANS Edit Screen—Displaying the First Physical Message.....	8-10
8-4.	MSGTRANS Edit Screen—Displaying Message 0 of the LOCALES Array.....	8-11
8-5.	MSGTRANS Save Screen.....	8-12
8-6.	MSGTRANS Restart Screen.....	8-13
10-1.	Arabic20EBCDIC Code Table (8CS11.01).....	10-2
10-2.	Arabic20ISO Code Table (8CS11.00).....	10-3
10-3.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 0 and 1.....	10-4
10-4.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 2 and 3.....	10-5
10-5.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 4 and 5.....	10-6
10-6.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 6 and 7.....	10-7
10-7.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 8 and 9.....	10-8
10-8.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 10 and 11.....	10-9
10-9.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 12 and 13.....	10-10
10-10.	Arabic20EBCDIC and Arabic20ISO Column Listing Columns 14 and 15.....	10-11
10-11.	ASCII Code Table (8CS1.00).....	10-12
10-12.	ASCII Column Listing Columns 0 and 1.....	10-13
10-13.	ASCII Column Listing Columns 2 and 3.....	10-14
10-14.	ASCII Column Listing Columns 4 and 5.....	10-15
10-15.	ASCII Column Listing Columns 6 and 7.....	10-16
10-16.	ASERIESEBCDIC Code Table (8CS1.01).....	10-17
10-17.	ASERIESEBCDIC Column Listing Columns 0 and 1.....	10-18
10-18.	ASERIESEBCDIC Column Listing Columns 2 and 3.....	10-19
10-19.	ASERIESEBCDIC Column Listing Columns 4 and 5.....	10-20
10-20.	ASERIESEBCDIC Column Listing Columns 6 and 7.....	10-21

Figures

10-21.	ASERIESEBCDIC Column Listing Columns 8 and 9.....	10-22
10-22.	ASERIESEBCDIC Column Listing Columns 10 and 11.....	10-23
10-23.	ASERIESEBCDIC Column Listing Columns 12 and 13.....	10-24
10-24.	ASERIESEBCDIC Column Listing Columns 14 and 15.....	10-25
10-25.	CANSupplEBCDIC Code Table (8CS51.01).....	10-26
10-26.	CANSupplISO Code Table (8CS51.00).....	10-27
10-27.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 0 and 1.....	10-28
10-28.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 2 and 3.....	10-29
10-29.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 4 and 5.....	10-30
10-30.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 6 and 7.....	10-31
10-31.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 8 and 9.....	10-32
10-32.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 10 and 11.....	10-33
10-33.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 12 and 13.....	10-34
10-34.	CANSupplEBCDIC and CANSupplISO Column Listing Columns 14 and 15.....	10-35
11-1.	CODEPAGE437 Code Table (8CS49.00).....	11-2
11-2.	CODEPAGE437 Column Listing Columns 0 and 1.....	11-3
11-3.	CODEPAGE437 Column Listing Columns 2 and 3.....	11-4
11-4.	CODEPAGE437 Column Listing Columns 4 and 5.....	11-5
11-5.	CODEPAGE437 Column Listing Columns 6 and 7.....	11-6
11-6.	CODEPAGE437 Column Listing Columns 8 and 9.....	11-7
11-7.	CODEPAGE437 Column Listing Columns 10 and 11.....	11-8
11-8.	CODEPAGE437 Column Listing Columns 12 and 13.....	11-9
11-9.	CODEPAGE437 Column Listing Columns 14 and 15.....	11-10
11-10.	CODEPAGE850 Code Table (8CS49.00).....	11-11
11-11.	CODEPAGE850 Column Listing Columns 0 and 1.....	11-12
11-12.	CODEPAGE850 Column Listing Columns 2 and 3.....	11-13
11-13.	CODEPAGE850 Column Listing Columns 4 and 5.....	11-14
11-14.	CODEPAGE850 Column Listing Columns 6 and 7.....	11-15
11-15.	CODEPAGE850 Column Listing Columns 8 and 9.....	11-16
11-16.	CODEPAGE850 Column Listing Columns 10 and 11.....	11-17
11-17.	CODEPAGE850 Column Listing Columns 12 and 13.....	11-18
11-18.	CODEPAGE850 Column Listing Columns 14 and 15.....	11-19
11-19.	CODEPAGE851 Code Table (8CS43.00).....	11-20
11-20.	CODEPAGE851 Column Listing Columns 0 and 1.....	11-21
11-21.	CODEPAGE851 Column Listing Columns 2 and 3.....	11-22
11-22.	CODEPAGE851 Column Listing Columns 4 and 5.....	11-23
11-23.	CODEPAGE851 Column Listing Columns 6 and 7.....	11-24
11-24.	CODEPAGE851 Column Listing Columns 8 and 9.....	11-25
11-25.	CODEPAGE851 Column Listing Columns 10 and 11.....	11-26
11-26.	CODEPAGE851 Column Listing Columns 12 and 13.....	11-27
11-27.	CODEPAGE851 Column Listing Columns 14 and 15.....	11-28
11-28.	CODEPAGE852 Code Table (8CS27.03).....	11-29
11-29.	CODEPAGE852 Column Listing Columns 0 and 1.....	11-30
11-30.	CODEPAGE852 Column Listing Columns 2 and 3.....	11-31
11-31.	CODEPAGE852 Column Listing Columns 4 and 5.....	11-32
11-32.	CODEPAGE852 Column Listing Columns 6 and 7.....	11-33
11-33.	CODEPAGE852 Column Listing Columns 8 and 9.....	11-34
11-34.	CODEPAGE852 Column Listing Columns 10 and 11.....	11-35

11-35.	CODEPAGE852 Column Listing Columns 12 and 13	11-36
11-36.	CODEPAGE852 Column Listing Columns 14 and 15	11-37
11-37.	CODEPAGE857 Code Table (8CS00.00)	11-38
11-38.	CODEPAGE857 Column Listing Columns 0 and 1	11-39
11-39.	CODEPAGE857 Column Listing Column 2 and 3	11-40
11-40.	CODEPAGE857 Column Listing Columns 4 and 5	11-41
11-41.	CODEPAGE857 Column Listing Columns 6 and 7	11-42
11-42.	CODEPAGE857 Column Listing Columns 8 and 9	11-43
11-43.	CODEPAGE857 Column Listing Columns 10 and 11	11-44
11-44.	CODEPAGE857 Column Listing Columns 12 and 13	11-45
11-45.	CODEPAGE857 Column Listing Columns 14 and 15	11-46
11-46.	CODEPAGE864 Code Table (8CS00.00)	11-47
11-47.	CODEPAGE864 Column Listing Columns 0 and 1	11-48
11-48.	CODEPAGE864 Column Listing Columns 2 and 3	11-49
11-49.	CODEPAGE864 Column Listing Columns 4 and 5	11-50
11-50.	CODEPAGE864 Column Listing Columns 6 and 7	11-51
11-51.	CODEPAGE864 Column Listing Columns 8 and 9	11-52
11-52.	CODEPAGE864 Column Listing Columns 10 and 11	11-53
11-53.	CODEPAGE864 Column Listing Columns 12 and 13	11-54
11-54.	CODEPAGE864 Column Listing Columns 14 and 15	11-55
11-55.	CODEPAGE866 Code Table (8CS66.03)	11-56
11-56.	CODEPAGE866 Column Listing Columns 0 and 1	11-57
11-57.	CODEPAGE866 Column Listing Columns 2 and 3	11-58
11-58.	CODEPAGE866 Column Listing Columns 4 and 5	11-59
11-59.	CODEPAGE866 Column Listing Columns 6 and 7	11-60
11-60.	CODEPAGE866 Column Listing Columns 8 and 9	11-61
11-61.	CODEPAGE866 Column Listing Columns 10 and 11	11-62
11-62.	CODEPAGE866 Column Listing Columns 12 and 13	11-63
11-63.	CODEPAGE866 Column Listing Columns 14 and 15	11-64
11-64.	CODEPAGE1250 Code Table (8CS58.03)	11-65
11-65.	CODEPAGE1250 Column Listing Columns 0 and 1	11-66
11-66.	CODEPAGE1250 Column Listing Columns 2 and 3	11-67
11-67.	CODEPAGE1250 Column Listing Columns 4 and 5	11-68
11-68.	CODEPAGE1250 Column Listing Columns 6 and 7	11-69
11-69.	CODEPAGE1250 Column Listing Columns 8 and 9	11-70
11-70.	CODEPAGE1250 Column Listing Columns 10 and 11	11-71
11-71.	CODEPAGE1250 Column Listing Columns 12 and 13	11-72
11-72.	CODEPAGE1250 Column Listing Columns 14 and 15	11-73
11-73.	CODEPAGE1251 Code Table (8CS57.03)	11-74
11-74.	CODEPAGE1251 Column Listing Columns 0 and 1	11-75
11-75.	CODEPAGE1251 Column Listing Columns 2 and 3	11-76
11-76.	CODEPAGE1251 Column Listing Columns 4 and 5	11-77
11-77.	CODEPAGE1251 Column Listing Columns 6 and 7	11-78
11-78.	CODEPAGE1251 Column Listing Columns 8 and 9	11-79
11-79.	CODEPAGE1251 Column Listing Columns 10 and 11	11-80
11-80.	CODEPAGE1251 Column Listing Columns 12 and 13	11-81
11-81.	CODEPAGE1251 Column Listing Columns 14 and 15	11-82
11-82.	CODEPAGE1252 Code Table (8CS60.03)	11-83
11-83.	CODEPAGE1252 Column Listing Columns 0 and 1	11-84
11-84.	CODEPAGE1252 Column Listing Columns 2 and 3	11-85
11-85.	CODEPAGE1252 Column Listing Columns 4 and 5	11-86
11-86.	CODEPAGE1252 Column Listing Columns 6 and 7	11-87

Figures

11-87.	CODEPAGE1252 Column Listing Columns 8 and 9	11-88
11-88.	CODEPAGE1252 Column Listing Columns 10 and 11	11-89
11-89.	CODEPAGE1252 Column Listing Columns 12 and 13	11-90
11-90.	CODEPAGE1252 Column Listing Columns 14 and 15	11-91
11-91.	CODEPAGE1253 Code Table (8CS00.00).....	11-92
11-92.	CODEPAGE1253 Column Listing Columns 0 and 1	11-93
11-93.	CODEPAGE1253 Column Listing Columns 2 and 3	11-94
11-94.	CODEPAGE1253 Column Listing Columns 4 and 5	11-95
11-95.	CODEPAGE1253 Column Listing Columns 6 and 7	11-96
11-96.	CODEPAGE1253 Column Listing Columns 8 and 9	11-97
11-97.	CODEPAGE1253 Column Listing Columns 10 and 11	11-98
11-98.	CODEPAGE1253 Column Listing Columns 12 and 13	11-99
11-99.	CODEPAGE1253 Column Listing Columns 14 and 15	11-100
11-100.	CODEPAGE1254 Code Table (8CS00.00).....	11-101
11-101.	CODEPAGE1254 Column Listing Columns 0 and 1	11-102
11-102.	CODEPAGE1254 Column Listing Columns 2 and 3	11-103
11-103.	CODEPAGE1254 Column Listing Columns 4 and 5	11-104
11-104.	CODEPAGE1254 Column Listing Columns 6 and 7	11-105
11-105.	CODEPAGE1254 Column Listing Columns 8 and 9	11-106
11-106.	CODEPAGE1254 Column Listing Columns 10 and 11	11-107
11-107.	CODEPAGE1254 Column Listing Columns 12 and 13	11-108
11-108.	CODEPAGE1254 Column Listing Columns 14 and 15	11-109
11-109.	CODEPAGE1256 Code Table (8CS00.00) CODEPAGE1256 Coded Character Set (8CS00.00)	11-110
11-110.	CODEPAGE1256 Column Listing Columns 0 and 1	11-111
11-111.	CODEPAGE1256 Column Listing Columns 2 and 3	11-112
11-112.	CODEPAGE1256 Column Listing Columns 4 and 5	11-113
11-113.	CODEPAGE1256 Column Listing Columns 6 and 7	11-114
11-114.	CODEPAGE1256 Column Listing Columns 8 and 9	11-115
11-115.	CODEPAGE1256 Column Listing Columns 10 and 11	11-116
11-116.	CODEPAGE1256 Column Listing Columns 11 and 12	11-117
11-117.	CODEPAGE1256 Column Listing Columns 13 and 14	11-118
12-1.	FrenchArabicEBCDIC Code Table (8CS46.01)	12-2
12-2.	FrenchArabicISO Code Table (8CS46.00)	12-3
12-3.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 0 and 1	12-4
12-4.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 2 and 3	12-5
12-5.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 4 and 5	12-6
12-6.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 6 and 7	12-7
12-7.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 8 and 9	12-8
12-8.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 10 and 11	12-9
12-9.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 12 and 13	12-10
12-10.	FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 14 and 15	12-11
12-11.	HungarianBTOS Code Table (8CS00.00)	12-12

12-12.	HungarianBTOS Column Listing Columns 0 and 1	12-13
12-13.	HungarianBTOS Column Listing Columns 2 and 3	12-14
12-14.	HungarianBTOS Column Listing Columns 4 and 5	12-15
12-15.	HungarianBTOS Column Listing Columns 6 and 7	12-16
12-16.	HungarianBTOS Column Listing Columns 8 and 9	12-17
12-17.	HungarianBTOS Column Listing Columns 10 and 11	12-18
12-18.	HungarianBTOS Column Listing Columns 12 and 13	12-19
12-19.	HungarianBTOS Column Listing Columns 14 and 15	12-20
12-20.	IBM297 Code Table (8CS62.01).....	12-21
12-21.	IBM297 Column Listing Columns 0 and 1	12-22
12-22.	IBM297 Column Listing Columns 2 and 3	12-23
12-23.	IBM297 Column Listing Columns 4 and 5	12-24
12-24.	IBM297 Column Listing Columns 6 and 7	12-25
12-25.	IBM297 Column Listing Columns 8 and 9	12-26
12-26.	IBM297 Column Listing Columns 10 and 11	12-27
12-27.	IBM297 Column Listing Columns 12 and 13	12-28
12-28.	IBM297 Column Listing Columns 14 and 15	12-29
12-29.	IBMSwedenEBCDIC Code Table	12-30
12-30.	IBMSwedenEBCDIC Column Listing Columns 0 and 1	12-31
12-31.	IBMSwedenEBCDIC Column Listing Columns 2 and 3	12-32
12-32.	IBMSwedenEBCDIC Column Listing Columns 4 and 5	12-33
12-33.	IBMSwedenEBCDIC Column Listing Columns 6 and 7	12-34
12-34.	IBMSwedenEBCDIC Column Listing Columns 8 and 9	12-35
12-35.	IBMSwedenEBCDIC Column Listing Columns 10 and 11	12-36
12-36.	IBMSwedenEBCDIC Column Listing Columns 12 and 13	12-37
12-37.	IBMSwedenEBCDIC Column Listing Columns 14 and 15	12-38
12-38.	JapanEBCDIC Code Table (8CS00.00).....	12-39
12-39.	JapanEBCDIC Column Listing Columns 0 and 1	12-40
12-40.	JapanEBCDIC Column Listing Columns 2 and 3	12-41
12-41.	JapanEBCDIC Column Listing Columns 4 and 5	12-42
12-42.	JapanEBCDIC Column Listing Columns 6 and 7	12-43
12-43.	JapanEBCDIC Column Listing Columns 8 and 9	12-44
12-44.	JapanEBCDIC Column Listing Columns 10 and 11	12-45
12-45.	JapanEBCDIC Column Listing Columns 12 and 13	12-46
12-46.	JapanEBCDIC Column Listing Columns 14 and 15	12-47
12-47.	JapanV24 Code Table (8CS00.00).....	12-48
12-48.	JapanV24 Column Listing Columns 0 and 1	12-49
12-49.	JapanV24 Column Listing Columns 2 and 3	12-50
12-50.	JapanV24 Column Listing Columns 4 and 5	12-51
12-51.	JapanV24 Column Listing Columns 6 and 7	12-52
12-52.	JapanV24 Column Listing Columns 8 and 9	12-53
12-53.	JapanV24 Column Listing Columns 10 and 11	12-54
12-54.	JapanV24 Column Listing Columns 12 and 13	12-55
12-55.	JapanV24 Column Listing Columns 14 and 15	12-56
12-56.	JISASCII Code Table (8CS00.00)	12-57
12-57.	JISASCII Column Listings Columns 0 and 1	12-58
12-58.	JISASCII Column Listing Columns 2 and 3	12-59
12-59.	JISASCII Column Listing Columns 4 and 5	12-60
12-60.	JISASCII Column Listing Columns 6 and 7	12-61
12-61.	JISASCII Column Listing Columns 8 and 9	12-62
12-62.	JISASCII Column Listing Columns 10 and 11	12-63
12-63.	JISASCII Column Listing Columns 12 and 13	12-64

Figures

12-64.	JISASCII Column Listing Columns 14 and 15.....	12-65
12-65.	Latin1EBCDIC Code Table (8CS26.01)	12-66
12-66.	Latin1ISO Code Table (8CS26.00)	12-67
12-67.	Latin1EBCDIC and Latin1ISO Column Listing Columns 0 and 1	12-68
12-68.	Latin1EBCDIC and Latin1ISO Column Listing Columns 2 and 3	12-69
12-69.	Latin1EBCDIC and Latin1ISO Column Listing Columns 4 and 5	12-70
12-70.	Latin1EBCDIC and Latin1ISO Column Listing Columns 6 and 7	12-71
12-71.	Latin1EBCDIC and Latin1ISO Column Listing Columns 8 and 9	12-72
12-72.	Latin1EBCDIC and Latin1ISO Column Listing Columns 10 and 11	12-73
12-73.	Latin1EBCDIC and Latin1ISO Column Listing Columns 12 and 13	12-74
12-74.	Latin1EBCDIC and Latin1ISO Column Listing Columns 14 and 15	12-75
12-75.	Latin2EBCDIC Code Table (8CS27.01)	12-76
12-76.	Latin2ISO Code Table (8CS27.00)	12-77
12-77.	Latin2EBCDIC and Latin2ISO Column Listing Columns 0 and 1	12-78
12-78.	Latin2EBCDIC and Latin2ISO Column Listing Columns 2 and 3	12-79
12-79.	Latin2EBCDIC and Latin2ISO Column Listing Columns 4 and 5	12-80
12-80.	Latin2EBCDIC and Latin2ISO Column Listing Columns 6 and 7	12-81
12-81.	Latin2EBCDIC and Latin2ISO Column Listing Columns 8 and 9	12-82
12-82.	Latin2EBCDIC and Latin2ISO Column Listing Columns 10 and 11	12-83
12-83.	Latin2EBCDIC and Latin2ISO Column Listing Columns 12 and 13	12-84
12-84.	Latin2EBCDIC and Latin2ISO Column Listing Columns 14 and 15	12-85
12-85.	Latin5EBCDIC Code Table (8CS29.01)	12-86
12-86.	Latin5ISO Code Table (8CS29.00)	12-87
12-87.	Latin5EBCDIC and Latin5ISO Column Listing Columns 0 and 1	12-88
12-88.	Latin5EBCDIC and Latin5ISO Column Listing Columns 2 and 3	12-89
12-89.	Latin5EBCDIC and Latin5ISO Column Listing Columns 4 and 5	12-90
12-90.	Latin5EBCDIC and Latin5ISO Column Listing Columns 6 and 7	12-91
12-91.	Latin5EBCDIC and Latin5ISO Column Listing Columns 8 and 9	12-92
12-92.	Latin5EBCDIC and Latin5ISO Column Listing Columns 10 and 11	12-93
12-93.	Latin5EBCDIC and Latin5ISO Column Listing Columns 12 and 13	12-94
12-94.	Latin5EBCDIC and Latin5ISO Column Listing Columns 14 and 15	12-95
12-95.	Latin9EBCDIC Code Table (8CS00.00)	12-96
12-96.	Latin9ISO Code Table (8CS00.00)	12-97
12-97.	Latin9EBCDIC and Latin9ISO Column Listing Columns 0 and 1	12-98
12-98.	Latin9EBCDIC and Latin9ISO Column Listing Columns 2 and 3	12-99
12-99.	Latin9EBCDIC and Latin9ISO Column Listing Columns 4 and 5	12-100
12-100.	Latin9EBCDIC and Latin9ISO Column Listing Columns 6 and 7	12-101
12-101.	Latin9EBCDIC and Latin9ISO Column Listing Columns 8 and 9	12-102
12-102.	Latin9EBCDIC and Latin9ISO Column Listing Columns 10 and 11	12-103
12-103.	Latin9EBCDIC and Latin9ISO Column Listing Columns 12 and 13	12-104
12-104.	Latin9EBCDIC and Latin9ISO Column Listing Columns 14 and 15	12-105
12-105.	LatinCyrillicEBCDIC Code Table (8CS55.01)	12-106
12-106.	LatinCyrillicISO Code Table (8CS55.00)	12-107
12-107.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 0 and 1	12-108
12-108.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 2 and 3	12-109
12-109.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 4 and 5	12-110
12-110.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 6 and 7	12-111
12-111.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 8 and 9	12-112
12-112.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 10 and 11	12-113
12-113.	LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 12 and 13	12-114

12-114. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 14 and 15	12-115
12-115. LatinGreekBTOS Code Table (8CS41.00)	12-116
12-116. LatinGreekBTOS Column Listing Columns 0 and 1	12-117
12-117. LatinGreekBTOS Column Listing Columns 2 and 3	12-118
12-118. LatinGreekBTOS Column Listing Columns 4 and 5	12-119
12-119. LatinGreekBTOS Column Listing Columns 6 and 7	12-120
12-120. LatinGreekBTOS Column Listing Columns 8 and 9	12-121
12-121. LatinGreekBTOS Column Listing Columns 10 and 11	12-122
12-122. LatinGreekBTOS Column Listing Columns 12 and 13	12-123
12-123. LatinGreekBTOS Column Listing Columns 14 and 15	12-124
12-124. LatinGreekEBCDIC Code Table (8CS30.01)	12-125
12-125. LatinGreekISO Code Table (8CS30.00)	12-126
12-126. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 0 and 1	12-127
12-127. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 2 and 3	12-128
12-128. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 4 and 5	12-129
12-129. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 6 and 7	12-130
12-130. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 8 and 9	12-131
12-131. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 10 and 11	12-132
12-132. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 12 and 13	12-133
12-133. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 14 and 15	12-134
12-134. LocalEBCDIC Code Table (8500.00)	12-135
12-135. LocalEBCDIC Column Listing Columns 0 and 1	12-136
12-136. LocalEBCDIC Column Listing Columns 2 and 3	12-137
12-137. LocalEBCDIC Column Listing Columns 4 and 5	12-138
12-138. LocalEBCDIC Column Listing Columns 6 and 7	12-139
12-139. LocalEBCDIC Column Listing Columns 8 and 9	12-140
12-140. LocalEBCDIC Column Listing Columns 10 and 11	12-141
12-141. LocalEBCDIC Column Listing Columns 12 and 13	12-142
12-142. LocalEBCDIC Column Listing Columns 14 and 15	12-143
12-143. MACROMAN Code Table (8CS61.03)	12-144
12-144. MACROMAN Column Listing Columns 0 and 1	12-145
12-145. MACROMAN Column Listing Columns 2 and 3	12-146
12-146. MACROMAN Column Listing Columns 4 and 5	12-147
12-147. MACROMAN Column Listing Columns 6 and 7	12-148
12-148. MACROMAN Column Listing Columns 8 and 9	12-149
12-149. MACROMAN Column Listing Columns 10 and 11	12-150
12-150. MACROMAN Column Listing Columns 12 and 13	12-151
12-151. MACROMAN Column Listing Columns 14 and 15	12-152
12-152. NorwayBTOS Code Table (8CS50.00)	12-153
12-153. NorwayBTOS Column Listing Columns 0 and 1	12-154
12-154. NorwayBTOS Column Listing Columns 2 and 3	12-155
12-155. NorwayBTOS Column Listing Columns 4 and 5	12-156
12-156. NorwayBTOS Column Listing Columns 6 and 7	12-157

Figures

12-157. NorwayBTOS Column Listing Columns 8 and 9.....	12-158
12-158. NorwayBTOS Column Listing Columns 10 and 11.....	12-159
12-159. NorwayBTOS Column Listing Columns 12 and 13.....	12-160
12-160. NorwayBTOS Column Listing Columns 14 and 15.....	12-161
13-1. Reading Mixed, Multibyte Coded Character Sets	13-4
13-2. ASTASCIIGB2312 Coded Character Set (816CS00.00)	13-5
13-3. ASTASCIIKSC5601 Coded Character Set (816CS00.00)	13-6
13-4. ASTASCIUUTL Coded Character Set (816CS00.00)	13-7
13-5. ASTUTL Coded Character Set (16CS00.00).....	13-8
13-6. ASUTL Coded Character Set (16CS00.00).....	13-9
13-7. CODEPAGE932 Coded Character Set (816CS00.00)	13-10
13-8. CODEPAGE936 Coded Character Set (816CS00.00)	13-11
13-9. CODEPAGE949 Coded Character Set (816CS00.00)	13-12
13-10. CODEPAGE950 Coded Character Set (816CS00.00)	13-13
13-11. EBCDICGB2312 Coded Character Set (816CS00.00).....	13-14
13-12. EBCDICKSC5601 Coded Character Set (816CS00.00).....	13-15
13-13. EBCDICUTL Coded Character Set (816CS00.00)	13-16
13-14. EUCJp Coded Character Set (816CS00.00).....	13-18
13-15. JapanEBCDICJBIS8 Coded Character Set (816CS00.00).....	13-20
13-16. JapanV24JBIS8 Coded Character Set (816CS00.00).....	13-22
13-17. JBIS7 Coded Character Set (16CS00.00)	13-23
13-18. JBIS8 Coded Character Set (16CS00.00)	13-24
13-19. JISASCIJBIS7 Coded Character Set (816CS00.00)	13-26
13-20. LetsJ Coded Character Set (816CS00.00)	13-28
13-21. UCS2 Coded Character Set (16CS00.00).....	13-29
13-22. UCS2NT Coded Character Set (16CS00.00).....	13-30

Tables

1-1.	Relationship of TLO and LAD.....	1-12
1-2.	Relationship of TLO and CAD.....	1-12
1-3.	Relationship of TLO, LAD, and CAD.....	1-12
1-4.	Relationship of TLO and CED.....	1-13
1-5.	Elements in Convention Definitions.....	1-16
2-1.	Access to Internationalization Features.....	2-4
2-2.	Impacts of System Levels for a Natural Language and Convention.....	2-5
3-1.	Alphabetic Control Characters for Date and Time Templates.....	3-4
3-2.	Numeric Control Characters for Date and Time Templates.....	3-5
3-3.	Date and Time Template Control Characters.....	3-10
3-4.	Date Elements.....	3-13
3-5.	Time Elements.....	3-15
3-6.	Control Characters for Numeric and Monetary Templates.....	3-18
3-7.	Monetary and Numeric Template Control Characters.....	3-23
4-1.	Communicating Between the Host and PC Environment.....	4-7
5-1.	Translation Tools for Your Products.....	5-2
5-2.	Required Help Book Commands.....	5-9
5-3.	Commands That Identify Help Text.....	5-11
5-4.	Keyword Special Characters.....	5-13
5-5.	Unisys Products That Can Be Localized.....	5-15
5-6.	Matching Help Utilities to Screen Interfaces.....	5-23
5-7.	Unisys Products Localized with SDF Plus.....	5-26
5-8.	Unisys Workstation Products That Can Be Localized.....	5-27
6-1.	Identifying the Available Coded Character Sets and Ccsversions.....	6-3
6-2.	Obtaining Coded Character Set and Ccsversion Information.....	6-5
6-3.	Mapping Data from One Coded Character Set to Another.....	6-6
6-4.	Option Parameter Values.....	6-7
6-5.	Processing Data According to a Ccsversion.....	6-9
6-6.	Comparing and Sorting Text.....	6-13
6-7.	Positioning Characters.....	6-15
6-8.	Determining the Available Natural Languages.....	6-16
6-9.	Accessing CENTRALSUPPORT Library Messages.....	6-20
6-10.	Identifying the Available Convention Definitions.....	6-21
6-11.	Obtaining Information about Conventions.....	6-22
6-12.	Formatting Dates According to a Convention.....	6-24
6-13.	Formatting Times According to a Convention.....	6-26
6-14.	Formatting Numeric Data According to a Convention.....	6-28
6-15.	Formatting Monetary Data According to a Convention.....	6-28
6-16.	Determining the Default Page Length and Width.....	6-30

Tables

6-17.	Adding, Modifying, and Deleting Conventions	6-31
7-1.	Comparison of International Features in Data Communications and Networks	7-1
7-2.	Configuring BNA Version 2 Networks for Internationalization.....	7-2
7-3.	Configuring HLCN for Internationalization	7-4
7-4.	Ccsversions and Their Character Sets.....	7-23
8-1.	Locale Categories	8-3
8-2.	Predefined Locales	8-4
8-3.	Locale Components in the LOCALES Array	8-6
8-4.	Effects of Locale Components on Character-Handling Functions.....	8-17
8-5.	Effects of Locale Components on Locale Function Members	8-18
8-6.	Effects of Locale Components on String-Handling Functions.....	8-25
9-1.	Coded Character Sets Supported in the Current Release	9-2
9-2.	Eight-Bit CCS Mapping Tables in the Current Release.....	9-6
9-3.	Double-Byte CCS Mapping Tables in the Current Release	9-10
9-4.	Mixed, Multibyte CCS Mapping Tables in the Current Release	9-11
13-1.	Double-Byte and Mixed, Multibyte Coded Character Sets by Nationality Supported	13-2
14-1.	Ccsversions in the Current Release	14-1
14-2.	Arabic20 Ccsversion Definitions	14-3
14-3.	ASeriesNative Ccsversion Definitions	14-5
14-4.	Brazilian Ccsversion Definitions.....	14-6
14-5.	CanadaEBCDIC Ccsversion Definitions	14-9
14-6.	CanadaGP Ccsversion Definitions	14-12
14-7.	Czechoslovakia Ccsversion Definitions	14-16
14-8.	France Ccsversion Definitions	14-19
14-9.	Hungarian Ccsversion Definitions.....	14-22
14-10.	LatinGreek Ccsversion Definitions	14-25
14-11.	Norway Ccsversion Definitions	14-30
14-12.	Polish Ccsversion Definitions	14-33
14-13.	Romanian Ccsversion Definitions.....	14-36
14-14.	Russian Ccsversion Definitions	14-39
14-15.	Spanish Ccsversion Definitions	14-43
14-16.	Swedish1 Ccsversion Definitions	14-46
14-17.	Swiss Ccsversion Definitions	14-49
14-18.	Turkish Ccsversion Definitions	14-53
16-1.	Best Fit Mapping to a Host CCS.....	16-4
16-2.	HRESULT Values	16-5
16-3.	Mapping Methods Results	16-8
16-4.	Methods of the IXlateEBCDIC Interface.....	16-10
16-5.	Enumerated Constants for CCS Names and Numbers	16-21
16-6.	Microsoft Codepage Values for a Host CCS.....	16-22
16-7.	Values for the RSLT Parameter	16-23
C-1.	Complex Procedure Parameters.....	C-4

D-1.	Help Text Available for Translation.....	D-2
E-1.	Messages Available for Localization with MSGTRANS	E-1
F-1.	Elements of a Railroad Diagram.....	F-2

Tables

Section 1

Understanding the MultiLingual System (MLS)

About This Guide

Purpose

The *MultiLingual System (MLS) Administration, Operations, and Programming Guide* presents a functional overview of the MLS environment. This environment provides access to internationalization features, which enable you to create and run application systems that are localized to display and process data for a particular language, culture, or line of business.

This guide includes the following information:

- A description of MLS capabilities, concepts, and terminology
- Guidelines for establishing the MLS environment on your system
- Guidelines for setting MLS options at the system, user, session, and task levels
- Guidelines for translating Unisys products
- Guidelines for using translated Unisys products
- Guidelines for creating application systems that can be effectively translated and localized

Audience

The primary audience for this guide includes

- System managers and operators who establish the MLS environment on a host system and determine the default settings at the system level
- Administrators, managers, and others who determine the requirements for localizing application systems.
- Application programmers who create application systems that can be localized

How to Use This Guide

You can read the information presented in this guide serially or randomly. For specific information, refer to the following sections:

- For an overview and general understanding of MLS, read Section 1, "Understanding the MultiLingual System (MLS)," and Section 2, "Locating and Interfacing with MLS Components."
- For a description of the templates used in the MLS environment, read Section 3, "Understanding Templates in Convention Definitions."
- To understand how to set up the MLS environment on your system, and for instructions on viewing and setting the MLS options, read Section 4, "Establishing an MLS Environment."
- For information on using translated Unisys products, read Section 5, "Localizing Products in an MLS Environment."
- To understand the requirements for creating application systems that can be translated, read Section 6, "Creating Application Systems in an MLS Environment."

Angle brackets (<>) are used to identify variable syntactical elements, for example, the construct <name>. Syntax diagrams and syntax elements are noted in the index under the appropriate construct. An explanation of railroad diagrams (a method of depicting syntax) is found in Appendix F, "Understanding Railroad Diagrams."

The names of the coded character sets, ccsversions, and conventions are not sensitive to capitalization. The system recognizes ASERIESNATIVE, aseriesnative, and ASeriesNative as the same name. Within text, the names are usually shown in uppercase and lowercase characters, for example, ASeriesNative. Within an example, the names are usually shown in uppercase characters, such as ASERIESNATIVE.

The terms translation table and transliteration table, which were used in previous releases, have been replaced by the term coded character set mapping table.

In the text, italics are used to introduce new terms or to identify words used in an unusual manner.

Understanding the MultiLingual System (MLS)

This section discusses MLS concepts, functions, and features. It provides a basic understanding of MLS and introduces the terminology used in the remainder of the guide.

An application system designed to meet the needs of international customers might have specific local needs. For example, it might need to support several languages, or it might need to use special formatting for a local market. One way to meet such requirements is to design your application system so that the screens, messages, and other elements of the user interface can be presented in languages and formats that are familiar to you.

MLS gives you the flexibility to create, tailor, and run application systems for a multilingual, multicultural business environment.

What Is MLS?

MLS is not a single product but an environment that encompasses many products. The MLS environment includes a collection of operating system features, productivity tools, utilities, and compiler extensions. This environment enables you to

- Translate online screens, forms, menus, help text, and messages.
- Define standard formats for the presentation of dates, times, numbers, and monetary data based on the needs of the application system user.
- Install and run translated application systems.
- Display text using the language, characters, symbols, and formatting relevant to the individual users of your application system.
- Present the application system in several languages or conventions at the same time on the same system.
- Dynamically change the language or convention specifications so that users can access the application system using the language and formatting most useful to them.
- Make changes to a previously translated application system and then translate only those changes, without retranslating the entire user interface.

Understanding MLS Terminology

The following text defines terms used in this guide to describe the MLS environment.

What Is an Application System?

The term *application system* refers to the combination of a user interface and an application program. The *user interface* encompasses the components within the application system that the application user sees and interacts with, such as screens, forms, menus, messages, online help, and input strings. The *application program* is an ALGOL, C, COBOL74, NEWP, Pascal, or RPG program that processes the data either entered by the user through the user interface or received from some other source, such as a file loaded into a database. The application program can contain elements of the user interface, such as error messages that are displayed to the application user.

What Is a MultiLingual System?

In its common use, the term *multilingual* indicates the ability to use several languages. In this guide, the term is expanded to encompass more than spoken or written languages. It indicates the ability to meet both the linguistic and the cultural requirements for a given group of users. For example, a multilingual application system not only presents data in the users' language, but formats data according to the conventions of the users' culture or line of business.

In this guide, the term *language* is generally used to refer to a *natural language*, that is, a human written or spoken language as opposed to a language used by machines. Some examples of natural languages are English, Spanish, and Japanese. The languages parsed by the compilers, such as ALGOL and COBOL74, are called *programming languages*.

Note that a natural language can differ from country to country, or from region to region. For example, English is written and spoken in many countries, but the English written and spoken in the United States differs from the English written and spoken in Great Britain.

The term *convention* designates the formatting requirements for a group of users. As with languages, conventions can vary based on factors such as country, line of business, or culture. For example, in some countries it is a convention to use a period (.) to indicate the decimal place; in other countries, a comma (,) is used. In the MLS environment, conventions include the agreed-upon formats for presenting date, time, numeric, or monetary information, and the default settings for printing data, such as page length and line width. In the system software, these convention elements are combined in groups that are called *convention definitions*.

How Are Cultures and Localized Environments Supported?

One reason for localizing an application system is to display the user interface components in a language or format familiar to the application user. By using the MLS environment, you can customize the user interface so that it can be displayed in a variety of languages, such as French and German. You can then present this interface simultaneously in many languages to many different users.

The ability to develop a user interface for a given group of users involves several factors, including

- The ability to present text in the users' natural language
- The availability of an appropriate character set to represent information in the natural language
- The ability to present and process data according to the rules of a natural language or a culture
- The ability to follow the business and cultural conventions for the presentation of dates, times, numbers, and currencies

The MLS environment provides support for developing custom user interfaces. The following concepts are used in describing this support:

- Character set

A *character set* contains the letters, numbers, symbols, and special characters used to represent data in a particular locality or line of business. A given character set can be useful for many environments, such as different countries, different languages, or different lines of business. For example, the LATIN1 character set contains the characters used for business applications in most Western European countries.

- Coded character set

A *coded character set* is a character set in which each character is assigned a code value. The same character set can exist with a variety of encodings. For example, the LATIN1-based character set can be encoded in an International Organization for Standardization (ISO) format or an EBCDIC format. Although the system uses EBCDIC codes, it also recognizes other format codes and can map characters from one coded character set to another with a coded character set mapping table. A *coded character set mapping table* is a table that maps characters from one coded character set to another coded character set.

- Ccsversion

A *ccsversion* designates the way a coded character set is processed to meet the needs of a particular natural language, culture, or line of business. The *ccsversion* includes rules concerning the way data is presented, collated, capitalized, and so on, which can vary from language to language. The rules also define the data classes to which characters belong, including alphabetic, numeric, and space. The *ccsversion* can address the needs of users who speak the same language but require different methods for processing data because of regional differences in the language or requirements for a specific line of business. Therefore, several *ccsversions* can be based on the same coded character set.

Understanding the MultiLingual System (MLS)

For example, the `ccsversion` might determine the way data is to be presented for the same language used in two countries, such as the French used in France versus the Canadian French used in Canada. It might also define different rules for a French word processing application versus a French financial application.

Figure 1-1 shows the relationship of coded character sets and `ccsversion`s.

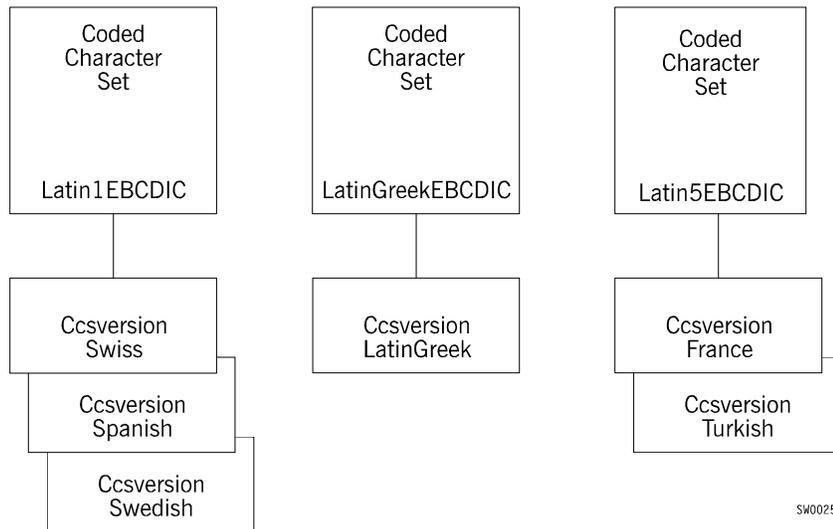


Figure 1-1. Coded Character Sets and Ccsversions

- Convention or convention definition

The various convention elements are grouped in a structure called a *convention* or a *convention definition*. A convention enables you to define templates for formatting dates, times, numbers, and currencies according to the requirements of a particular locality, culture, or line of business. It also enables you to set default values for page length and page width. The templates and other information in a convention are referred to as *elements* of the convention.

The system software contains the coded character sets and `ccsversion`s used in the MLS environment. You cannot modify these components. The system software also contains standard convention definitions, which you cannot modify. However, you can create, modify, and delete custom convention definitions. The system software also has tools that enable you to translate products into any language.

What Processes Are Performed in the MLS Environment?

The guide uses three related terms to describe the processes associated with the MLS environment: internationalization, localization, and translation.

Internationalization is the process of making available the MLS environment. This environment includes the software, firmware, and hardware features that enable you to develop and run application systems that are customized to meet the needs of a specific language or business environment. The internationalization features provide support for several character sets, a variety of international business and cultural conventions, extensions to data communications protocols, and the ability to use one or more natural languages concurrently.

Localization is the process of customizing an application system for a particular locality or environment by using the internationalization features. It includes translating the screens, messages, and online text to another language; choosing a particular character set for data to be processed in; and defining the formats for representing date, time, numbers, and currency for a particular business application.

Translation is part of localization and is the process of converting text from one language to one or more other languages. Several software products are used in the MLS environment to assist you in the translation process.

Understanding the Components of the MLS Environment

Support for a variety of languages and cultures means that support of coded character sets, ccsversions, languages, conventions, and the HostCCS is part of the MLS environment. These components are defined earlier in this section under the heading “How Are Cultures and Localized Environments Supported?” The following paragraphs describe these elements in more detail.

Coded Character Sets

Each system software release includes a data file containing all coded character sets that are supported on the system. You cannot choose a coded character set directly, but by choosing a ccsversion, you implicitly designate the default coded character set for your application system. Refer to Section 9, “Coded Character Sets and Coded Character Set Mapping Tables,” for a list of the coded character sets provided on the release media.

Each coded character set contains the following types of characters:

Type of Characters	Explanation
Graphic characters	Characters that are the actual or coded representation of a digit, an uppercase or lowercase letter, or a special symbol in printable or displayable form
Space character	A graphic character that when received by a printer or display device causes the printing or display position to advance one character space
Control characters	Characters whose occurrence in a particular context initiates, modifies, or stops control operations—for example, the carriage return character or the end-of-file marker

Associated with each coded character set are the following items:

- The coded character set name
Each coded character set is identified by a unique coded character set name. The name must start with a letter, contain only AseriesNative alphanumeric characters, and be no longer than 17 characters.
- The library coded character set number
The library coded character set number is a positive integer value that is used to refer to a coded character set in calls to the CENTRALSUPPORT library. The library coded character set number is sometimes referred to as the library ccsnumber.
The library coded character set number can also be used as a value for the INTMODE and EXTMODE file attributes. The INTMODE file attribute specifies the internal or logical character size of the records in a file. The EXTMODE file attribute specifies the external or physical character size of the records in a file. Either the common name or the library coded character set number can be used as the value

for the INTMODE and EXTMODE file attributes. For example, you can use either Latin5EBCDIC or the number 14 as the INTMODE or EXTMODE value for the Latin5EBCDIC coded character set.

Files based on 8-bit coded character sets can have the same or different INTMODE and EXTMODE values. Whenever the INTMODE and EXTMODE values differ, the system attempts to map the data by using the coded character set mapping tables in the SYSTEM/CCSFILE file. If no coded character set mapping table exists in the SYSTEM/CCSFILE file that explicitly maps the coded character sets, the system issues an OPEN error message. Refer to the *File Attributes Programming Reference Manual* for information on the way to use the INTMODE and EXTMODE file attributes.

Unisys software personnel have the final authority to assign the library coded character set number.

- The number of bits needed to represent each character

The number of bits per character is 8 bits, 16 bits, or a mixture of 8 and 16 bits.

- The repertoire number

The repertoire number is a unique number that identifies a distinct collection of characters. Coded character sets that consist of the same characters have the same repertoire number. The repertoire number is not used by coded character sets defined in the SYSTEM/CCSFILE from the HMP 3.0/SSR 44.2 release and forward (that is, the value is zero).

- The encoding number

The encoding number is a unique number that distinguishes different encodings of the same character repertoire. The encoding number is not used by coded character sets defined in the SYSTEM/CCSFILE from HMP 3.0/SSR 44.2 and forward (that is, the value is zero).

- The hexadecimal coded value for each character in the set

To find the hexadecimal number that corresponds to a character on a coded character set table, refer to Section 9, "Coded Character Sets and Coded Character Set Mapping Tables."

- The coding format for the coded character set

The coding format identifies the underlying format for the coded character set.

Understanding the MultiLingual System (MLS)

The following information provides the acronym, integer value, and full name for each supported coding format.

Acronym	Integer Value	Full Name
2200MB	10	ClearPath IX multibyte
ASERIESEBCDIC	2	A Series EBCDIC
BTOS	4	B 2x or B 3x operating system, such as a B 20 or a B 38
DOUBLEBYTE	13	Pure double-byte (16-bit) data
EBCDICMB	6	Standard EBCDIC multibyte
EUCMB	9	Extended UNIX code multibyte
ISO	1	International Standards Organization
ISOMB	7	International Standards Organization multibyte
PC	5	Personal computer
PCMB	8	Personal computer multibyte
STDEBCDIC	3	Standard EBCDIC as defined in Corporate Standard 4000 0085
UCSBMP	14	16-bit Universal code set standard definition (big-endian)
UCSBMPNT	15	16-bit Universal code set definition (little-endian)

The ASERIESEBCDIC and STDEBCDIC coded character sets are based on the same EBCDIC format, but the placement of a few characters differs between the two.

- The ccsversion that can be used with the coded character set
- The coded character set mapping tables used to map characters from one coded character set to another

A coded character set mapping table maps data in one coded character set to data in another coded character set. For more information on coded character set mapping tables, refer to Section 9, "Coded Character Sets and Coded Character Set Mapping Tables."

Ccsversions

For each coded character set, the release media contains standard ccsversions that describe the rules for processing the coded character set to meet the requirements of a variety of countries. You can process a coded character set with more than one set of rules; therefore, several ccsversions can be based on the same coded character set. One file includes both the standard ccsversions and the coded character sets. Information on each ccsversion is provided in Section 14, "Ccsversions."

One ccsversion is established as the default for the system. You can set the ccsversion at the system level only. Data processed by the system is then processed according to the rules for that ccsversion. The application can access the information for any defined ccsversion, whether it is based on the same coded character set or not. This feature enables the program to process data by using a ccsversion other than the system default.

Each ccsversion contains the following information:

- A number that uniquely identifies the ccsversion
- The name of the ccsversion, for example, CanadaEBCDIC
- The name of the coded character set on which the ccsversion is based (base coded character set)
- The information needed to process data according to ccsversion rules. This information includes data class sets, mapping relationships, and ordering rules. More about the information needed to process data appears later in this section.
- Escapement definitions that define the escapement characteristics in absolute terms within a line of text. Information about escapement definitions is provided later in this section.

Escapement Definitions

Each ccsversion contains the escapement definitions. Escapement definitions include the following elements:

- Text line orientation (TLO). The TLO value can be either horizontal (H) or vertical (V).
- Line advancement direction (LAD). The LAD defines the order text lines follow when they are advancing. The LAD value is either positive (+) or negative (-) relative to the TLO.
- Character advancement direction (CAD). The CAD defines the order in which characters normally follow one another within a line of text and the text line starting position. The CAD value is either positive (+) or negative (-) relative to the TLO.

In addition, each code position within a ccsversion can have character escapement directions (CEDs). For each code position within a ccsversion, the CED defines the escapement characteristics in absolute terms within a line of text.

A character can have one or two CED values. Each value is separated by a comma (,). The first value is used for the presentation of the character. The second value becomes active once the character has been presented. For example, depending on the language,

when a letter is imaged, the cursor can stay at the same active position or the cursor can move to another active position. A conversion might keep the same active position to overstrike zeroes to distinguish them from the letter *O*.

Table 1-1 shows the relationship of the TLO and LAD.

Table 1-1. Relationship of TLO and LAD

TLO	LAD	Lines Advanced
H	+	Top to bottom
H	-	Bottom to top
V	+	Left to right
V	-	Right to left

Table 1-2 shows the relationship of the TLO and CAD.

Table 1-2. Relationship of TLO and CAD

TLO	CAD	Starting Position	Characters Advance
H	+	Leftmost	Left to right
H	-	Rightmost	Right to left
V	+	Topmost	Top to bottom
V	-	Bottommost	Bottom to top

Together the TLO, LAD, and CAD define the home position as shown in Table 1-3.

Table 1-3. Relationship of TLO, LAD, and CAD

TLO	LAD	CAD	Home Position
H	+	+	Top left
H	+	-	Top right
H	-	+	Bottom left
H	-	-	Bottom right
V	+	+	Top left
V	+	-	Bottom left
V	-	+	Top right
V	-	-	Bottom right

The CED value is relative to the TLO value. Table 1-4 relates the TLO and CED.

Table 1-4. Relationship of TLO and CED

TLO	CED	Result
H	+	Advances characters from left to right.
H	-	Advances characters from right to left.
V	+	Advances characters from top to bottom.
V	-	Advances characters from bottom to top.
H or V	0	Presents next character at the same active position.
H or V	*	Uses the same escapement direction as the most recent escaping character.
H or V	blank	Uses the same escapement as the CAD.
H or V	S+	Sets the CAD to the value of the CAD + (plus) parameter; the character CED is identical to the CAD.
H or V	S-	Sets the CAD to the value of the CAD -(minus) parameter; the character CED is identical to the CAD.
H or V	R	Sets the CAD to the value of the CAD that was active before the occurrence of the character that set the current value.

Information Needed to Process Data by Ccsversion Rules

Each ccsversion contains the information needed to process data according to ccsversion rules. This information includes the following:

- The graphic characters that make up each of the six data classes: alphabetic, numeric, space, presentation, lowercase characters, and uppercase characters. An explanation of each data class is as follows:

Data Class Name	Is a Collection of Characters that Are Treated as . . .
Alphabetic	Alphabetic characters
Numeric	Numbers
Space	Space characters
Presentation	Characters that can be imaged on a presentation device
Lowercase	Lowercase characters (optional)
Uppercase	Uppercase characters (optional)

Each ccsversion must have a definition for the alphabetic, numeric, space, and presentation data classes. The lowercase and uppercase data classes are optional.

- Contents of mapping tables.

Each ccsversion can contain as many as five mapping tables. Each ccsversion must have an escapement table definition. The other tables are optional.

The following list describes each mapping table:

Mapping Table Name	Maps
Escapement	Each character to its escapement value. The escapement value of each character and line defines the movement of the active position after a character is displayed. For example, in Latin-based languages, the active position typically advances one character position to the right after a character is imaged, and the text lines advance from top to bottom. Many languages require more elaborate escapement. For example, languages such as Arabic and Hebrew require characters to advance from right to left instead of left to right.
Lowercase to uppercase	Lowercase characters to uppercase characters.
Uppercase to lowercase	Uppercase characters to lowercase characters.
Numeric to alternate digits	The numbers 0 through 9 to alternate representations of 0 through 9.
Alternate digits to numeric	Alternate digits to the corresponding numerical values of 0 through 9.

- Rules for ordering information.

Each ccsversion contains ordering information that is used when data is collated or sorted. The ordering information is explained in the following list:

- The *ordering sequence value (OSV)* for each character, which enables different characters to have the same ranking. The OSV is an integer value between 0 and 255, which is assigned to each code position in a coded character set and represents a relative ordering value. An ordering value of 0 signifies the lowest ranking in the coded character set and an ordering value that is greater than all other ordering values signifies the highest ranking. More than one code position can be assigned the same ordering value.
- The *priority sequence value (PSV)* for each character, which can assign a different ranking to each character. The PSV is an integer value between 1 and 15, which is assigned to each code position in a coded character set and represents a relative priority value within an OSV. Each code position that has a unique ordering value is assigned a priority value of 1. For code positions with the same OSV, a priority value of 1 signifies the lowest ranking in the OSV and a priority value that is greater than all other priority values signifies the highest ranking.
- The *substitution rules* that enable a string of characters to be ordered as one character or one character to be ordered as a string. A predetermined string of as many as three characters can be ordered as if the string were one character. The

ordered character string is assigned an OSV and a single PSV. For example, in Spanish the letter pair *ch* is ordered as if it were a single letter, different from either *c* or *h* and ordered between *c* and *d*. Conversely, a single character can be used to generate a string of two or three OSVs and PSVs. Substitution rules are optional in ccsversions.

Languages

Many application systems have user interfaces that are written in English. Therefore, an application system often must be translated to another natural language. Tools exist that enable you to translate many elements of an application system to your natural language. You can use MLS features to set a natural language at the system, usercode, session, and task level. This feature enables users connected to the same system to view and process data in different languages. In fact, this feature enables a single user to set different languages for individual tasks or sessions. Once a language is in effect, all menus, screens, forms, messages, and help text appear in that language. If the designated language has not been defined to the system, the text appears in the original language.

You can define a new language to the system by translating menus, screens, forms, messages, and help text to another language and by adding a new language name to the system. See "Support for Natural Languages" in Section 2 for information on using natural languages on your system.

Convention Definitions

The release media contains a set of standard convention definitions for formatting dates, times, numbers, currencies, and page sizes as required by a particular locality, culture, or line of business.

The standard convention definitions are described in Section 15, "Standard Conventions." In addition, you can create custom convention definitions to meet your specific needs either by using the Menu-Assisted Resource Control (MARC) screen interface, as described in Section 4, "Establishing an MLS Environment," or by calling CENTRALSUPPORT procedures from an application program, as described in Section 6, "Creating Application Systems in an MLS Environment."

A custom convention definition is one that is created to meet the specific needs of a site or user. It has the same elements as a standard convention but defines different formatting needs. Once you establish a custom convention, it has the same characteristics as a standard convention except that you can modify or delete it.

You can set the convention at the system, usercode, session, or task level. The date, time, numeric values, and currencies are then formatted according to that convention.

Table 1-5 describes the elements contained in each convention definition. Some items contain single values; other items consist of a series of values. When an item contains a single value, the value is defined in its own field of a convention. However, when an item consists of a series of values, the format for the values is designated in a *template*.

Table 1-5. Elements in Convention Definitions

Element	Description
Convention name	Specifies the name of the convention definition. The name can be a maximum of 17 alphanumeric characters, including uppercase A through Z, lowercase a through z, and numerals 0 through 9. The name must start with a letter and cannot have any blanks.
Long date template	Gives the format and editing instructions for the day, month, and year—including all literals—when the day and month are listed using their full names. Prefixes and suffixes are included. For example, Monday, January 15th, 1990 is a long date. The maximum length of a formatted long date is 45 characters, including spaces.
Short date template	Gives the format and editing instructions for the day, month, and year—including all literals—when the day and month can be abbreviated. For example, Mon., Jan. 15, 1990 is a short date. The maximum length of a formatted short date is 20 characters, including spaces.
Numeric date template	Gives the format and editing instructions for the day, month, and year as numeric values. For example, 90-01-15 is a numeric date. The maximum length of a formatted numeric date is 10 characters.
Long time template	Gives the format and editing instructions for the time. The template can contain a time prefix (such as @ or on); the hours (either in 12-hour or 24-hour notation); the minutes, seconds, and partial seconds; and a time addendum (such as a.m. or p.m.). For example, 12:47:24.025 a.m. is a long time. In addition, the template can specify the use of the literals, prefixes, or suffixes. The maximum length of the formatted long time is 45 characters.
Numeric time template	Gives the format and editing instructions for the time as numeric values. The template can contain the hours (either in 12-hour or 24-hour notation), the minutes, and the seconds. For example, 21:17:09 and 01:33 are numeric times. The maximum length of the formatted numeric time is 8 characters.
Numeric template	Contains the positive and negative sign symbols, the thousands separator symbol, and the decimal symbol for numeric applications. In addition, the numeric specifications show the placement of the symbols in relation to the amount. For example, the template can specify that the thousands separator is a comma (,) and that the comma is to be used for all amounts equal to or greater than 1,000,000. Therefore, the amount -48569.17928 would not have a thousands separator.
Monetary template	Contains the currency symbol, the positive and negative sign symbols, the thousands separator symbol, and the decimal symbol for monetary applications. In addition, the monetary specifications show the placement of the symbols in relation to the amount and in relation to the positive and negative symbols. For example, \$-1,999,888.77 and F 99,988- are monetary amounts in which the thousands separator is a comma and the separator is used for all amounts equal to or greater than 1,000.

Table 1-5. Elements in Convention Definitions

Element	Description
Number of maximum integer monetary digits	Establishes the maximum number of integer digits for any monetary amount. For example, if the data item value is 22, the total number of integers cannot exceed 22.
Number of fractional monetary fractional digits	Establishes the maximum number of fractional digits for any monetary amount. In most convention definitions, the number of fractional digits is 2.
Number of international monetary fractional digits	Establishes the maximum number of fractional digits for any monetary amount in international currency notation.
International currency notation	Designates an abbreviated currency designator to be used in the international payment transfer sector such as Swift. For example, FRF represents the French currency.
Default page length	Establishes the default print page length (lines per page). The default page length can be from 12 through 255 lines, inclusive.
Default page width	Establishes the default print page width (characters per line). The default page width can be from 40 through 255 characters, inclusive.

Templates

You define formats for dates, times, monetary values, and numeric values by using templates. Each template contains the pattern for representing an element. You use predefined characters, called *control characters*, to designate the components of this pattern. For example, you define a numeric date, such as 02/25/1990, by creating the following template:

```
!0o!/!dd!/!yyyy!
```

This template uses the following control characters to define the pattern for representing a numeric date:

Control Characters	Explanation
!	A delimiter separating the components of the pattern
0	A character indicating that the leading zero is to appear in the date component, if it exists
o	A character indicating a digit of the month
d	A character indicating a digit of the day
y	A character indicating a digit of the year
/	A literal character separating the components of the date when the date is displayed

For a complete list of the control characters used in templates, see Section 3, "Understanding Templates in Convention Definitions."

HostCCS

The HostCCS system option allows system administrators to identify the default character set for their systems when the default ccversion is ASERIESNATIVE but the default CCS used on the system is not ASERIESEBCDIC. This makes it possible to properly interpret data and messages that are not encoded in ASERIESEBCDIC. It is primarily useful for systems that use a complex or nonstandard default CCS that has no CCSVERSION defined with its use.

Section 2

Locating and Interfacing with MLS Components

This section describes the components that establish the MLS environment and the features that you can use to localize system software products and your own application systems. The operating system, compiler extensions, programming tools, and utilities provide access to the components of the MLS environment. More detailed information on using these features is presented in later sections of this guide.

Where Are the MLS Components Defined?

The *Centralsupport architecture* and the natural languages bound into code files establish the MLS environment on your system. The coded character sets, *ccsversion*s, and conventions are three components of the MLS environment that are supported by the Centralsupport architecture. This architecture includes

- The SYSTEM/CCSFILE file, which is a file containing the coded character sets and the *ccsversion*s
- The SYSTEM/CONVENTIONS file, which is a file containing the conventions
- The CENTRALSUPPORT library, which is a system library providing programmatic access to these files

The operation of the Centralsupport architecture is generally not visible to you or your application system users. However, it is useful to understand the way the components of this architecture are used.

One or more natural languages can be defined within the application program. See “Support for Natural Languages” later in this section for more information about using application systems to communicate with the system in your natural language.

You can create the application system by using one natural language as a base and then translate the user interface components into other languages without impacting the underlying application program. At run time, the user interface components are displayed in the appropriate language based on a user's default settings for the language option.

The following text describes the key components of the Centralsupport architecture and explains the way natural languages are bound into the code files.

SYSTEM/CCSFILE File

The file SYSTEM/CCSFILE supplied on the release media contains the data for the standard coded character sets, coded character set mapping tables, and ccsversions. The SYSTEM/CCSFILE file contains the following information:

- The definitions of each coded character set
- The coded character set mapping tables used to map characters from one coded character set to another coded character set
- The ccsversions that can be used with the coded character sets

Note that this file is required only if you choose to use the internationalization features. A default ccsversion, called ASeriesNative, can be used without a SYSTEM/CCSFILE. The ASeriesNative ccsversion enables your system to run without the internationalization features.

When using ccsversions or the SYSTEM/CCSFILE with KEYEDIOII files or Enterprise Database Server databases across multiple hosts, the SYSTEM/CCSFILE must be the same on all of the hosts. If the CCSFILEs differ, you might encounter data corruption in the KEYEDIOII or database files.

For more information about the coded character sets, ccsversions, and coded character set mapping tables contained in the SYSTEM/CCSFILE file, refer to Section 9, "Coded Character Sets and Coded Character Set Mapping Tables" and to Section 14, "Ccsversions."

SYSTEM/CONVENTIONS File

The SYSTEM/CONVENTIONS file contains the standard convention definitions provided on the release media. You cannot alter these convention definitions.

The record format in this file can change on any release, without prior notice.

You can add your own custom conventions to the SYSTEM/CONVENTIONS file. You can modify and delete custom conventions as needed. When you add a new convention, it becomes available for use immediately. If you modify or delete an existing convention, it is modified or deleted after the next halt/load.

For more information about conventions, refer to Section 15, "Standard Conventions." For information about adding custom convention definitions to the SYSTEM/CONVENTIONS file, refer to "Adding Custom Convention Definitions" in Section 4.

CENTRALSUPPORT Library

The CENTRALSUPPORT library provides your application system with an interface to the SYSTEM/CCSFILE and SYSTEM/CONVENTIONS files. The library contains a set of procedures that your program can call to perform tasks including, but not limited to, the following:

- Determine the current system settings for the ccsversion, language, and convention.
- Obtain information about the way to process and display data for a ccsversion or convention.
- Add, modify, or delete custom conventions.
- Determine if a designated coded character set, ccsversion, or convention is available on the host system.
- Use mapping tables to map data from one coded character set to another coded character set, map lowercase letters to uppercase letters, and perform other conversions.
- Use ALGOL truthset tables to obtain the characters in a data class and to determine if a character is in a specific data class.
- Compare strings.

The CENTRALSUPPORT library procedures that access the SYSTEM/CCSFILE file enable you to inquire about information in the file, but not to modify the file. The procedures that access the SYSTEM/CONVENTIONS file enable you to inquire about both standard and custom conventions and to add, modify, or delete custom conventions from a privileged program.

Note that the system software also uses the CENTRALSUPPORT library to process your data according to the ccsversion and convention in use.

Support for Natural Languages

The language feature of MLS enables you to communicate with the system in your own natural language. This feature is supported through translated input and output messages, which are bound into the code files for system software products and your own application systems. Application programs that support localized messages contain OUTPUTMESSAGE ARRAY declarations and the MESSAGESEARCHER statement. Application programs with localized forms and screens use the Screen Design Facility Plus (SDF Plus) and Interactive Menugraph Generator (IMG). The Help Utility and the SDF Plus Help utility enable online help text to be translated. For more information about using these tools, refer to Section 5, "Localizing Products in an MLS Environment," and Section 6, "Creating Application Systems in an MLS Environment."

Interfaces to the MLS Environment

Once you establish the ccsversions, languages, and conventions, you can view and set internationalization components at five levels. The levels described in Table 2–1 are listed in order from lowest to highest level. Note that you can set the ccsversion only at the system level. You can set language and convention at all levels. Table 2–2 shows the impact of these settings at each level.

Table 2–1. Access to Internationalization Features

Level	Description
System	<p>A person authorized to use the system console can use the SYSTEMOPTIONS system command or the MARC menus and screens to do the following:</p> <ul style="list-style-type: none"> • Set and view the system defaults for ccsversion, language, and convention. • View, add, modify, or delete custom conventions.
Usercode	<p>A person authorized to use the system console can override the system defaults for an individual user. The defaults are changed by defining the language and convention options in the USERDATAFILE file for that user.</p>
Session	<p>Any user can change the language and convention for a MARC or CANDE session by using either the LANGUAGE and CONVENTION commands or the MARC screen interface. Setting the options at the session level overrides both the system default settings and the options set in the USERDATAFILE file.</p> <p>Users can set the language and convention for certain products that support sessions, such as SDF Plus. In the SDF Plus environment, users can designate a language for the forms, text, and messages they are creating. At run time, the SDF Plus processing mechanism displays these elements in the appropriate form or language and continues processing based on the default settings of the user for language and convention.</p>
Task	<p>At run time, an application system can designate the language and convention with the LANGUAGE and CONVENTION task attributes. Setting these attributes changes the language and convention for the task while the task is running.</p>
Program	<p>An application program can access the procedures in the CENTRALSUPPORT library to perform actions based on any ccsversion, language, or convention. Some programming languages have syntax that enables you to specify a ccsversion or convention within the program.</p>

Table 2-2. Impacts of System Levels for a Natural Language and Convention

Level	Impact
System	Establishes the system default language or convention.
Usercode	Overrides the system level setting for that usercode.
Session	Overrides the settings of both the system and usercode levels for that session.
Task	Overrides system, usercode, and session settings for that task.
Program	Overrides all other settings for that program.

Tools for Localizing Multilingual Application Systems

You can use compiler extensions, programming tools, and utilities in the MLS environment to create and localize application systems. The following text describes the roles of these tools in the MLS environment.

Compilers That Support the MLS Environment

The ALGOL, C, COBOL74, NEWP, Pascal, and RPG compilers enable you to design application programs that support coded character sets, ccsversion, natural languages, and business and cultural conventions. Programs using these compilers can access the internationalization features through calls to procedures that are predefined in the CENTRALSUPPORT library and, in some cases, through constructs provided in the programming language.

For more information on one of these compilers, refer to the programming reference manual of the compiler you are using. These manuals are listed in the bibliography of this guide. You additionally should refer to Section 8, "Using C Programs in an MLS Environment," when you are using the C compiler.

Programming Tools and Utilities Used in the MLS Environment

To enable users to view an application system in a natural language other than the one on which the application is based, you must make the translated user interface portion of that system available to users. Four products exist that help you to create a new user interface that can be localized or to localize an existing user interface. These products are

- SDF Plus
You can use SDF Plus as a developmental tool to create a new form-based user interface. Once you create the user interface, you can then localize it by using the SDF Plus features.
- IMG
You can use IMG to create new screens for the Menu-Assisted Resource Control (MARC), IMG, and Interactive Datacomm Configurator (IDC) products.
- MSGTRANS
You can use MSGTRANS to localize the messages in an existing user interface. These messages must be contained in a compiled program.
- Help Utility and SDF Plus Help utility
You can use these utilities to process translated help text and make it available to users.

For more information on using these tools and utilities, refer to Section 5, "Localizing Products in an MLS Environment," and Section 6, "Creating Application Systems in an MLS Environment."

Section 3

Understanding Templates in Convention Definitions

This section explains the purpose and the structure of templates. This information can help you to understand the standard conventions and to create your own custom conventions. Each convention has the same groups of items, as follows:

- Convention name
- Date and time templates
 - Long date template
 - Short date template
 - Numeric date template
 - Long time template
 - Numeric time template
- Monetary and numeric templates
 - Numeric template
 - Monetary template
 - Maximum integer digits in monetary amounts
 - Maximum fractional digits in monetary amounts
 - International monetary fractional digits
 - International currency notation
- Printing templates
 - Default lines per page
 - Default characters per line

Although you cannot modify or delete the standard conventions supplied on the release media, you can add custom conventions to the `SYSTEM/CONVENTIONS` file. You use the same groups of items to define a custom convention.

Each template defines an output format through a combination of control characters and literal text. The templates specify the manner in which the components are expanded to a formatted string.

Parts of a Template

A template is composed of control characters and literal text. The following paragraphs define these two elements of a template. Additional characteristics and special uses of the control characters are explained later in this section under “Control Characters for Date and Time Templates” and “Control Characters for Numeric and Monetary Templates.”

Control Characters

Control characters determine the values that are part of the formatted output string; for example, a control character can determine whether the month is output as a name or as a number.

Control characters must be enclosed in exclamation marks (!). Each pair of exclamation marks can enclose one or more control characters. The control characters can be a series of the same control character or a mixture of different control characters. For example, !w!, !www!, and !00ddy! are all valid strings of control characters.

Control characters are not case sensitive. For example, the following strings of control characters are equivalent: !00ddy!, !00DDYY!, and !00ddy!.

To improve readability, you can place blanks between control characters. For example, !00 dd yy! and !00ddy! are equivalent and produce the same output.

Literal Text

Literal text, including blanks, is presented in its exact form and position.

Literal text can appear anywhere in a template except between adjacent control characters. Therefore, assuming you want to use the slash character (/) as a literal character to separate the components of a numeric date, !00/dd/yy! would not be valid. The literal text must appear outside of the control characters, for example, !00!/dd!/yy!.

If literal text appears after all the control characters, the text must be terminated by an exclamation mark. For example, if you want the literal text BCE (for Before the Common Era) to appear after the date template, you can use the following string: !00!/dd!/yy!BCE!

Null Templates

To create a custom convention definition, you must include a template for all the date and time components required in a convention definition. If the custom or grammar of a particular language does not require one of the date or time components, use the null template in place of the date or time component.

The null template contains the minimum information required for templates. It is composed of one blank space followed by an exclamation mark (!).

Understanding Date and Time Templates

The following text explains the components of date and time templates and the way they are processed.

Processing Date and Time Templates

The date and time components are passed to the CENTRALSUPPORT formatting procedures as EBCDIC input arrays. The date and time components occupy fixed-position fields in the input arrays. Each field also has a fixed length in the input array. The date and time input fields can contain zeros, unsigned integers, or blanks.

The date EBCDIC input array is YYYYMMDD, where the fixed fields are

YYYY	A 4-character field for the year
MM	A 2-character field for the month
DD	A 2-character field for the day

The time EBCDIC input array is HHMMSSPPPP, where the fixed fields are

HH	A 2-character field for the hour
MM	A 2-character field for the minutes
SS	A 2-character field for the seconds
PPPP	A 4-character field for the partial seconds (as a decimal fraction)

The number of character positions designated for each component must be included in the input array. For example, the year must be passed as four character positions. Therefore, both the input values 1991 and 91 are passed to the CENTRALSUPPORT formatting procedures as four characters.

If an input value contains fewer character positions than the length of its corresponding field, the number must be right justified in the field and be blank- or zero-filled. Input values preceded by blanks, zeros, and blank fields themselves, are processed differently depending on the way the template is defined. For example, 1:00 a.m. would result in 01 or 1, depending on the way the template is defined.

Control Characters for Date and Time Templates

Date and time control characters can cause the CENTRALSUPPORT formatting procedures to output alphabetic text, such as the name of the day of the week, or numeric representations, such as a 24-hour clock. Control characters that generate alphabetic text are known as *alphabetic control characters*. Control characters that generate numeric representations are known as *numeric control characters*.

Understanding Templates in Convention Definitions

Depending on the control character, there can be one or more occurrences of the same control character. For example, using the letter *g* as a generic control character, the template might contain *!g!*, *!gg!*, or *!g . . . g!*. The number of occurrences of the control character is identified by the uppercase letter *N*. For example, *N=4* means a field of four consecutive occurrences of the same control character.

Alphabetic Control Characters

Alphabetic control characters cause alphabetic text stored in output message arrays to be placed into the formatted output string. For example, the suffix *a.m.* can be placed after a numeric representation of a 12-hour clock, such as 12 a.m.

Alphabetic control characters access output message arrays in the CENTRALSUPPORT library and display the values stored in those output message arrays. The content of the output message arrays can be translated into any language. For information on translating alphabetic literal text, read “Translating Literal Text Used in Conventions” later in this section.

The alphabetic control characters are shown in Table 3-1.

Table 3-1. Alphabetic Control Characters for Date and Time Templates

Control Character	Output Message Array	Examples of Output
a	TIME_SUFFIX	a.m., p.m.
c	DOM_PREFIX (day of month)	the
e	DOM_SUFFIX	st, nd, rd, th
f	TIME_PREFIX	at
i	HOUR_SUFFIX	hours
k	MIN_SUFFIX	minutes
l	MONTH_PREFIX	of
n	LONG_MONTH	January, December
1n	SHORT_MONTH	Jan, Dec
2n	ALT_MONTH	I, XII
r	SEC_SUFFIX	seconds
v	DOW_PREFIX (day of week)	on
w	LONG_DOW	Monday, Saturday
1w	SHORT_DOW	Mon, Sat
x	YEAR_PREFIX	year
z	YEAR_SUFFIX	A.D.

Alphabetic control characters can appear in one of two syntaxes: as single control characters and as strings of the same control characters.

- A single control character within exclamation marks causes all significant characters in that field to be displayed. For example, `!n!` displays all significant characters for the name of the month. Trailing blanks are ignored.
- When a string of the same control character is used, such as `!aaaa!`, the first N characters in that field are displayed, including trailing blanks. (The first N characters refers to the number of occurrences of the control character.) For example, `!aaaa!` displays the first four characters of a time suffix.

For example, in the date template `!w!, !n! !d!, !yyyy!`, the `w` and `n` are alphabetic control characters. (Note that blanks have been inserted into the date templates as literal text. These blanks are used to create spaces in the output.) Using the example template, the date January 1, 1991 would appear as Tuesday, January 1, 1991, with the full name of the day and month. However, if the template were changed to `!1w!, !1n! !d! !yyyy!`, the date would appear as Tues, Jan 1, 1991.

Numeric Control Characters

Numeric control characters cause dates and times to be output as numeric representations. For example, March 15th, 1991 could be output as 03/15/91 or 15/3/1991.

When output values are generated, numbers preceded by blanks, numbers preceded by zeros, and blank fields are processed differently depending on the control character. For example, leading zeros are always displayed for partial seconds; however, for other control characters, the display of leading zeros depends on the template definition.

The numeric control characters, shown in Table 3-2, can have varying syntactical representations.

Table 3-2. Numeric Control Characters for Date and Time Templates

Control Character	Element	Valid Syntaxes
d	Day of month	!d! or !dd! or !0d!
h	Hour (12-hour clock)	!h! or !hh! or !0h!
j	Day of year	!j! or !j . . . j! or !0j . . . j!
m	Minute	!m! or !mm! or !0m!
o	Month of year	!o! or !oo! or !0o!
p	Partial seconds	!p . . . p!
s	Seconds	!s! or !ss! or !0s!
t	Hour (24-hour clock)	!t! or !tt! or !0t!
y	Year	!y . . . y! or !0y . . . y!

There are six basic syntaxes for numeric date and time control characters:

- Single numeric control character
- Series of two numeric control characters
- Series of three numeric control characters
- Numeric control character preceded by a zero (0)
- Partial seconds (p) control character
- Year (y) control character

Each of these syntaxes is explained in the following text.

Single Numeric Control Character

A single numeric control character within exclamation marks causes all significant characters in that field to be displayed. For example, !m! displays all significant characters for the minute. Trailing blanks are ignored.

A single numeric control character causes the numeric output to be displayed as a 0-, 1-, or 2-digit field. Leading zeros are ignored. For example, !t! displays the 1-digit field 5 for 5 a.m. and the 2-digit field 14 for 2 p.m.

Series of Two Numeric Control Characters

A series of two of the same numeric control character causes the number to be displayed as a 2-digit field. The 2-digit field can contain two digits, a space and a digit (if a leading zero occurs in the input number), or two spaces (if both input digits are zero). For example, !tt! displays a space and a 5 for 5 a.m.

Series of Three Numeric Control Characters

A series of three of the same numeric control character causes a number to be displayed as a 3-digit field. The 3-digit field can contain three digits, a space and two digits, two spaces and a digit, or three spaces. For example, if you use the template !yyjj!, the date July 14th, 1991 is displayed as the Julian date 91195.

Numeric Control Character Preceded by a Zero

In general, if a numeric control character is preceded by a zero, any unspecified position in the formatted output value is represented by a zero. Therefore, using the date template `!0d!/!0o!/!yy!`, March 5th, 1991 is output as 05/03/91. If the template did not have zeros, such as `!dd!/!oo!/!yy!`, blanks are output for empty positions; for example, 5/ 3/91. (Note that for partial seconds output, zeros are always used.)

A zero before a single numeric control character causes the numeric output to be displayed as a 2-digit field, including leading zeros. For example, `!0t!` outputs 05 for 5 a.m.

A zero before two numeric control characters causes the number to be displayed as a 3-digit field, including leading zeros. For example, if you use the template `!yy0jj!`, the date February 1st, 1991 is displayed as the Julian date 91032.

Partial Seconds Control Character

The partial seconds control character can be expressed as one, two, three, or four occurrences of the letter `p`, such as `!p!`, `!pp!`, `!ppp!`, or `!pppp!`. Each of these syntaxes displays the first N digits of the partial seconds. (N can be 1, 2, 3, or 4.) Leading zeros are displayed. Note that leading blanks (spaces) and zeros are displayed as zeros.

A maximum of 4 digits is allowed. If the partial second value is less than 4 digits in length, the value must be left justified in the input field and blank- or zero-filled. The output is also left justified, with trailing-zero padding. For example, the following table shows the output when various partial seconds templates are applied to an input value. The word blank is used in the table to represent a blank in the input value.

Partial Seconds Template	Input Value	Output Value
<code>!pppp!</code>	blank123	0123
<code>!ppp!</code>	blank123	012
<code>!pp!</code>	blank123	01
<code>!p!</code>	blank123	0
<code>!pppp!</code>	123blank	1230
<code>!ppp!</code>	123blank	123
<code>!pp!</code>	123blank	12
<code>!p!</code>	123blank	1

If the input value is N zeros or blanks, the formatted output results in N zeros.

If the length of the partial seconds value is greater than the N digits specified by the partial seconds template, the value is right-truncated. For example, an input value of 0123 with a partial seconds template of `!pppp!` outputs 0123. However, the template `!ppp!` outputs 012.

Year Control Character

The year control character can be expressed as one, two, three, or four occurrences of the letter y, such as !y!, !yy!, !yyy!, or !yyyy!, or one, two, or three occurrences of the letter y preceded by a zero, such as !0y!, !0yy!, or !0yyy!.

When the syntax is the letter y only, the template displays the first N least significant digits of the year. Leading zeros are converted to spaces.

When the syntax is a zero followed by the letter y, the template displays the N+1 least significant digits of the year. Leading zeros are displayed.

For example, the year template !yyyy! outputs the year 800 with a leading space. The template !0yyy! outputs the year with a leading zero, (0800).

Note that the year template !yy! outputs only the two least significant digits of the year. For example, 1991 is output as 91, and 2000 is output as two blanks. The template !0y! outputs the year 2000 as two zeroes (00).

Examples of Date and Time Templates

The first table shows the results of a number of date templates. Column 1 shows the date input array in its YYYYMMDD format. Column 2 gives the date template. Column 3 shows the output date.

Value in EBCDIC Input Array	Date Template	Displayed Output
19900112	!dd!/!0o!/!0yyy!	12/01/1990
19900112	!dd!/!oo!/!0yyy!	12/ 1/1990
19900112	!o!/!d!/!yyyy!	1/12/1990
19900112	! o!/! d!/! yy!	1/12/90
19900112	!1w!, !0d! !nnnn! !yy!	Mon, 12 Janu 90
19900112	!w!, !c! !dd!!e! of !n!	Monday, the 1st of January
20010306	!v! !w!, !c! !de! !! !n!, !yyyy! !z!	on Monday, the 6th of March, 2001 A.D.
20010306	!wwwwwwwwww!, !dde! !1nnnnn!, !x! !yy!	Monday , 6th Mar, year 1
19900715	!yyjjj!	91196
19900715	!w!, !n! !d!, !yyyy! (!yyjjj!)!	Monday, July 15, 1991 (91196)

The second table shows the output from a number of time templates. Column 1 shows the time input array in its HHMMSSPPPP format. Column 2 gives the time template. Column 3 shows the time output.

Value in EBCDIC Input Array	Time Template	Displayed Output
1259011234	!0t!:!0m!:!0s!.!pppp!	12:59:01.1234
125901 234	!0t!:!0m!:!0s!.!pppp!	12:59:01.0234
1259011234	!0t!:!0m!:!0s!.!p!	12:59:01.1
1259011234	!tt!:!mm!:!ss!	12:59: 1
1259011234	!t!:!m!:!s! !aaaa!	12:59:1 p.m.
125901	!0t!:!0m!:!0s!.!pppp!	12:59:01.0000
125901	!hh! !i! !mm! !k!	12 hours 59 minutes
125901	!f! !0t!:!0m! !a!	@ 12:59 hrs

Summary of Date and Time Template Control Characters

Table 3-3 summarizes the control characters used in date and time templates. The control characters are listed alphabetically. The explanations include the names of the date or time elements accessed by the control characters.

Control characters must be enclosed in a set of exclamation marks (!), as shown in the table. The template is not case sensitive; therefore, !dd! is the same as !dD!, !DD!, and !Dd!.

The date elements, such as ALT_MONTH, are detailed in "Summary of Date Elements" later in this section. The time elements, such as TIME_SUFFIX, are detailed in "Summary of Time Elements" later in this section.

To translate the literal text of the date and time elements, such as the names of the months, use the MSGTRANS utility. You can find the output message arrays in the object file SYSTEM/CENTRALSUPPORT on the release media.

Table 3-3. Date and Time Template Control Characters

Control Character	Explanation
!a!	Displays all significant characters of the text following the time value (TIME_SUFFIX). Trailing blanks are ignored.
!a . . . a!	Displays the first N characters of the TIME_SUFFIX. Trailing blanks are included. The letter N represents the number of occurrences of the letter a within the exclamation marks. N must be greater than 1. For example, !aa! displays the first two characters.
!c!	Displays the text preceding the day of the month (DOM_PREFIX).
!d!	Displays all the significant digits for the numeric representation of the day of the month (DAY_OF_MONTH). Leading zeros are ignored; for example "3" is displayed.
!dd!	Displays the DAY_OF_MONTH. Any leading zero is shown as a blank; for example, " 3" is displayed.
!0d!	Displays the DAY_OF_MONTH. Leading zeros are included; for example, "03" is displayed.
!e!	Displays the text following the day of the month (DOM_SUFFIX).
!f!	Displays the text preceding the time elements (TIME_PREFIX).
!h!	Displays the numeric values for all the significant digits of the hours of the day using the 12-hour clock (HOD_12_HRCL). Leading zeros are ignored; for example, "1" is displayed.
!hh!	Displays the HOD_12_HRCL. Any leading zero is shown as a blank. For example, " 1" is displayed.
!0h!	Displays the HOD_12_HRCL. Leading zeros are included. For example, "01" is displayed.
!i!	Displays the text following the value for hours (HOUR_SUFFIX).
!j!	Displays the numeric representation of all significant digits for the day of the year (DAY_OF_YEAR). Any leading zeros are ignored. For example, the day of year value "32" with !j! displays 32.
!j . . . j!	Displays the first N least significant digits of the DAY_OF_YEAR. Any leading zero is shown as a blank. The letter N represents the number of occurrences of the letter j within the exclamation marks. N can be 1 or greater than 1. For example, the day of year value "225" with !jjj! displays 225.
!0j . . . j!	Displays the first N+1 least significant digits of the DAY_OF_YEAR. Leading zeros are included. The letter N represents the number of occurrences of the letter j within the exclamation marks. N can be 1 or greater than 1. For example, the day of year value "45" with !0jj! displays 045.
!k!	Displays the text following the value for minutes (MIN_SUFFIX).

Table 3-3. Date and Time Template Control Characters

Control Character	Explanation
!!!	Displays the text preceding the month (MONTH_PREFIX).
!m!	Displays the numeric values of all significant digits for minutes (MINUTES). Leading zeros are ignored; for example, "1" is shown.
!mm!	Displays the MINUTES. Any leading zero is shown as a blank. For example, " 1" is shown.
!0m!	Displays the MINUTES. Leading zeros are included. For example, "01" is shown.
!n!	Displays all significant characters of the full name for the month (LONG_MONTH). Trailing blanks are ignored.
!n . . . n!	Displays the first N characters of the full name for the month (LONG_MONTH). Trailing blanks are included. The letter N represents the number of occurrences of the letter n within the exclamation marks. N must be greater than 1. For example, !nn! displays the first two characters, such as JA, FE, or MA.
!1n!	Displays all significant characters of the abbreviated name for the month (SHORT_MONTH). Trailing blanks are ignored.
!1n . . . n!	Displays the first N characters of the SHORT_MONTH. Trailing blanks are included. The letter N represents the number of occurrences of the letter n within the exclamation points. N must be greater than 1. For example, !1nn! displays the first two characters.
!2n!	Displays all significant characters of the alternate month designator (ALT_MONTH). Trailing blanks are ignored.
!2n . . . n!	Displays the first N characters of the ALT_MONTH. Trailing blanks are included. The letter N represents the number of occurrences of the letter n within the exclamation marks. N must be greater than 1. For example, !2nnn! displays the first three characters.
!o!	Displays the numeric representation of all significant digits of the month (MONTH_OF_YEAR). Leading zeros are ignored; for example, "8" is shown.
!oo!	Displays the MONTH_OF_YEAR. Any leading zero is shown as a blank. For example, " 8" is shown.
!0o!	Displays the MONTH_OF_YEAR. Leading zeros are included. For example, "08" is shown.
!p . . . p!	Displays the first most significant N digits of the partial seconds as a decimal fraction (PARTIAL_SECS). Trailing zeroes are included. The letter N represents the number of occurrences of the letter p within the exclamation marks. N can be 1 or greater than 1. For example, !ppp! displays the first three characters.
!r!	Displays the text following the value for seconds (SEC_SUFFIX).

Table 3-3. Date and Time Template Control Characters

Control Character	Explanation
!s!	Displays all significant digits of the value for seconds (SECONDS). Leading zeros are ignored; for example, "5" is shown.
!ss!	Displays the SECONDS. Any leading zero is shown as a blank. For example, " 5" is shown.
!0s!	Displays the SECONDS. Leading zeros are included. For example, "05" is shown.
!t!	Displays all significant digits for the numeric values of the hours when a 24-hour clock is used (HOUR_OF_DAY). Leading zeros are ignored; for example, "5" is shown.
!tt!	Displays the HOUR_OF_DAY. Any leading zero is shown as a blank. For example, " 5" is shown.
!0t!	Displays the HOUR_OF_DAY. Leading zeros are included. For example, "05" is shown.
!v!	Displays the text preceding the day of the week (DOW_PREFIX).
!w!	Displays all significant characters of the full name for the day of the week (LONG_DOW). Trailing blanks are ignored.
!w . . . w!	Displays the first N characters of the LONG_DOW. Trailing blanks are included. The letter N represents the number of occurrences of the letter w within the exclamation points. N must be greater than 1. For example, !www! displays the first three characters.
!1w!	Displays all significant characters of the abbreviated name for the day of the week (SHORT_DOW). Trailing blanks are ignored.
!1w . . . w!	Displays the first N characters of the SHORT_DOW. Trailing blanks are included. The letter N represents the number of occurrences of the letter w within the exclamation marks. For example, !1ww! displays the first two characters.
!x!	Displays the text preceding the year (YEAR_PREFIX).
!1x!	Displays the first significant character of the YEAR_PREFIX.
!y . . . y!	Displays the first N least significant digits of the numeric representation for the year (YEAR). Any leading zero is shown as a blank. The letter N represents the number of occurrences of the letter y within the exclamation marks. N can be 1 or greater than 1. For example, for the year 1990, !yy! displays only 90.
!0y . . . y!	Displays the first N+1 least significant digits of the YEAR. Leading zeros are included. The letter N represents the number of occurrences of the letter y within the exclamation marks. N can be 1 or greater than 1.
!z!	Displays the text following the numeric representation for the year (YEAR_SUFFIX).

Summary of Date Elements

Table 3-4 summarizes the date elements used by the date control characters. The table also specifies the control character or series of control characters that is associated with the element and the ASeriesNative default values for the element.

The elements are listed alphabetically. Examples of templates and the output they might produce are provided under “Examples of Date and Time Templates” earlier in this section.

To translate the literal text specified by the date elements, such as the names of the months, use the MSGTRANS utility. You can find the message arrays in the array CNV_MSGS in the object file SYSTEM/CENTRALSUPPORT on the release media.

Table 3-4. Date Elements

Element	Use	Control Character	ASeriesNative Values
ALT_MONTH	Alternate month designator	2n	I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII
DAY_OF_MONTH	Numeric representation for the day of the month	d	! 0o ! / ! 0d ! / ! yyyy !
DAY_OF_WEEK	Index that determines the values for the day of the week elements	Not applicable	01 indexes Monday, 02 indexes Tuesday, 03 indexes Wednesday, 04 indexes Thursday, 05 indexes Friday, 06 indexes Saturday, and 07 indexes Sunday
DAY_OF_YEAR	Numeric representation for the day of the year	j	1 through 356
DOM_PREFIX	Text preceding the day of the month	c	the
DOM_SUFFIX	Text following the day of the month	e	st, nd, rd, and th
DOW_PREFIX	Text preceding the day of the week	v	on

Table 3-4. Date Elements

Element	Use	Control Character	ASeriesNative Values
LONG_DOW	Full name for the day of the week	w	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday
LONG_MONTH	Full name for the month	n	January, February, March, April, May, June, July, August, September, October, November, and December
MONTH_OF_YEAR	Numeric representation for the month	o	01 through 12
MONTH_PREFIX	Text preceding the month	l	of
SHORT_DOW	Abbreviated name for the day of the week	1w	Mon, Tues, Wed, Thurs, Fri, Sat, and Sun
SHORT_MONTH	Abbreviated name for the month	1n	Jan, Feb, Mar, Apr, May, June, July, Aug, Sept, Oct, Nov, and Dec
YEAR	Numeric representation for the year	y	0 through 9999
YEAR_PREFIX	Text preceding the year	x	year
YEAR_SUFFIX	Text following the numeric representation for the year	z	ad

Summary of Time Elements

Table 3-5 summarizes the time elements used by the time control characters. The table also specifies the control character or series of control characters that is associated with the element and the ASeriesNative default values for the element.

The elements are listed alphabetically. Examples of templates and the output they might produce are provided under “Examples of Date and Time Templates” earlier in this section.

To translate the literal text specified by the time elements, such as the suffix *minutes*, use the MSGTRANS utility. You can find the message arrays in the array CNV_MSGS in the object file SYSTEM/CENTRALSUPPORT on the release media.

Table 3–5. Time Elements

Element	Use	Control Character	ASeriesNative Values
HOD_12_HRCL	Numeric values for the hours when a 12-hour clock is used	h	01 through 12
HOUR_OF_DAY	Numeric values for the hours when a 24-hour clock is used	t	00 through 23
HOUR_SUFFIX	Text following the value for hours	i	hours
MINUTES	Numeric values for minutes	m	00 through 59
MIN_SUFFIX	Text following the value for minutes	k	minutes
PARTIAL_SECS	Partial seconds as a decimal fraction	p	0000 through 9999
SECONDS	Numeric values for seconds	s	00 through 59
SEC_SUFFIX	Text following the value for seconds	r	seconds
TIME_PREFIX	Text preceding time elements	f	@
TIME_SUFFIX	Text following the value for time	a	a.m. or p.m.

Understanding Monetary and Numeric Templates

The following text explains the components of monetary and numeric templates and the way they are processed.

Processing Numeric and Monetary Templates

Numeric templates determine the way numeric values are to be expanded into a formatted string. Monetary templates determine the way monetary values are to be expanded into a formatted string. The minimum acceptable numeric value or monetary value is $1 \times 10^{-(11)}$. The largest value is the maximum allowable single-precision real number.

Numeric and monetary templates contain

- A series of control characters
- Literal text
- A numeric value placement symbol (#)

Except for the currency control character (c), both numeric and monetary templates can process the same control characters. Only a monetary template can process the currency control character.

The general characteristics of control characters and literal text are described under “Control Characters” and “Literal Text” earlier in this section. Specifics that relate to numeric and monetary templates only are discussed in the following paragraphs.

Associating Symbols with Numeric and Monetary Control Characters

Each numeric and monetary control character must have an associated symbol or symbols. These symbols can be specified through symbol definition lists in the template or by retrieving values in the SYSTEM/CONVENTIONS file.

Using Symbol Definition Lists

Both numeric and monetary control characters can be followed immediately by a symbol definition list. The symbol definition list contains one or more symbols that are edited into a formatted string. Symbol definition lists are enclosed in square brackets ([]). For example, the syntax for the decimal control character can take the form `!d[<symbol>!]`. The decimal template `!d[.]!` outputs a period (.) as the decimal symbol.

Each symbol defined in the symbol definition list can be from 1 to 127 characters in length. However, all symbols defined in a single convention are stored in a common storage area that cannot exceed 150 characters.

You can use hexadecimal code to represent symbols that cannot be represented in the standard ASeriesEBCDIC coded character set. The hexadecimal notation is the same as for the ALGOL language. For example, the Japanese yen symbol is defined as code position 05/11 in the Latin1EBCDIC coded character set. The template `!c[48"5B"]!` defines

the yen symbol. (The Latin1EBCDIC coded character set must be in use on the system when the yen symbol is needed.)

Retrieving Symbols from the SYSTEM/CONVENTIONS File

If a control character is not followed immediately with a symbol definition list, the symbol is retrieved from the SYSTEM/CONVENTIONS file. For example, you can use the template !c! to retrieve the national currency symbol from the SYSTEM/CONVENTIONS file.

Note: This feature is available only when a program calls one of the CENTRALSUPPORT template procedures in which the template is passed as a parameter. The calling program must create a new template and specify an existing convention name.

Placing Blanks in Numeric and Monetary Templates

While blanks can be used to improve readability, you cannot include blanks in symbol definition lists, that is, within the square brackets ([]). For example, you cannot create the template !c[\$]!. However, you can create the template !c[\$] d[.] t[.,0,3]!.

Performing Straight Conversions of Numeric Values

If the numeric template does not contain any control characters, the CENTRALSUPPORT procedure converts numeric values into formatted strings without any editing. For example, without a numeric template, the value 1,234.87 is output as 123487.

Placing Literal Text in Numeric and Monetary Templates

Literal text cannot be used

- Between a control character and the start of its symbol definition list, that is, between the control character and the left bracket ([)
- Inside the symbol definition list, that is, between paired brackets ([])

To output literal text at the end of the output, place the literal text after the closing exclamation mark (!). Then end the template with another exclamation mark. For example, the template !T[.,0,3]D[.]#!<CR>! causes the amount 1024.95 to be output as 1,024.95<CR>.

Literal text is copied verbatim from a template to a formatted string. Blanks appearing in literal text are copied exactly.

Using the Numeric Value Placement Symbol

Monetary and numeric templates are always processed from left to right. The numeric value placement symbol (#) shows the relative position of the numeric value to the formatting and editing values.

Understanding Templates in Convention Definitions

Formatting defines arrangement and order. Editing defines specifics. For example, specifying that a numeric value have a thousands separator is a formatting definition. Using a comma (,) as the separator is an editing definition. Another example of formatting and editing is seen in the output from the template !P[<DB>] # N[<CR>]!. The value 1000 is edited as <DB>1000 when the value is positive and 1000<CR> when the value is negative.

The control characters C, N, and P are position sensitive with respect to the numeric value placement symbol. For example, !c[\$]#! results in \$12345, while !#d[.]c[\$]! results in 12345\$. All the other control characters are not position sensitive. For example, !t[,:0,3]#! and !#t[,:0,3]! both result in 2,200.

Control Characters for Numeric and Monetary Templates

The six control characters associated with numeric and monetary amounts are shown in Table 3-6.

Table 3-6. Control Characters for Numeric and Monetary Templates

Control Character	Element Controlled	Valid Syntaxes
c	Currency control character	!c[<symbol>]! !1c! !c!
d	Decimal control character	!d[<symbol>]! !d! !d[c[<symbol>]]!
e	Enclosure for negative values	!e[<left enclosure>,<right enclosure>]!
n	Negative sign	!n[<symbol>]! !n!
p	Positive sign	!p[<symbol>]! !p!
t	Thousands separator	!t[<symbol>:<mindigits>,<group1>,...<groupn>]! !t!

Currency Control Character

The currency control character (c) can be used to define the national currency symbol. It can also be used to indicate that the currency symbol is to replace the decimal symbol or that the international currency notation is to be used.

The syntax `!c[<symbol>]!` defines the national currency symbol. For example, the monetary template `!c[$]!` defines a dollar sign as the national currency symbol.

You can retrieve the national currency symbol from the SYSTEM/CONVENTIONS file by using the syntax `!c!`. For more information, read "Retrieving Symbols from the SYSTEM/CONVENTIONS File" earlier in this section.

To retrieve the international currency notation, use the syntax `!1c!`. The number 1 preceding the control character signals a CENTRALSUPPORT procedure to use the international notation currency symbol in the output. Because no symbol definition list exists, the CENTRALSUPPORT procedure retrieves the symbol from the SYSTEM/CONVENTIONS file as explained in "Retrieving Symbols from the SYSTEM/CONVENTIONS File" earlier in this section. For example, the monetary template `!N[-]1CT[;:0,3]D[.]#!` causes the value 100000 to be edited as USD 1,000.00 where USD is the international currency notation for the ASeriesNative convention.

To use the currency control character to define the decimal symbol, use the currency control character and its symbol definition list as the symbol definition list for the decimal symbol. The syntax is `!#d[c[<symbol>]]!`. For example, `!#d[c[$]]t[;:0,3]!` outputs the value 125.00 as 125\$00. The syntax `d[c]` does not retrieve the currency symbol from the SYSTEMS/CONVENTIONS file. Instead, this syntax causes the character "c" to be used as the decimal symbol.

Decimal Control Character

The decimal control character (d) defines the decimal symbol in both numeric and monetary amounts.

The syntax `!d[<symbol>]!` explicitly defines the decimal symbol. For example, the template `!d[.]!` defines a period as the decimal symbol.

You can retrieve the decimal symbol from the SYSTEM/CONVENTIONS file by using the syntax `!d!`. For more information, read "Retrieving Symbols from the SYSTEM/CONVENTIONS File" earlier in this section.

Enclosure Control Character

The enclosure control character (e) defines the pair of symbols that encloses a negative value in both numeric and monetary amounts.

The syntax !e[<left enclosure symbol>,<right enclosure symbol>! explicitly defines the enclosure symbols. The left enclosure symbol precedes the first digit of an edited value. The right enclosure symbol follows the last digit of an edited value. Both a left and a right enclosure symbol must be specified. For example, the template !e[<,>! causes the negative value -288 to be edited as <288>.

Negative Sign Control Character

The negative sign control character (n) defines the negative sign in both numeric and monetary amounts.

The syntax !n[<symbol>! explicitly defines the negative sign. For example, the template !n[-]! defines a minus sign as the negative sign.

You can retrieve the negative sign from the SYSTEM/CONVENTIONS file by using the syntax !n!. For more information, read "Retrieving Symbols from the SYSTEM/CONVENTIONS File" earlier in this section.

A value of zero is not output with a negative sign.

Note: The e and n control characters are mutually exclusive. If they both appear in the same template, whichever occurs last takes precedence.

Positive Sign Control Character

The positive sign control character (p) defines the positive sign in both numeric and monetary amounts.

The syntax !p[<symbol>! explicitly defines the positive sign. For example, the template !p[+]! defines a plus sign as the positive sign.

You can retrieve the positive sign from the SYSTEM/CONVENTIONS file by using the syntax !p!. For more information, read "Retrieving Symbols from the SYSTEM/CONVENTIONS File" earlier in this section.

A value of zero is not output with a positive sign.

Thousands Separator Control Character

The thousands separator control character (t) defines the symbol used to separate groups of numbers to the left of the decimal place in monetary or numeric values. The symbol definition list can contain a thousands separator symbol, the number of digits that must be present before using the thousands separator, and the number of digits between each use of the thousands separator.

Understanding Templates in Convention Definitions

To explicitly define the thousands parameters, use the following syntax:

```
!t[<symbol>:<mindigits>,<group1>,<group2>...<groupn>]!
```

The following table explains the elements in the symbol definition list:

Element	Explanation
<symbol>	The symbol used as the separator.
<mindigits>	The minimum number of digits that must be present, to the left of the decimal place, before inserting the thousands separator. If <mindigits> is 0 or omitted, the thousands separator is inserted according to the group specifications.
<groupn>	An unsigned integer specifying the number of digits between the placement of the thousands separator. The first group designator, <group1>, represents the first group to the left of the decimal symbol. The second group designator, <group2>, represents the second group to the left of the decimal symbol. The <groupn> designator is the nth group to the left of the decimal symbol. If no number for <group> is specified, then a value of 0 is assumed; the thousands separator is not used. If only <group1> is specified, then the thousands separator is placed between each group of <group1> digits. For example, if group1 is 3 and the thousands separator is a blank, the integer 123456789 is output as 123 456 789.

You must specify the <mindigits> element or the <groupn> element or both. You cannot use this command without specifying at least one of these elements.

If the number of digits to the left of the decimal symbol is greater than all the preceding group specifications, then the last group specification is applied repeatedly. For example, the template !C[\$] T[,:0,3]D[.]#! causes the amount 1234567.89 to be edited as 1,234,567.89. The group specification of 3 is applied repeatedly.

Understanding Templates in Convention Definitions

When both a minimum digits variable and a group variable are specified, the following editing occurs:

- If the value of the minimum digits variable is less than or equal to the value specified in the group variable, then the placement of thousands separators is governed by the group variable.
- If the value of the minimum digits variable is greater than the value specified in the group variable, and the number of digits in the output is less than the minimum digits variable, then no thousands separators are placed in the edited output.
- If the value of the minimum digits variable is greater than the value specified in the group variable, and the number of digits in the output is larger than the minimum digits variable, then thousands separators are placed in the edited output according to group specifications.

For example, a template such as `!c[$]t[.:5,3]d[.]#!` outputs \$1234., \$12,345., and so on. A template such as `!t[.:2,3,4]c[$]d[.]#!` outputs \$12,1234,1234,123,12.00.

Examples of Numeric and Monetary Templates

The following templates format and edit the value 527736.29. The templates that contain a currency control character are monetary templates.

Template	Output
<code>!n[-]c[\$]t[.:0,3]d[.]#!</code>	\$527,736.29
<code>!#n[-]d[.]!</code>	527736.29
<code>!c[\$]#n[-]t[.:0,3]d[.,]!</code>	\$527.736,29
<code>!t[.:0,3]d[.,]c[\$]#n[-]!</code>	\$527 736,29
<code>!p[+]t[.:0,3]d[c[\$]]#!</code>	+527,736\$29

Summary of Monetary and Numeric Template Control Characters

Table 3-7 summarizes the control characters used in monetary and numeric templates. Unlike the date and time control characters, which reference output arrays for their literal text, the monetary and numeric templates use symbol definition lists within the control character syntax.

The symbol definition list is noted between a set of brackets. For example, the following specification assigns a dollar sign (\$) as the currency symbol:

`c[$]`

The monetary and numeric templates consist of a series of control characters with their respective symbol definition lists, other literal text, and the numeric placement value symbol grouped between one set of exclamation points (!). For example, the following series of control characters could be used as a monetary template:

!c[\$]n[-]p[+]#!

The control characters are listed alphabetically. The explanations include the default literal text for the ASeriesNative convention.

The template is not case sensitive; therefore, c and C are considered the same control character.

Note: *The currency control character is valid only in monetary templates.*

Table 3-7. Monetary and Numeric Template Control Characters

Control Character	Explanation
#	Designates the placement of the amount in relation to the negative sign, positive sign, and currency symbol.
c[]	Designates the symbol used as a currency symbol. For the ASeriesNative convention, the currency symbol is a dollar sign (\$), designated as c[\$].
1c	Designates that the international currency symbol is to be retrieved from the SYSTEM/CONVENTIONS file.
c	Designates that the national currency symbol is to be retrieved from the SYSTEM/CONVENTIONS file.
d[]	Designates the symbol used as a decimal symbol. For the ASeriesNative convention, the decimal symbol is a period (.), designated as d[.].
d[c[]]	Designates that the currency symbol is to be used in place of the decimal symbol. This format is referred to as an infix currency symbol.
d	Designates that the decimal symbol is to be retrieved from the SYSTEM/CONVENTIONS file.
e[]	Designates the enclosure symbols for a negative value. Both the right and left enclosure symbol must be specified. For example, e[<,>] specifies that a negative amount is enclosed in angled brackets.
n[]	Designates the symbol used to show a negative amount. For the ASeriesNative convention, the negative sign is a minus sign (-), designated as n[-].
n	Designates that the negative sign is to be retrieved from the SYSTEM/CONVENTIONS file.
p[]	Designates the symbol used to show a positive amount. This control character is not part of the ASeriesNative convention.
p	Designates that the positive sign is to be retrieved from the SYSTEM/CONVENTIONS file.

Table 3-7. Monetary and Numeric Template Control Characters

Control Character	Explanation
t[]	<p>Designates the symbol used to separate thousand groupings, indicates when first to use the symbol, and specifies the number of digits between placements of the separator.</p> <p>For the ASeriesNative convention, the thousands grouping is coded as t[,0,3]. This means that the thousands separator is a comma (.). The comma is used for every amount equal to or greater than 1,000. Every subsequent grouping of three numbers is separated by a comma, such as 1,999,888,345.</p>
t	<p>Designates that the thousands separator is to be retrieved from the SYSTEM/CONVENTIONS file.</p>

Translating Literal Text Used in Conventions

The predetermined text specified by the control characters, such as day and month names or prefixes and suffixes, is stored in output message arrays in the object file SYSTEM/CENTRALSUPPORT. For example, for the days of the week (DOW) elements, the SYSTEM/CENTRALSUPPORT file contains the following arrays:

DAY_OF_WEEK	DOW_PREFIX	LONG_DOW	SHORT_DOW
01	on	Monday	Mon
02	on	Tuesday	Tue
03	on	Wednesday	Wed
04	on	Thursday	Thurs
05	on	Friday	Fri
06	on	Saturday	Sat
07	on	Sunday	Sun

Use the MSGTRANS utility to translate messages in output message arrays to other languages.

If, for a particular language, the custom or grammar does not require one of the elements, such as the DOW_PREFIX *on*, the translated element must be null. Read the *MSGTRANS Operations Guide* for details on the way to translate the output message array.

Section 4

Establishing an MLS Environment

This section describes the system requirements for establishing the MLS environment. The text includes the following:

- The installation requirements for the MLS environment
- A description of the effects of establishing the MLS environment
- Guidelines and procedures for setting default values for the MLS options

Installing the MLS Environment on Your System

To install the MLS environment, perform the following steps:

1. Install and configure your internationalized data communications environment. Refer to Section 7, "Configuring Unisys Products for Internationalization," later in this manual.
2. Install and configure the MLS software. Refer to the *Migration Guide* for information about installing the release. You can install the software for an MLS environment by using the Simple Installation (SI) program or by manually performing the installation tasks.
3. Copy the following files to your system:

File Name	Contents
SYSTEM/CENTRALSUPPORT	CENTRALSUPPORT library
SYSTEM/CCSFILE	Standard coded character set and ccsversion definitions
SYSTEM/CONVENTIONS	Standard and custom convention definitions
SYSTEM/SLICESUPPORT	Locales defined for C programs. This file is required only when you are using C programs in an MLS environment. Refer to Section 8, "Using C Programs in an MLS Environment," for more information.

4. Copy any translated help files or other files that have been translated to the system (for example, *SYSTEM/MARC/MENUGRAPH/FRANCAIS).

Establishing an MLS Environment

5. Define the SYSTEM/CENTRALSUPPORT file as a system library with the name CENTRALSUPPORT.

If you . . .	Then . . .
Use the Simple Installation (SI) program to copy and install the software release	The library is defined for you.
Do not use the SI program	You must make SYSTEM/CENTRALSUPPORT a system library by using the SL (Support Library) system command. For example, SL CENTRALSUPPORT = *SYSTEM/CENTRALSUPPORT ON DISK

6. Put the CCSFILE and CONVENTIONS files on the same family as the CENTRALSUPPORT library. The CENTRALSUPPORT library checks for the availability of the CCSFILE and CONVENTIONS files during system initialization in the following order:
- <Usercode of CENTRALSUPPORT library> ON <family of CENTRALSUPPORT library>
 - <No usercode> ON <family of CENTRALSUPPORT library>

When using ccsversions or the SYSTEM/CCSFILE with KEYEDIOII files or Enterprise Database Server databases across multiple hosts, the SYSTEM/CCSFILE must be the same on all of the hosts. If the CCSFILES differ, you might encounter data corruption in the KEYEDIOII or database files.

Installing the ASERIESNATIVE Environment

If you choose not to use the internationalization features, your system runs under the default ccsversion, ASeriesNative. The ASeriesNative ccsversion has a ccsversion number of 0 (zero) and corresponds exactly to the processing done in a non-internationalized environment. In addition, the system default language is set to English, and the system default convention is set to ASeriesNative.

You must have the SYSTEM/CENTRALSUPPORT file present on the system; however, the SYSTEM/CCSFILE and SYSTEM/CONVENTIONS files are not required.

Installing the **SYSTEM/CONVENTIONS** File for the First Time

You can install the SYSTEM/CONVENTIONS file by using the Simple Installation (SI) program. To install the conventions file the first time, follow the instructions for using the SI program in the *Simple Installation Operations Guide*.

Installing Updates to the **SYSTEM/CONVENTIONS** File

You can use the SI program to install a new release of the SYSTEM/CONVENTIONS file and to include your existing custom conventions in the new file. The SI program enables you to create an updated SYSTEM/CONVENTIONS file that consists of the following:

- The conventions provided in the new SYSTEM/CONVENTIONS file
- The custom conventions that you migrate from your previous version of the SYSTEM/CONVENTIONS file

To install and update a SYSTEM/CONVENTIONS file, you must observe one precaution. When you have custom conventions that you want to migrate to the new file, you must ensure that the SI program does not copy the new SYSTEM/CONVENTIONS file over the existing file. If the SI program copies the new file over the old file, all your custom conventions are destroyed.

To prevent the new SYSTEM/CONVENTIONS file from being copied over your existing file, perform one or more of the following steps as you run the SI program:

- Copy the new SYSTEM/CONVENTIONS file under a usercode other than the usercode of the currently running SYSTEM/CENTRALSUPPORT file. You can designate the usercode on the INSTALL screen.
- Copy the SYSTEM/CONVENTIONS file to a family name other than the name used by the currently running SYSTEM/CENTRALSUPPORT file. You can designate the family name on the INSTALL screen.
- Insert the HMP or SSR number in the SYSTEM/CONVENTIONS file title when the file is copied from the release media. The HMP or SSR number, which you designate on the CONTROL screen, indicates the level of the release you are installing. Examples of these numbers are HMP 1.0 and SSR 43.2.

For example, assume the current System Library specification is as follows:

```
SL CENTRALSUPPORT = *SYSTEM/CENTRALSUPPORT ON DISK
```

In this case, you would not want to copy the conventions file as *SYSTEM/CONVENTIONS ON DISK.

Establishing an MLS Environment

During installation, the SI program copies the SYSTEM/CONVENTIONS file from the release media to the usercode and family name that you designate on the INSTALL screen. The copied file is named using one of the following formats:

```
<usercode from Simple Install>SYSTEM/<software release number>  
/CONVENTIONS ON <family name from Simple Install>
```

```
<usercode from Simple Install>SYSTEM/CONVENTIONS ON  
<family name from Simple Install>
```

After the SI program copies and installs the conventions file, it starts a WFL job to update the file. This job merges your custom conventions into the SYSTEM/CONVENTIONS file that you copied from the release media. The job saves the updated conventions file under a new file name, using the same usercode and family name designated for the current SYSTEM/CONVENTIONS file used by the CENTRALSUPPORT library. The new file title has the following form:

```
<usercode>SYSTEM/CONVENTIONS/<software release number><major ID>  
ON <family name>
```

The software release number in this name indicates the release of the installed SYSTEM/CONVENTIONS file. The major ID is an internal conventions file number that identifies the file. An example of the file name is *SYSTEM/CONVENTIONS/432 ON DISK.

At the next halt/load, the CENTRALSUPPORT library looks for an updated conventions file. If an updated file is available, the CENTRALSUPPORT library saves the old file conventions under a new name. The CENTRALSUPPORT library then changes the name of the updated conventions file to SYSTEM/CONVENTIONS and begins using this updated file.

The CENTRALSUPPORT library saves the old conventions file under the name:

```
<usercode>SYSTEM/CONVENTIONS/<old software release number> ON <family name>
```

The old software release number in this name indicates the release under which the old conventions file was created.

Conventions File Update Report

The program creates a printer backup file as a record of the update process. This file contains the following information:

- The names of the three conventions files used in the update: the file on the release media, the file currently being used, and the updated conventions file
- A list of the conventions in the new release of the conventions file
- A list of custom conventions that were merged into the new conventions file
- The total number of system-provided and custom conventions copied into the updated file

You can find this file on the backup pack under the mix number of the WFL job. Do not confuse the backup file for the conventions file update report with the backup file for the SI report. Both reports are BD files under the same mix number. The last node of the conventions update report file title is 001LST.

Example

An example of the conventions file update report follows:

```
Conventions File Upgrade Utility version 41.2.281.053

Conventions file (released version): *SYSTEM/412/CONVENTIONS ON DISK.
Conventions file ('in use' version): *SYSTEM/CONVENTIONS ON DISK.
Conventions file (upgraded version): *SYSTEM/CONVENTIONS/412 ON DISK.

UNISYS conventions:
ASERIESNATIVE
Netherlands
Belgium
EuropeanStandard
Norway
Sweden

Customized conventions:
Dutch
Militarytime

7 conventions copied
```

Initializing the CENTRALSUPPORT Library

During system initialization, the CENTRALSUPPORT library is one of the first components to be initialized. If you have established an internationalized system by setting the `ccsversion` or `convention` features, then you might encounter the following conditions if your system is not configured correctly. If one of these conditions occurs, the system continues to initialize.

Condition 1

You set the `ccsversion` or `convention` to a valid value other than `ASeriesNative`, and then reinitialize the system. Since the system is now internationalized, the `SYSTEM/CCSFILE` or `SYSTEM/CONVENTIONS` file is required. If the file, `SYSTEM/CCSFILE`, is not available or if the CENTRALSUPPORT library encountered an error in trying to read the file, then you would see this message:

```
AX: *SYSTEM/CCSFILE ON DISK NOT OPEN F.AVAILABLE = 2.  
??COPY or CHANGE file and AX RETRY or AX SETASN to return to  
default setting.
```

If the `SYSTEM/CONVENTIONS` file is missing, then that file name appears in the message.

The integer that follows `F.AVAILABLE=` indicates the reason that the attempt to open the file failed. All possible results are documented in the *File Attributes Programming Reference Manual*, under the description of the `AVAILABLE` attribute. The most common result is 2, which indicates that the file is missing. To attempt to open the file again, you can copy a new file in and enter the mix number followed by `AX RETRY`. Alternatively, you can enter the mix number followed by `AX SETASN`, which restores the system to `ASeriesNative` so that system initialization can continue, but without internationalization features.

Condition 2

You set the `ccsversion` or `convention` to a valid value other than `ASeriesNative` and reinitialize the system. If during reinitialization the `ccsversion` or `convention` value is no longer valid, the following message appears:

```
AX: The system default CCSVERSION is invalid. Use AX <CCSVERSION  
name> to enter a new ccsversion or AX ASERIESNATIVE to use  
default settings.
```

If the system default convention is invalid, the word `CONVENTION` appears in the message instead of the word `CCSVERSION`.

This situation would occur if the original `SYSTEM/CCSFILE` file was replaced with a `SYSTEM/CCSFILE` that did not have the new `ccsversion` in it. You can enter the mix number followed by `AX` and the name of another valid `ccsversion`, or you can enter the mix number followed by `AX ASERIESNATIVE` to return the system to a non-internationalized state.

Localizing the Operator Display Terminal (ODT) for Use with a Personal Computer Environment

On systems such as the Micro A that can run a variety of operating systems on the PC, you can communicate localized data between the operating system on the PC and the ODT. To do so, you must establish a coded character set—also referred to as a code page on PCs—for each environment. You must choose a combination of character sets that enables you to map data from one character set to another by way of coded character set mapping tables provided on the release media.

Use the following process to establish communication between the host and the PC environment.

Table 4-1. Communicating Between the Host and PC Environment

Stage	Description
1	Select a code page in the PC environment as described in the installation guide for the system. For example, the procedure for selecting a code page with the OS/2 operating system is described in the Micro A 825 OS/2 System Installation Guide.
2	Initialize the system to the host environment as described in the installation guide for the system.
3	<p>Choose a host coded character set by entering the command <code>SYSTEMOPTIONS CCSVERSION <ccsversion></code>. The coded character set you choose must be compatible with the code page in the PC environment. To be compatible, the character sets must support most of the same characters.</p> <ul style="list-style-type: none"> • Refer to the <code>SYSTEMOPTIONS</code> command later in this section for more information about setting the <code>ccsversion</code>. • Refer to Table C-1, "Ccsversions in the Current Release," for a list of the available <code>ccsversion</code>s and their base coded character sets. • Refer to Table B-2, "Coded Character Set Mapping Tables in the Current Release," for a description of the combinations of character sets that are supported through the use of coded character set mapping tables.
4	Halt/load the system to establish the new <code>ccsversion</code> as the system default.
5	Verify that the change took effect by entering the system command <code>OL SC1</code> and checking to see that the CCS number is loaded.
6	Optionally, translate the host messages to the appropriate language. Either contact your local support representative for a localized version of the MCP messages, or translate the messages yourself by using the <code>MSGTRANS</code> utility.

Messages

If the system cannot successfully establish communication between the PC and the host, it notifies you during system initialization by displaying a message at the ODT. The system then loads the default EBCDIC to ASCII translate table and proceeds with the initialization process.

WARNING: THERE IS NO EBCDIC MATCH TO THE ODT CCS NUMBER
AVAILABLE. NO TRANSLATE TABLE WILL BE LOADED.

WARNING: THERE IS NO MATCHING ODT CHARACTER SET NUMBER AVAILABLE.
NO TRANSLATE TABLE WILL BE LOADED.

You received one of the previous messages because you chose a code page that is not supported or one for which no compatible host coded character set exists.

Choose another code page and repeat the steps to establish communication between the PC and the ODT.

WARNING: THE ODT'S CCS NUMBER IS INCOMPATIBLE WITH THE CCS
OF THE SYSTEM DEFAULT CCSVERSION. NO TRANSLATE TABLE
WILL BE LOADED.

The code page you chose is supported, but the host coded character set you selected (with a ccsversion) is incompatible with that code page.

Choose a different ccsversion and halt/load the system, or choose a different code page and repeat all the steps for establishing communication between the PC and the host.

TRANSLATE TABLE CANNOT BE LOADED TO THE ODT.

The code page and the host coded character set are compatible, but the SYSTEM/CCSFILE does not contain a coded character set mapping table between the two character sets.

Refer to Table 9-2, "Coded Character Set Mapping Tables in the Current Release," to determine if a coded character set mapping table for the combination of character sets that you chose is provided.

If a coded character set mapping table . . .	Then . . .
Does exist	The *SYSTEM/CCSFILE is missing. Reload the *SYSTEM/CCSFILE.
Does not exist	Choose a different ccsversion and halt/load, or choose a different code page and repeat the steps for establishing communication between the PC and the ODT.

Setting the Ccsversions, Languages, Conventions, and HostCCS at the System Level

If you are the system administrator, a privileged user, or a person authorized to use the system console, you can use either the SYSTEMOPTIONS system command or the MARC menus and screens to set the system default values for

- Ccsversion
- Language
- Convention
- HostCCS

Understanding the Impacts of Setting the MLS Options

The selected ccsversion and language become the default settings for the system after the next halt/load. The selected convention becomes the default setting for the system immediately.

You can set the ccsversion only at the system level. Once set, the system ccsversion becomes the default for all users and all programs on the system. Programs can also access and use information from other ccsversions.

Caution

If you change the system default ccsversion to something other than its initial setting, you might cause programs that use it to operate differently. For example, the results of comparing or sorting data might be different. You would also need to reorganize your database to conform to the definitions of the new ccsversion.

Once a language is in effect, all menus, screens, messages, and help text appear in the designated language unless no localized text in that language exists. If no localized text exists, the text appears in the original language. If you change the system default language, ensure that you designate the language name consistently and that the translated files are available on the system.

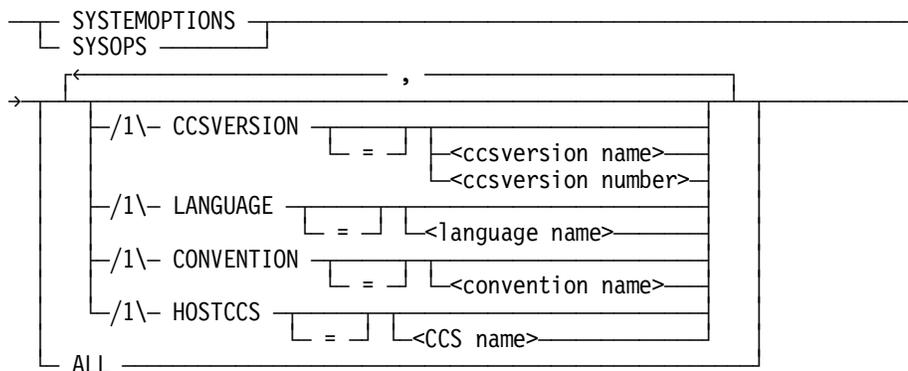
You must use the language name as it is designated in the MSGTRANS utility for the output message array to be bound to the MCP. For example, if you designate in MSGTRANS that an output message array is to be bound to the MCP using the language name Francais, then designate the language name for the localized versions of screens, messages, and help text as Francais, not French. The system does not map the name French to the files designated as Francais.

If you change the system convention, then the dates, times, and other data items formatted according to the convention might be displayed differently. Since changing the system convention does not require a halt/load, the system administrator should be cautious when making the change. A change made while internationalized programs are running could, for example, cause programs that print the date to begin suddenly printing the date in a different format because the system convention was changed.

Using the **SYSTEMOPTIONS** Command to Set the System Defaults

You can use the SYSTEMOPTIONS system command to set or display the system defaults for the three MLS options. The following diagram shows the syntax for the MLS options that use this command. You can use the abbreviation SYSOPS for SYSTEMOPTIONS. The SYSTEMOPTIONS command has some options that are for uses other than internationalization. For the complete syntax of the SYSTEMOPTIONS command, refer to the *System Commands Reference Manual*.

SYSTEMOPTIONS Command



Explanation

<ccsversion name>

The ccsversion name is a string of 1 to 17 alphanumeric characters. The name must begin with a letter. If you designate a ccsversion name that is not in the SYSTEM/CCSFILE file, an error message is displayed and the ccsversion is not changed.

<ccsversion number>

The ccsversion numbers are contained in the file, SYSTEM/CCSFILE. You can use the ccsversion number in place of the ccsversion name. If you designate a ccsversion number that is not in the SYSTEM/CCSFILE file, an error message is displayed and the ccsversion is not changed.

<language name>

The language name must begin with a letter and contain an alphanumeric string of 1 to 17 characters. The language name you enter as the system default does not have to be bound to the MCP. However, if it is not, the MCP will not recognize the language. After the system is halt/loaded, the MCP continues to display messages in the original language, even though the system language is now set to the new value.

<convention name>

The convention name must begin with a letter and contain an alphanumeric string of 1 to 17 characters. If you designate a convention that is not defined in the SYSTEM/CONVENTIONS file, an error message is displayed and the convention is not changed.

<ccs name>

The CCS names are defined in the SYSTEM/CCSFILE. Only certain CCS names are valid for the HostCCS option. Valid names are:

ARABIC20EBCDIC	LATIN1EBCDIC
ASERIESEBCDIC	LATIN2EBCDIC
CANSUPPLEBCDIC	LATIN5EBCDIC
EBCDICGB2312	LATIN9EBCDIC
EBCDICKSC5601	LATINCYRILLICEBC
IBMSWEDENEBCDIC	LATINGREEKEBCDIC
JAPANEBCDICJBIS8	LOCALEBCDIC
JAPANV24JBIS8	

SYSTEMOPTIONS SYSOPS

When you use the command with no parameters, it displays the current system default settings for ccsversion, language, and convention.

ALL

When you designate the ALL option, the command displays the current system default settings and all valid values for ccsversion, language, and convention.

Csversion

If you enter this option with no parameters, the command displays the system default ccsversion. If you specify a name or number with this option, the system default ccsversion is changed to the designated ccsversion after the next halt/load.

To display a list of the ccsversions available on your system, use the ALL option. Refer to Section 14, "Ccsversions," for more information.

LANGUAGE

If you enter this option with no parameters, the command displays the system default language. If you specify a name with this option, the system default language is changed to the designated language after the next halt/load.

To display a list of languages bound to the MCP on your system, use the ALL option.

CONVENTION

If you enter this option with no parameters, the command displays the system default convention. If you specify a name with this option, the system default convention is changed to the designated convention immediately.

To display a list of the conventions available on your system, use the ALL option. Refer to Section 15, "Standard Conventions," for more information on the standard convention definitions.

The following examples illustrate the use of the SYSTEMOPTIONS system command. The display that results from entering the command examples might include information on options that are unrelated to internationalization.

HostCCS

If you enter this option with no parameters, the command displays the system default HostCCS. If you specify a valid CCS name with this option, the system default HostCCS is changed to the designated CCS name after the next halt/load. It is only valid to change the HostCCS value if the pending CCSVERSION is ASERIESNATIVE.

Example 1

SYSTEMOPTIONS

This command displays the current values for the system options, as follows:

```
-----SYSTEM OPTIONS-----  
  
System Style: MicroA System MCP 41.18.2879  
  
The setting(s) for system options are as follows:  
  
The current CCSVERSION is ASERIESNATIVE (0).  
The current LANGUAGE is ENGLISH.  
The current CONVENTION is ASERIESNATIVE.
```

Example 2

SYSOPS ALL

This command displays the current system options and available option values, as follows:

```
-----SYSTEM OPTIONS-----  
  
System Style: MicroA System MCP 41.18.2879  
  
The setting(s) for system options are as follows:  
  
The current CCSVERSION is ASERIESNATIVE (0).  
The current LANGUAGE is ENGLISH.  
The current CONVENTION is ASERIESNATIVE.  
  
The following object(s) are available for use in each system option  
category.  
  
There are 5 CCSVERSIONS available. They are as follows:  
  
ASERIESNATIVE SWISS SPANISH  
SWEDISH1 FRANCE  
  
MCP BOUND LANGUAGE IS: ENGLISH  
  
There is 1 CONVENTION available. The CONVENTION is ASERIESNATIVE.
```

Example 3

```
SYSOPS CCSVERSION = FRANCE, CONVENTION = STANDARDTURKEY
```

This command sets a new value for the system default ccsversion and convention as follows:

```
-----SYSTEM OPTIONS-----  
  
System Style: MicroA System MCP 41.18.2879  
  
SYSTEM OPTION(S) are changed as follows:  
  
The current CCSVERSION is ASERIESNATIVE (0).  
The CCSVERSION following the next HALTLOAD will be FRANCE (35).  
System CONVENTION is being set to STANDARDTURKEY
```

Using MARC to Display and Set the System Defaults

You can also set the system defaults for the ccsversion, language, and convention options by using a series of MARC menus and screens. The sequential method of setting MLS options is described in the following text. However, if you are familiar with MARC commands, you might prefer to use the GO action to move directly to a MARC screen or menu. Type *GO <screen name>* in any Action field, substituting the name of the screen or menu for the screen name construct; for example, *GO LOCAL*. Also, you can use the SYSTEMOPTIONS command from the MARC Action field to set or display MLS options. For details on using MARC to set the MLS options, see the *MARC Operations Guide*.

Displaying the LOCAL Menu for System Localization Options

To set the MLS options, you must first display the LOCAL menu. You can access the LOCAL menu by using one of the following methods:

- From any MARC menu, type *GO SYS* in the Action field and transmit. Or, from the MARC Home menu, type *SYS* in the Choice field and transmit. Then type *LOCAL* in the Choice field and transmit. MARC displays the LOCAL menu.
- Type *GO LOCAL* in any Action field and transmit.

From the LOCAL menu, you can display the current system settings for the MLS options by typing *SS* in the Choice field and transmitting.

MARC displays the current system settings for the ccsversion, language, and convention.

Setting the System Default Ccsversion

To set the system default ccsversion, follow these steps:

1. Display the LOCAL menu.
2. Type *CCSVSN* in the Choice field of the LOCAL menu and transmit.
MARC displays the SYSTEMCCSVSN—Change System Ccsversion screen, from which you can do the following:
 - Display the current default ccsversion for the system.
 - Display the available ccsversions that can be used as the default ccsversion for the system.
 - Change the default ccsversion for the system.
Before you decide to change the ccsversion, you need to display the current ccsversion and the list of valid ccsversions. Consider carefully the impact of the change you are contemplating.
3. To change the current default ccsversion for the system, type either the name or the number of the ccsversion and transmit.

The new default ccsversion for the system becomes effective after the next halt/load.

Setting the System Default Language

To set the system default language, follow these steps:

1. Display the LOCAL menu.
2. Type *LANG* in the Choice field of the LOCAL menu and transmit.
MARC displays the SYSTEMLANGUAGE—Change System Language screen, from which you can do the following:
 - Display the current default language for the system.
 - Display the languages that are bound to the MCP.
 - Change the default language for the system.
It is a good practice to display the current language and the languages that are bound to the MCP before entering any changes.
3. To change the current default language for the system, type the name of the language in the field designated for the new system language name and transmit.
 - You must enter the language name as it is bound to the MCP. For example, if the language is bound to the MCP as *FRANCAIS*, type *FRANCAIS* rather than French.

If the language name is bound to the MCP, MARC displays the following message:

SYSTEM OPTION(S) are changed as follows:

System LANGUAGE is being set to <language>

If the language name is not bound to the MCP, MARC displays the following message:

```
SYSTEM OPTION(S) are changed as follows:
```

```
System LANGUAGE is being set to <language>
```

```
WARNING: <language> IS NOT AN MCP BOUND LANGUAGE  
The MCP bound LANGUAGE is: <previous language>
```

The selected language remains pending until the system is halt/loaded. Programs that are running or initiated before the halt/load continue to use the current language as the system default. After the halt/load, programs use the new language as the system default.

Setting the System Default Convention

To set the system default convention, follow these steps:

1. Display the LOCAL menu.
2. Type *CONV* in the Choice field of the LOCAL menu and transmit.
MARC displays the SYSTEMCONVENTION—Change System Convention screen, from which you can do the following:
 - Display the name of the current default convention for the system.
 - Display the available conventions that can be used as the default convention for the system.
 - Change the default convention for the system.

It is a good practice to display the name of the current convention and the names of the valid conventions before entering any changes.

For information on the way to display the elements in a convention definition, refer to “Viewing Convention Definitions at the System Level” or “Viewing Convention Definitions at the MARC Session Level” later in this section.
3. To change the current default convention for the system, type the name of the desired convention, as defined in the SYSTEM/CONVENTIONS file, in the field designated for the new system convention and transmit.

The new default convention for the system becomes effective immediately.

Viewing and Customizing Conventions at the System Level with MARC

System administrators, privileged users, and anyone allowed to use the system console can view, add, modify, or delete convention definitions at the system level by using the MARC menus and screens.

The ability to add, modify, and delete conventions depends on two factors: the level of the request and the type of convention. You can view convention definitions at the system and session levels. In addition to the standard conventions provided on the release media, you can create custom conventions. You can add, modify, or delete custom conventions at the system level and through an application program.

Two copies of the convention definitions reside on the system. One copy is in memory and is stored in the CENTRALSUPPORT library during initialization; the other copy is on disk and is stored in the SYSTEM/CONVENTIONS file. One or both copies can be affected when convention definitions are added, modified, or deleted. When a new, custom convention definition is added, both copies are updated. Any modification to an existing, custom convention updates the copy on disk. The copy in memory is not updated. Updating only the copy on disk ensures that programs that are running continue their execution with the unmodified convention. When the next halt/load occurs, the modification is stored both on disk and in memory. If you delete a custom convention, the order in which updates are made to the copies on disk and in memory is exactly the same as for the modification of a custom convention. Because of the order in which updates are made, the current status of the system is available from the copy in memory.

Caution

If you reload the SYSTEM/CONVENTIONS file from the release media, all custom convention definitions are lost. As a precaution, save all custom convention definitions as part of your backup procedures.

When you add a custom convention, you must ensure that the convention name is unique within the SYSTEM/CONVENTIONS file.

Refer to Section 3, "Understanding Templates in Convention Definitions," for an explanation of the way to define date, time, monetary, and numeric templates.

Displaying the CONVDF—View/Add/Modify/Delete a Convention Definition Screen

To view, add, modify, or delete a convention, you must first display the CONVDF—View/Add/Modify/Delete a Convention Definition screen.

To display the CONVDF screen, follow these steps:

1. Display the LOCAL menu.
2. Type *CONVDF* in the Choice field of the LOCAL menu and transmit.

MARC displays the CONVDF—View/Add/Modify/Delete a Convention Definition screen, from which you can do the following:

- View the elements in a convention definition.
- Define the elements to create a custom convention definition.
- Modify custom convention definitions.
- Delete custom convention definitions.

Viewing Convention Definitions at the System Level

To view a convention at the system level, follow these steps:

1. Display the CONVDF—View/Add/Modify/Delete a Convention Definition screen.
2. To view a convention, make two entries on the CONVDF screen:
 - Type the name of the convention that you want to view in the Enter the Convention Name field.
 - Type an *X* in the applicable View Convention Definition field to display the current convention definitions stored in memory, or to display copies of convention definitions that are stored on disk. The copy on disk contains any changes made since the last halt/load. After typing your choice in the field, transmit the screen.

Adding Custom Convention Definitions

1. Display the CONVDF—View/Add/Modify/Delete a Convention Definition screen.
2. To add a new convention, make two entries on the CONVDF screen:
 - Type the proposed name of the convention in the Enter the Convention Name field. The name must be unique within the SYSTEM/CONVENTIONS file.
 - Type an X in the Add New Convention Definition field and transmit.
MARC displays the ADDCNV—Add Convention screen.
3. From the ADDCNV screen, you can enter values for all the data items that make up the convention, except the convention name. The convention name is specified on the previous CONVDF screen.

Be sure that you enter a value for each element. Otherwise, you encounter processing problems when you use the convention. If a data item has a pre-filled value, you can accept the value or change it.

Refer to Section 15, “Standard Conventions,” for a list of the contents of each of the standard convention definitions.

Transmit the screen when you have finished entering values for all data items.

The new convention definition is added immediately. It can be designated as the default convention at the system, session, usercode, or task level. It can also be accessed by an application program.

Remember that you must maintain any custom convention definitions. If the file SYSTEM/CONVENTIONS is reloaded from the release media, you must reenter all custom convention definitions. As a precaution, save all custom convention definitions as part of your backup procedures.

Modifying Custom Convention Definitions

Caution

Modifying a convention might cause unexpected program results when the program is rerun with the new convention.

To modify a custom convention at the system level, follow these steps:

1. Display the CONVDF—View/Add/Modify/Delete a Convention Definition screen.
2. To modify a convention, make two entries on the CONVDF screen:
 - Type the name of the convention in the Enter the Convention Name field. The name must be unique within the SYSTEM/CONVENTIONS file.
 - Type an X in the Modify Convention Definition field and transmit.
MARC displays the MODCNV—Modify Convention screen.

3. From the MODCNV screen, you can change values for any or all of the data items that make up the convention, except the convention name.

Be sure that a valid value is associated with each data item. Otherwise, you might encounter processing problems when you use the convention. If a data item has a pre-filled value, you can accept the value or change it.

Consult Section 15, "Standard Conventions," for an explanation of the control characters for the date, time, monetary, and numeric templates.

Transmit the screen when you have finished entering values.

The modifications are stored in the SYSTEM/CONVENTIONS file immediately. However, to prevent concurrent processing problems, the unmodified convention remains in memory until the next halt/load. The new definition of the convention becomes effective after the next halt/load.

Deleting Custom Convention Definitions

To delete a custom convention at the system level, follow these steps:

1. Display the CONVDF—View/Add/Modify/Delete a Convention Definition screen.
2. To delete a convention, make two entries on the CONVDF screen:
 - Type the name of the convention in the Enter the Convention Name field. The name must be unique within the SYSTEM/CONVENTIONS file.
 - Type an X in the Delete Convention Definition field and transmit.

The deletion is made to the SYSTEM/CONVENTIONS file immediately. However, programs that are currently running and using the deleted convention can continue to access the convention because the convention is deleted from memory only after the next halt/load.

Setting a Default Language and Convention at the Usercode Level

When you set the language and convention at the system level, these become the defaults for all users on the system. If a user requires a different language or convention, you can establish defaults that are associated with the usercode of the individual.

You can set the LANGUAGE and CONVENTION options for individual users in the USERDATAFILE file. The option is then set every time the user logs on. For more information about this file and the LANGUAGE and CONVENTION options, refer to the *Security Administration Guide*.

Viewing and Setting the Language and Convention at the Session Level

Any user can set a language or a convention for a specific session at the session level. The impact of setting a session language varies depending on whether you are in a MARC session or a CANDE session. Setting a session convention has the same impact in MARC and in CANDE. In any case, setting a language or a convention for your MARC session does not impact your CANDE session, nor does setting a language for a CANDE session impact your MARC session.

When you designate a language for your MARC session, MARC sets both a MARC session language and a Transaction Server application session language. The impact of setting a MARC session language depends on whether you are on a MARC/1 dialog or another MARC dialog, and whether the language you designate is recognized by MARC. The following table shows the impact of setting a language based on these factors.

Language Designation	MARC/1 Dialog	Other MARC Dialog
Language Bound to MARC	Both the MARC session language and the Transaction Server application session language are set to the designated language.	Only the MARC session language for the current dialog is set. The Transaction Server application session language is unchanged.
Language Not Bound to MARC	The Transaction Server application session language is set to the designated language. The MARC session language is unchanged.	Neither the MARC session language nor the Transaction Server application session language is changed.

Establishing an MLS Environment

When you set the session language for a CANDE session, the language is accepted whether or not it is bound to CANDE or the MCP. The language is valid only for the current CANDE session.

The selected language overrides the default language for the system and the language selected at the usercode level. However, once you end the session, the session language is discarded. You must reset the language, if desired, for any subsequent sessions.

If you initiate an application product created through SDF Plus—such as SDF Plus itself—from MARC, the current Transaction Server application session language becomes the default language for the application product. For example, assume your Transaction Server application session language is Espanol and your MARC/2 session language is French. If you initiate SDF Plus from the MARC/2 session, SDF Plus searches for the Espanol, not French, version of the SDF Plus form, text, and help libraries. If you initiate an application product from CANDE, the current CANDE session language becomes the default for the product.

If no localized libraries in the session language exist, SDF Plus searches for localized libraries in the language for your usercode. If no localized libraries in the usercode language exist, SDF Plus searches for localized libraries in the current system language. And, if no localized libraries in the current system language exist, SDF Plus uses the unlocalized version of the libraries.

When you set the convention for your CANDE session, it does not set the convention for your MARC session. When you set the convention on your MARC/1 dialog, it does not affect your MARC/2 session.

To change the session language once you are running an SDF Plus application product, you must exit the SDF Plus application product and use the standard MARC or CANDE methods for changing the session-level language as explained in the following text.

Applications based on Transaction Server can call a service function to retrieve the language and convention associated with a session. Refer to the *Transaction Server Operations Guide* for details.

Viewing and Setting a Natural Language

You can set a language for your MARC session by using either MARC menus and screens or the LANGUAGE command. You can also use the LANGUAGE command to set a language for your CANDE session.

Using MARC to Determine the Current Session Language

You can find out the name of the current MARC session language by using MARC menus and screens as follows:

1. From any MARC menu, type *GO SC* in the Action field and transmit. Or, from the MARC Home menu, type *SC* in the Choice field and transmit.

MARC displays the SC—SESSION CONTROL menu.

2. Type *SSLANG* in the Choice field of the SC—SESSION CONTROL menu and transmit.

MARC displays the SSLANG—Show/Set Language menu.

3. To determine the current session language, type *LANG* in the Choice field of the SSLANG menu and transmit.

MARC displays the current session language.

Using MARC to Set the MARC Session Language

You can set the MARC session language by using MARC menus and screens as follows:

1. From any MARC menu, type *GO SC* in the Action field and transmit. Or, from the MARC Home menu, type *SC* in the Choice field and transmit.

MARC displays the SC—SESSION CONTROL menu.

2. Type *SSLANG* in the Choice field of the SC—SESSION CONTROL menu and transmit.

MARC displays the SSLANG—Show/Set Language menu.

3. To change the current session language, type *SLANG* in the Choice field of the SSLANG menu and transmit.

MARC displays the LANGUAGE—SET SESSION LANGUAGE screen.

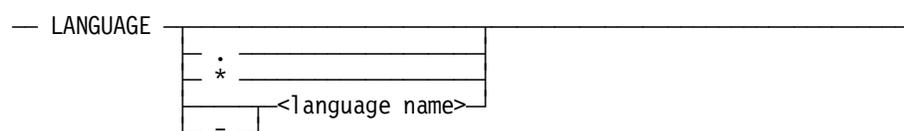
4. Type the desired language in the Language To Be Used field and transmit.

The new session language becomes effective immediately.

Using Commands to View and Set the Session Language

You can view and set the session language by using the LANGUAGE command either in CANDE or in the Action field on any MARC screen. The following diagram shows the syntax for this command:

LANGUAGE Command



Establishing an MLS Environment

The following table explains the elements of the LANGUAGE command:

Element	Explanation
LANGUAGE	Causes the system to display the current session language when used by itself, without any other parameters.
. (period)	Causes the session to use the system default language as the current language.
* (asterisk)	Causes the session to use the default language set for the usercode as the current language.
<language name>	Identifies the name of the language to use as the session language. The name must be composed of from 1 to 17 alphanumeric characters and begin with a letter. It must identify a language that is bound to MARC. For example, if the language is bound to MARC as Espanol, do not type Spanish. In this case, MARC does not recognize Spanish as the same language as Espanol.

Note: The use of the equal sign (=) and the period (.) is not supported in CANDE.

If the language name is not valid, the current session language is not changed.

The following examples illustrate the use of the LANGUAGE command.

Example 1

```
LANGUAGE
```

The system displays the current language for the session.

Example 2

```
LANGUAGE .
```

The system changes the session language to the system default language when the period is used with the LANGUAGE command.

Example 3

```
LANGUAGE *
```

The system changes the session language to the language set at the usercode level.

Example 4

```
LANGUAGE ESPANOL  
LANGUAGE = ESPANOL
```

Both these commands set the current session language to Espanol. Note that the second command, which uses the equal sign, can be entered only in MARC. The equal sign is not valid in CANDE.

Viewing Convention Definitions at the MARC Session Level

To view convention definitions at the session level, follow these steps:

1. Display the LOCAL menu.
2. Type *SSCONV* in the Choice field of the LOCAL menu and transmit. MARC displays the SSCONV—Show/Set Convention screen.
3. Type *SCONV* in the Choice field of the SSCONV menu and transmit.
MARC displays the CONVENTION—SESSION CONVENTION MANAGEMENT screen.
4. From this screen, you can view the elements of a convention definition by making the following two entries:
 - Type the name of the convention in the New Session Convention field.
 - Type an *X* in the applicable Convention Definition field to display the current convention definitions stored in memory, or to display copies of convention definitions stored on disk. The copy on disk contains any changes made since the last halt/load. After typing your choice in the field, transmit the screen.

The system displays the elements in the specified convention.

Setting Conventions at the Session Level

You can set a convention for your MARC session by using either MARC menus and screens or the CONVENTION command. You can set a convention for your CANDE session by using the CONVENTION command.

Using MARC to Set Conventions at the MARC Session Level

You can set a convention at the session level by using MARC menus and screens as follows:

1. From any MARC menu, type *GO SC* in the Action field and transmit. Or, from the MARC Home menu, type *SYS* in the Choice field and transmit.

MARC displays the SC—SESSION CONTROL menu.

2. Type *SSCONV* in the Choice field of the SC—SESSION CONTROL menu and transmit.

MARC displays the SSCONV—Show/Set Convention screen, from which you can do the following tasks:

- Display the current convention for the session.
 - Access the screen to change the convention for the session.
 - Show the current convention setting for the system.
 - List the available conventions.
3. To change the current convention, type *SCONV* in the Choice field of the SSCONV screen and transmit.

MARC displays the CONVENTION—SESSION CONVENTION MANAGEMENT screen, from which you can do the following tasks:

- List the available conventions.
- View the elements of a convention definition.
- Change the convention for the session.

It is a good practice to display the elements of a convention before selecting that convention.

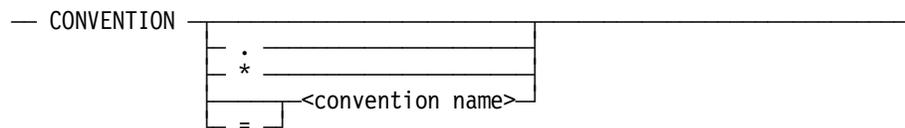
To change the convention for the session, type the name of the desired convention in the New Session Convention field and transmit. Leave all other fields blank.

If the convention name is valid, the new convention becomes effective immediately. If the specified convention name is not valid, the current convention stays in effect and MARC displays a message that the name has not been defined in the SYSTEM/CONVENTIONS file.

Using Commands to View and Set the Session-Level Convention

You can view and set the session-level convention by using the CONVENTION command either in CANDE or in the Action field on any MARC screen. The following diagram shows the syntax for this command:

CONVENTION Command



The following table explains the elements of the CONVENTION command:

Element	Explanation
CONVENTION	Used by itself, without any parameters, the command causes the system to display the name of the current session convention.
. (period)	Causes the session to use the system default convention as the current convention.
* (asterisk)	Causes the session to use the default convention designated for the usercode as the current convention.
<convention name>	Identifies the name of the convention to use as the session convention. The name must be composed of from 1 to 17 alphanumeric characters. It must identify a convention definition found in the SYSTEM/CONVENTIONS file. For information about obtaining the names of conventions available on your system, refer to Table 6-9, "Identifying the Available Convention Definitions."

Note: The use of the equal sign (=) and the period (.) is not supported in CANDE.

If the specified convention name is not valid, MARC displays an error message and the session convention is not changed.

The following examples illustrate the use of the CONVENTION command.

Example 1

CONVENTION

This command displays the name of the current convention for the session.

Example 2

CONVENTION.

The session convention is changed to the system default convention for the system.

Example 3

```
CONVENTION *
```

This command changes the session convention to the convention designated at the usercode level.

Example 4

```
CONVENTION BELGIUM  
CONVENTION = BELGIUM
```

Both these commands set the current session convention to Belgium. Note that the second command, which uses the equal sign, can be entered only in MARC. The equal sign is not valid in CANDE.

Setting the Language and Convention at the Task Level

At the task level, you can set the language and the convention by using the LANGUAGE and CONVENTION task attributes. This method overrides the language or convention that was set at the system, session, or usercode level.

You can specify a language and convention in a program through a task attribute. For example, a language can be specified in a Work Flow Language (WFL) job with the LANGUAGE task attribute set to a particular language.

For more information on the LANGUAGE and CONVENTION task attributes, refer to the *Task Attributes Programming Reference Manual* and the *Task Management Programming Guide*.

Using Ccsversions, Languages, or Conventions at the Program Level

At the program level, an application program can access the procedures in the CENTRALSUPPORT library to perform actions based on any ccsversion, language, or convention. Settings at the system, session, usercode, or task levels are overridden.

Syntax in some programming languages enables you to specify a ccsversion or convention programmatically. For example, if you establish a specific ccsversion in COBOL74, the compiler uses the collating sequence associated with that ccsversion to perform alphanumeric comparisons.

The following programming language reference manuals contain information on the syntax required for using ccsversions, languages, and conventions at the program level:

- *ALGOL Programming Reference Manual, Volume 1: Basic Implementation*
- *C Programming Reference Manual, Volume 2: Headers and Functions*
- *COBOL ANSI-74 Programming Reference Manual, Volume 1: Basic Implementation*

- *COBOL ANSI-85 Programming Reference Manual, Volume 1: Basic Implementation*
- *Pascal Programming Reference Manual, Volume 1: Basic Implementation*
- *Report Program Generator (RPG) Programming Reference Manual, Volume 1: Basic Implementation*

Section 5

Localizing Products in an MLS Environment

This section explains the way to localize the products that you have created by using products in the MLS environment. This section also explains how to localize system software products. For information about creating products that can be localized, refer to Section 6, "Creating Application Systems in an MLS Environment."

In order to view your products and the system software in your native language, you must first ensure that the translated user interfaces for the products are available on your system. You must then make the translated user interfaces accessible.

This section briefly summarizes the following information:

- Localizing your products
- Localizing screens
- Localizing help text
- Localizing system software products
- Making translated files available

Localizing Your Products

Some of the products that you have created by using the system software can be localized. The screens, messages, and help text for your products must have been created so that they can be localized.

Table 5–1 lists the translation tools that you can use to localize the screens, messages, and help text for your products. The table also describes the way the screens, messages, and help text must be defined for you to use the translation tools listed.

Once you have translated the screens, messages, and help text for your product, you must select the appropriate system options to make the localized product available. Refer to Section 4, “Establishing an MLS Environment,” for more information.

Table 5–1. Translation Tools for Your Products

Element	How the Element Must Be Defined	Translation Tool
Screens	Menugraph	IMG
	Form library	SDF Plus
Messages	Output message array	MSGTRANS
	Text library	SDF Plus
Help text	Help text symbol file	Help Utility
	Help library	SDF Plus Help

Translating Screens for Your Products

To translate the screens in your products, the screens must be created either in an SDF Plus form library or in an IMG menugraph. You can translate these screens as follows:

- Translate the screens in an SDF Plus form library by using the procedures described under “Localizing SDF Plus Screens” later in this section.

These procedures provide a general description of the process. For complete details, refer to the *SDF Plus Installation and Operations Guide*.

- Translate the screens in IMG menugraphs by using the procedures described under “Localizing Screens with IMG” later in this section.

These procedures provide a general description of the process. For complete details, refer to the *IMG Operations Guide*.

Before you can translate any screens, you must know the name of the form library or menugraph that contains the screens for your product.

Translating Messages for Your Products

To translate the messages in your products, the messages must be created either in an SDF Plus text library or in output message arrays defined in an application program. You can translate these messages as follows:

- Translate the messages in an SDF Plus text library by using the procedures described under “Localizing SDF Plus Screens” later in this section.

These procedures provide a general description of the process. For complete details, refer to the *SDF Plus Installation and Operations Guide*.

- Translate the messages in output message arrays by using the procedures described in the *MSGTRANS Operations Guide*.

Before you can translate any messages, you must know the name of the text library or object file that contains the messages for your product.

Translating Help Text for Your Products

To translate the help text in your products, the help text must be defined as one of the following:

- An SDF Plus help library created with the SDF Plus Forms Editor
- A help library created with the SDF Plus Help utility
- A help book created with the Help Utility

You can translate these types of help text as follows:

- Translate the help text in an SDF Plus help library by using the procedures described under “Localizing SDF Plus Screens” later in this section.

These procedures provide a general description of the process. For complete details, refer to the *SDF Plus Installation and Operations Guide*.

- Translate the help text created with the SDF Plus Help utility or the Help Utility by using the procedures described under “Localizing Help Text” later in this section.

Before you can translate any help text, you must know the name of the help library or help text symbol file that contains the help text for your product.

Localizing Screens

Screens are displays that require a user to enter information or that present information. Entered information is then processed by the underlying program. Menus are displays that require a user to select from a list of choices to move to the next display. For the purposes of this guide, the term *screen* is used to mean both screens and menus.

Two programming tools are available to help you develop mainframe internationalized screens: IMG and SDF Plus. MARC, IMG, BNA Version 2, Transmission Control Protocol/Internet Protocol (TCP/IP) Application Services for ClearPath MCP, and Interactive Datacomm Configurator (IDC) screens are developed with IMG. Other internationalized mainframe product screens are developed with SDF Plus.

Localizing Screens with IMG

Any screens based on IMG that you have customized or added to the MARC product can be localized. However, do not localize the screens of the MARC product itself. To obtain a localized version of MARC screens in the language you want, contact your Unisys representative or subsidiary.

All screens that are based on IMG are stored in a file referred to as a *menugraph*. The set of menugraphs in use on a system is known as the *active menugraph list*. You can display the menugraph file titles in the active menugraph list by entering the MENU command in the Action field of any MARC screen that accepts the MENU command. The MENU command is useful for obtaining the names of the menugraphs that you want to localize.

The screens based on IMG that you can localize are either MARC system menugraphs that you have customized or new menugraphs that you have created. The localization of screens based on IMG depends on the type of menugraph you are localizing, as follows:

- To localize MARC system menugraphs that you have customized, you need to use IMG to apply the changes that you made in your custom MARC system menugraphs to the MARC system menugraphs in the new software release.

If your changes cannot be applied, IMG prints a report that identifies the changes that could not be applied. For each of the items reported, you must translate the text in the menugraph.

- To localize menugraphs that you have created, you need to translate the text for each menugraph you have created.

Regardless of the type of screens based on IMG you are localizing, you must establish the languages of these screens in MARC after you have localized the screens. To connect a localized version of your screens based on IMG with the MARC/COMMANDER software, you must add the name of the localized language to message 0 (zero) of the MLSMSG array in the file *SYSTEM/MARC/COMMANDER.

Every language in which the screens based on IMG are localized, except English, must be identified in message 0 of the MLSMSG array. For example, in order for MARC to recognize that menugraphs have been localized in French and Dutch, you must add both French and Dutch to message 0 of the MLSMSG array. The name of the language must

match the name of the language as it is bound to the MCP. When the customer selects a language, the *SYSTEM/MARC/COMMANDER file searches for the appropriate menugraph—for example, the localized Dutch menugraph.

You can use the MSGTRANS utility to add a new language to message 0 of the MLSMSG array. The steps to do this procedure are listed in the *MARC Operations Guide*. For information on using MSGTRANS, refer to the *MSGTRANS Operations Guide*.

For complete instructions on localizing custom IMG screens, refer to the *MARC Operations Guide* and the *IMG Operations Guide*.

Localizing SDF Plus Screens

The following text describes the way to translate and process screens for the product SDF Plus as well as all products that use SDF Plus to develop their user interfaces on the host system.

Each product has its own form, text, and help libraries. A form library contains the forms displayed to the user and the instructions for processing the user interface. A text library contains messages and other text displayed to the user by the user interface. A help library contains information that is displayed online when a user requests help on a prompt, field, menu, or form.

The form libraries and text libraries are localized by using the procedures and guidelines detailed in the *SDF Plus Installation and Operations Guide*.

Briefly, to create a localized version of a library, you must

1. Process the initial release of the library by using the SDF Plus Store feature, and process subsequent releases by using the SDF Plus Compare feature. The Store and Compare features create a user interface definition language (UIDL) file for the library.
2. Translate the desired elements in the UIDL file, such as text strings, and localize other elements, such as the language and ccsversion entries, and the date and time formats.
3. Process the translated and localized UIDL file by using the SDF Plus Load feature. The Load feature creates a localized library.
4. Use the SDF Plus Forms Editor to adjust the position of components on the form image if translated text is longer than the original text string.
5. If you are localizing a product created with SDF Plus, use the MSGTRANS utility to translate the following OUTPUTMESSAGE arrays in the *SYSTEM/SDFPLUS/ARCHIVEMANAGER file:

AM_ERRORS	RESERVED_WORDS
FATAL_ERRORS	SCRIPT_ERRORS
FS_CONTROL_CHARS	SCRIPT_MESSAGES
FS_ERRORS	SDF_PROMPTS
NONFATAL_ERRORS	SYNTAX_ERRORS

If you are localizing the SDF Plus product itself, translate all OUTPUTMESSAGE arrays in the *SYSTEM/SDFPLUS/ARCHIVEMANAGER file.

For information on using MSGTRANS, refer to the *MSGTRANS Operations Guide*.

6. Translate all associated help text.

While you can use these steps to localize help libraries, you should use the SDF Plus Help utility to localize all help libraries for system software products that are based on SDF Plus.

Refer to “Identifying Unisys Products Localized Using SDF Plus” later in this section and Appendix D, “Help Text Available for Localization,” for more information about localizing help libraries.

Establishing Languages for SDF Plus

Two distinct SDF Plus environments exist: run time and edit. The run-time environment determines the language that is used when the system displays the screens, messages, and help text for products created with SDF Plus and for the SDF Plus product. The edit environment determines the forms that can be edited by the SDF Plus Forms Editor.

Setting a Language for the SDF Plus Run-Time Environment

When you initiate an application product created through SDF Plus from MARC or from CANDE, the system uses the current session language for MARC or CANDE as the default language for displaying the application product. For example, if you initiate SDF Plus from a CANDE session and the language for the CANDE session is Francais, SDF Plus searches for the Francais version of the SDF Plus form, text, and help libraries.

If no libraries in the session language exist, SDF Plus performs a hierarchical search of the following libraries, stopping when a library is found:

- Localized libraries for the system language
- Unlanguageed libraries

The term *unlanguageed* means that no language is associated with the form, help, or text library. If a library is unlanguageed, it does not mean that it is not translated or that it is the language in which the library was created originally. The unlanguageed library is the one SDF Plus attempts to find if it cannot find a library in the session language of the user or in the system language. Therefore, if you localize the form and text libraries in the session language but do not localize the help library for the language, the localized forms and text are used along with the unlanguageed help library.

Once you have initiated the SDF Plus application product, you must exit the product to change languages. Use the standard MARC or CANDE methods for changing the session-level language as explained under “Viewing and Setting a Natural Language” in Section 4 before restarting the SDF Plus application product.

Setting a Language for an SDF Plus Edit Session

The SDF Plus Forms Editor is used to create and modify forms. The same form can exist in several languages. For example, a form can have English, French, German, and Japanese versions.

By setting a language on the SDF Plus Session Options form, you determine the version of the form that is shown for editing. If you do not set a language, the unlanguageed version of the form is used.

To set a language for an edit session, use the following procedure:

1. Initiate SDF Plus either from the MARC Tools screen or from a CANDE session. If you use a CANDE session, enter and transmit the following statement:

```
RUN $SYSTEM/SDFPLUS ON <pack>
```

Use the syntax *ON <pack>* to identify the pack on which the program resides. If the program is on your default pack, you can omit this syntax from the statement.

SDF Plus displays the SDF Plus Main Menu.

2. Enter *SETOPTIONS* in the Choice field and transmit. The SDF Plus Session Options form is displayed.
3. On the Session Options form, enter information for the data dictionary, directory, language, and ccsversion. Remember the following guidelines:
 - If you leave the Language field blank or if you enter an invalid language, SDF Plus accesses the unlanguageed libraries for the edit session.
 - The ccsversion must be valid for the designated language.
 - If you want to use the designated language for subsequent SDF Plus edit sessions, enter a *Y* in the Retain Options for Future Sessions field.
4. Transmit the screen and return to the SDF Plus Main Menu to begin using the Forms Editor.

Localizing Help Text

Help text files contain information about the elements of a user interface. The text is displayed online at the user's request. For example, a user might display information by placing the cursor on a field or a prompt and pressing the SPCFY key.

Help text exists for both mainframe products and workstation products. The help text that is created for mainframe products is created very differently than the help text created for workstation products. The following paragraphs explain the way to localize help text created on the mainframe. For information on localizing help text created for workstation products, refer to "Localizing Workstation Products" later in this section.

Translating a Help Text Symbol File

To localize mainframe help text, you must have either the help text symbol file or the SDF Plus help library. A symbol file contains text and embedded help utility commands. You translate the text and then process the file through a help utility to create a help file that your users can access. For each natural language used on your system, you must produce a separate help file. For example, if you are localizing help text for both Spanish-speaking and Italian-speaking audiences, then you must produce one file in Spanish and one file in Italian.

A help utility is a program that processes the help text symbol file to produce the local help text file. Currently, you must use one of two help utilities to process your files:

- Help Utility

To translate help text created with the Help Utility, you need the help text symbol file for the particular product that you want to translate.

Once you have the help text symbol file, follow the guidelines provided in the remainder of this discussion.

- SDF Plus Help utility

To translate an SDF Plus help library, you need either the help text symbol file or the help library for the particular product that you want to translate.

If you are using the help text symbol file, follow the guidelines provided in the remainder of this discussion. If you are using the help library, refer to "Localizing SDF Plus Screens" earlier in this section for an overview of the process. For complete details on translating help libraries, refer to the *SDF Plus Installation and Operations Guide*.

Note: *If you are localizing help text for a Unisys product, the help text symbol files are not provided with the purchase of the product—they must be purchased separately. However, help libraries are provided with those products that are SDF Plus applications.*

When you translate a help text symbol file, you need to translate only part of the information contained in the file. The information you must translate includes the text the user sees and the commands that affect the access to the text.

The help utility commands are prefixed with a period (.). The period must always be in column 1 (the leftmost column) of the file. You do not translate the command, but might translate text associated with the command.

For more detailed information on the Help Utility, refer to the *Help Utility Operations Guide*. For more detailed information on the SDF Plus Help utility, refer to the *SDF Plus Installation and Operations Guide*.

The remainder of this discussion describes the text and commands in a help text symbol file that should and should not be translated and the special requirements for each help utility.

Translating Commands That Identify the Help Book

For a help text symbol file to be processed correctly by a help utility, certain help utility commands must be present at the beginning of the file.

The help utility commands are used to identify the help text file and the product. The information provided in each command can vary according to the help utility used to generate the help book, as explained in Table 5-2.

Table 5-2. Required Help Book Commands

Help Utility Command	For Products Using the Help Utility	For Products Using the SDF Plus Help Utility
.BOOK FILETITLE	Identifies the name of the product.	Identifies the name of the help library.
.LANGUAGE	Identifies the language of the help book.	Identifies the language of the help library.
DIRECTORY	Not present.	Identifies the name of the directory for the help library.

When the help file is processed by the help utility, the information in these lines is used to create the name of the localized help file. The names have the following formats:

- For products using the Help Utility, the default format is
`BOOK/<product name>/<language>`
- For products using the SDF Plus Help utility, the required format is
`HELPLIB/<directory name>/1<help library name>/<language>`

Translating the **.BOOK FILETITLE** Command

For help books created with the Help Utility, the `.BOOK FILETITLE` command specifies the product associated with the file. For help books created with the SDF Plus Help utility, this command specifies the name of the help library. When you translate the help text symbol file, you must not change this command. Examples of `.BOOK FILETITLE` commands are

```
.BOOK FILETITLE OCM
```

```
.BOOK FILETITLE IDC
```

```
.BOOK FILETITLE FEHELP
```

Translating the **.LANGUAGE** Command

The `.LANGUAGE` command specifies the language of the text. When you localize the file, you must change this command to reflect the language you are translating the file to— for example, French or Spanish. Examples of `.LANGUAGE` commands are

```
.LANGUAGE ENGLISH
```

```
.LANGUAGE ESPANOL
```

```
.LANGUAGE FRANCAIS
```

The language you assign must be recognized by the product software; for example, if the product recognizes the language `GERMAN`, you must use `GERMAN` and not `DEUTSCH`.

Translating the **.DIRECTORY** Command

The `.DIRECTORY` command in an SDF Plus help text symbol file identifies the directory for the help library. When you translate the file, you must not change this line. An example of a `.DIRECTORY` command is

```
.DIRECTORY SDFPLUS
```

Translating Commands That Identify the Help Text

Besides the help utility commands that identify the help book or library, help text symbol files also contain .SECTION commands and .KEYWORD commands. These two commands define the amount of help text that can be accessed and the help text to access. For example, the .SECTION command could define the portion of help text displayed for a particular screen.

Table 5–3 describes each of these commands.

Table 5–3. Commands That Identify Help Text

Help Utility Command	Explanation
.SECTION "<text>"	<p>A help utility command designating a section of help text. For example,</p> <p style="text-align: center;">.SECTION "Using the PAINT Screen"</p> <p>A section is a block of help text that can be scrolled through. The text between the quotation marks (") is displayed as the title of the help screen.</p> <p>You should translate the text between the quotation marks.</p> <p>Note that sometimes no text appears between the quotation marks.</p>
.KEYWORD <keyword>	<p>A help utility command that creates a link between the help text and a piece of the product or another piece of help text.</p> <p>The keyword can be designated by the product or by the writer of the help text.</p> <p>Translate the keywords only if the keywords in the product are being translated or if the keywords in the product are embedded in another piece of help text. Work with the person who is localizing the product to ensure that the help text keywords match the product keywords.</p>

Translating .SECTION Commands

When you translate the text of a section command, you must be aware of the limitations put on the length of the text by the product you are documenting.

- For help text symbol files processed using the Help Utility, the text following a .SECTION command must not exceed 48 characters. If you exceed 48 characters, the text is truncated when it is displayed.
- For help text symbol files processed using the SDF Plus Help utility, the text following a .SECTION command must not exceed 59 characters. If you exceed 59 characters, the text is truncated when it is displayed.

Translating .KEYWORD Commands

You should add localized keywords instead of replacing the keywords that you want to translate. Localize keywords only if the keywords are embedded in help text or if the keywords in the product are translated. If you localize keywords, observe the following restrictions:

- Keywords are restricted in length to
 - characters for files processed with the Help Utility
 - characters for files processed with the SDF Plus Help utility
- The first 30 characters of the keyword must be unique.

For example, the following two keywords are considered identical by both help utilities because the first 30 characters are the same:

```
.KEYWORD 12345678901234567890123456789011111111
```

```
.KEYWORD 12345678901234567890123456789044444444
```

If identical keywords occur in a help text symbol file, an error is returned when the file is processed through the help utility.

- No convention is imposed for keyword naming. However, some products using IMG use the following convention:

```
<identifier>H
```

```
<identifier>T
```

The following table explains the elements of the naming convention:

Element	Explanation
<identifier>	Identifies an element on a screen.
H	Denotes the short help keyword.
T	Denotes the long help keyword.

Note: *The short and long help for a product can be included in the same help book or can be separated into two help books.*

- Keywords that contain any of the special characters listed in Table 5-4 must be delimited as follows:
 - By asterisks (*) in files processed using the Help Utility
 - By asterisks (*) in files processed using the SDF Plus Help utility
- Phrases embedded in the text that have associated keywords must appear on one line of the text and must be delimited by asterisks (*).
- For files processed using the SDF Plus Help utility, you can use either the .NEXTKEYWORD command or the .AUTONEXTKEYWORD command to identify the long help associated with a section of short help.

Table 5-4. Keyword Special Characters

Character	Description	Character	Description
!	Exclamation mark	#	Number sign
%	Percent sign	&	Ampersand
(Left parenthesis)	Right parenthesis
_	Underscore	`	Single quotation mark
{	Left brace	}	Right brace
]	Right square bracket	:	Colon
<	Left angle bracket	>	Right angle bracket
?	Question mark	/	Slash
;	Semicolon	,	Comma
~	Tilde	"	Double quotation mark
\$	Dollar sign	=	Equal sign
'	Apostrophe	@	At sign
[Left square bracket	+	Plus sign
	Vertical bar	.	Period
\	Back slash		

- If the .AUTONEXTKEYWORD command is used, the keywords in the help text symbol file must use the following convention for identifying the short and long help text associated with a screen element:

`<identifier>S`

`<identifier>L`

The `<identifier>` variable indicates a screen element, the `S` denotes the short help keyword, and the `L` denotes the long help keyword.

If the help text symbol file you are localizing uses this convention, the line .AUTONEXTKEYWORD appears near the beginning of the file.

Translating Text and Embedded Keywords

Following the help utility commands that identify each section of the help text is the text that describes the product. This area of the help text symbol file contains most of the text you localize. You can translate the help text associated with each .KEYWORD command.

No set format for this part of the file exists, except for short help. Short help text must be no more than 2 lines of 72 characters each. Additional text is truncated and is not displayed at run time.

When you translate the help text, you might find embedded keywords defined within the help text rather than with the .KEYWORD command. Embedded keywords enable a user to access more information about a topic mentioned in the help text. You must translate embedded keywords.

You can identify the embedded keywords in several ways:

- They are delimited by asterisks (*) or angle brackets (<>). The following example illustrates this type of keyword:

How do I get off this screen?

- They are usually terms or phrases used in the product.

When you translate an embedded keyword, you must ensure that any similar text in the rest of the file is translated in the same way. For example, assume the embedded keyword *house* is used in the file. If you are localizing the file for French users and decide to translate *house* as *maison* in one place, you must use *maison* throughout the file rather than using *maison* interchangeably with *villa*.

If you are not consistent in your translation of embedded keywords, the user cannot access the information. Using the example in the preceding text, you should add *maison* to the .KEYWORD command instead of replacing *house* in case references to *house* are in the product.

Many embedded keywords are also terms or phrases used on product screens. When you translate these types of embedded keywords, your changes should match those changes made to the product screens.

Localizing Unisys Products

Many Unisys products are available in a variety of natural languages. Check with your Unisys representative to determine if the products you use are available in the language you want. If you are a customer outside of the United States, you should contact your Unisys subsidiary or representative to obtain localized versions of Unisys products.

If your Unisys representative or subsidiary does not have a localized version of a product in your language and the product you want can be localized, you can localize the product yourself.

Table 5-5 lists the Unisys products that are available for localization. The table also describes the portions of each product that can be localized (screens, messages, or help text), and provides a reference to further information about localization of each product. References to Appendix D and Appendix E in the table refer to Appendix D, "Help Text Available for Localization," and Appendix E, "Messages Available for Localization," in this guide. For example, ALGOL compiler can have localized messages, and the messages available for localization can be found in Appendix E.

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
Administration Center (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
ALGOL compiler	Messages	Appendix E
ALGOL Test and Debug System (TADS)	Messages Help text	Appendix E Appendix D
Binder	Messages	Appendix E
BNA installation	Help text	Appendix D
BNA Version 2	Screens Messages Help text	Overview: "Localizing Screens with IMG" earlier in this section Details: <i>MARC Operations Guide</i> <i>and the IMG Operations Guide</i> Appendix E Appendix D
C compiler	Messages	Appendix E
C TADS	Messages	Appendix E
CD-ROM Formatter (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
CENTRALSUPPORT library	Messages	Appendix E

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
Client Access Services (Host)	Messages	Appendix E
COBOL Migration Tool (CMT)	Screens Help Text	Appendix D
COBOL74 compiler	Messages	Appendix E
COBOL74 TADS	Messages Help text	Appendix E Appendix D
COBOL74LOADER utility	Screens Messages Help text	Table 5-7, "Unisys Products Localized with SDF Plus" Appendix E Appendix D
COBOL85 compiler	Messages	Appendix E
COBOL85 TADS	Messages	Appendix E
Command and Edit (CANDE)	Messages Help text	Appendix E Appendix D
COMTI for ClearPath MCP (PC)	Dialog boxes, String tables, Help text, event log, error messages, HTML files	Table 5-8, "Unisys Workstation Products That Can Be Localized"
Data Comm ALGOL Compiler	Messages	Appendix E
DATAAID	Help text	Appendix D
Database Certification	Messages	Appendix E
Database Monitor	Messages	Appendix E
Database Operations Center (Host)	Messages	Table E-1, "Messages Available for Localization with MSGTRANS"
Database Operations Center (PC)	Dialog boxes, menu bars, string tables, Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
DCDLPDUMPANALYZER utility	Messages	Appendix E
DCSTATUS	Messages	Appendix E
Disk Cache	Messages	Appendix E
DMALGOL compiler	Messages	Appendix E
DMINTERPRETER library	Messages	Appendix E

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
Documentation Library Overview	Help text	Appendix D
DUMPALL utility	Messages	Appendix E
End User Center (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
Enterprise Database Server for ClearPath MCP	Messages Help text	Appendix E
Enterprise Database Server Inquiry	Messages	Appendix E
Enterprise Database Server Transaction Processing System (TPS)	Messages	Appendix E
FAULTLOGGER utility	Messages	Appendix E
FILEDATA utility	Messages	Appendix E
GENERALSUPPORT library	Messages	Appendix E
Generic Security Services—API	Messages	Appendix E
Glossary	Help text	Appendix D
HLCNTS	Messages	Appendix E
Host LAN Connection (HLCN)	Messages	Appendix E
Host Services: Distributed System Services (DSS) Native File Transfer	Screens Messages Help text	Overview: "Localizing Screens with IMG" earlier in this section Details: <i>MARC Operations Guide and the IMG Operations Guide</i> Appendix E Appendix D
HWERRORSUPPORT library	Messages	Appendix E
InfoExec Advanced Data Dictionary System (ADDS)	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus," and Appendix D Appendix E
InfoExec Main Menus	Help text	Appendix D

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
InfoExec Operations Control Manager (OCM)	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus" and Appendix D Appendix E
InfoExec Workstation Query Facility (WQF)	Help text	Appendix D
Installation Center (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
Interactive Configurator (IC)	Screens Help text	Table 5-7, "Unisys Products Localized with SDF Plus," and Appendix D
Interactive Datacomm Configurator (IDC)	Screens Messages Help text	Overview: "Localizing Screens with IMG" earlier in this section Details: <i>MARC Operations Guide and the IMG Operations Guide</i> Appendix E Appendix D
Interactive Menugraph Generator (IMG)	Help text	Appendix D
Interactive Sort (ISORT)	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus" Appendix D Appendix E
Java Servlet Manager	Dialog boxes, Screens, String Tables	Table 5-8, "Unisys Workstation Products That Can Be Localized"
JOBFORMATTER	Messages	Appendix E
KEYEDIOII utility	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus" Appendix D Appendix E
LANPRT (LAN Printing Enabler)	Messages Help Text	Appendix E Appendix D
Local port provider (LPP)	Messages	Appendix E
LOGANALYZER utility	Messages	Appendix E
Logon Agent (PC)	Dialog boxes and String tables	Table 5-8, "Unisys Workstation Products That Can Be Localized"
Mail System	Help text	Appendix D

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
MAPPER Relational Interface for ClearPath NX and A Series (Business Information Server Relational Interface)	Messages	Appendix E
MAPPER System for ClearPath NX and A Series (Business Information Server)	Messages	Appendix E
MARCUS	Messages Help Text	Appendix E Appendix D Note: <i>The MARC software provides screens for DSS, Native File Transfer, OSI FTAM, and TCP/IP Application Services for ClearPath MCP. The MARC help book includes the help text for DSS, Native File Transfer, OSI FTAM, and TCP/IP Application Services for ClearPath MCP.</i>
Master control program (MCP)	Messages	Appendix E
Menu-Assisted Resource Control (MARC)	Screens Messages Help text	Contact your Unisys representative. For information on using a localized version of MARC, refer to the <i>MARC Operations Guide</i> . Appendix E Appendix D
Message Translation Utility (MSGTRANS)	Help text	Appendix D
MESSAGESUPPORT library	Messages	Appendix E
Network Database OLE DB Data Provider for ClearPath OS 2200 - Host	Messages	Table E-1, "Messages Available for Localization with MSGTRANS"
Network Database OLE DB Data Provider for ClearPath OS 2200 - PC	String Tables and Help Text	Table 5-8, "Unisys Workstation Products That Can be Localized"
Network Definition Language II (NDLII) compiler	Messages	Appendix E

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
Network Definition Language II (NDLII) post compiler	Messages	Appendix E
NETWORKSUPPORT library	Messages	Appendix E
NX/View (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
ODBC Access Drivers [Host Component]	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus," Table D-1, "Help Text Available for Translation," and Table E-1, "Messages Available for Localization with MSFTRANS"
ODBC Access Drivers [PC Product]	Dialog boxes, String tables, Help text	Table 5-8, "Unisys Workstation Products That Can be Localized"
ONC+ Remote Procedure Call	Messages	Appendix E
Open Systems Interconnection (OSI) File Transfer, Access, and Management (FTAM)	Screens Messages Help text	Overview: "Localizing Screens with IMG" earlier in this section Details: MARC Operations Guide and the IMG Operations Guide Appendix E Appendix D Note: The MARC software includes the screens for DSS, Native File Transfer, OSI FTAM, and TCP/IP Application Services for ClearPath MCP. The MARC help book provides the help text for DSS, Native File Transfer, OSI FTAM, and TCP/IP Application Services for ClearPath MCP.
Operations Center (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
Print Center (PC)	Dialog boxes Menu bars Messages Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
Programmer's Workbench for ClearPath MCP (Host)	Messages	Table E-1, "Messages Available for Localization with MSGTRANS"
Programmer's Workbench for ClearPath MCP (PC)	Messages	Table 5-8, "Unisys Workstation Products That Can be Localized"
Remote Database Backup	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus," and Appendix D Appendix E
Remote Display (RD) utility	Messages	Appendix E
Remote Job Entry/ Bisynchronous Communication (RJE/BSC)	Help text	Appendix D
Remote Print System	Messages	Appendix E
RPG compiler	Messages	Appendix E
RPRINTER	Messages	Appendix E
Screen Design Facility (SDF)	Messages Help text	Appendix E Appendix D
Screen Design Facility Plus (SDF Plus)	Screens Help text Messages	Table 5-7, "Unisys Products Localized with SDF Plus," and Appendix D Appendix E
Screen Design Facility Plus (SDF Plus) Migration Tool	Help text	Appendix D
Simple Installation (SI)	Screens Messages Help text	Appendix E Appendix D
SORT compiler	Messages	Appendix E
System Assistant	Messages	Appendix E
TCP/IP Remote Printing	Messages	Appendix E
Terminal Gateway	Messages	Refer to the <i>BNA Version 2 Network Software Implementation Guide, Volume 2: Configuration</i>
Transaction Center (PC)	String tables and Help text	Table 5-8, "Unisys Workstation Products That Can Be Localized"
Transaction Server	Messages Help text	Appendix E Appendix D

Table 5-5. Unisys Products That Can Be Localized

Product or Feature	What Can Be Localized	Source for Localization Information
Open/OLTP for ClearPath NX and A Series Distributed Transaction Processing (DTP)	Messages Help Text	Appendix E Appendix D
Transmission Control Protocol/Internet Protocol (TCP/IP) Application Services for ClearPath MCP	Screens Messages Help text	Overview: "Localizing Screens with IMG" earlier in this section Details: <i>MARC Operations Guide and the IMG Operations Guide</i> Appendix E Appendix D Note: <i>The MARC software includes the screens for DSS, Native File Transfer, OSI FTAM, and TCP/IP Application Services for ClearPath MCP. The MARC help book provides the help text for DSS, Native File Transfer, OSI FTAM, and TCP/IP Application Services for ClearPath MCP.</i>
WinRPC	Messages	Appendix E
Work Flow Language (WFL)	Messages	Appendix E
XlateEBCDIC	String tables	Table 5-8, "Unisys Workstation Products That Can Be Localized"

Once you have translated the screens, messages, or help text for a Unisys product, you must select the appropriate system options to make the localized product available. Refer to Section 4, "Establishing an MLS Environment," for more information.

Translating Messages for Unisys Products

Many of the messages for Unisys products have been put into output message arrays. These messages include the following:

- Operating system messages, which are messages that are produced by the master control program (MCP).
- System environment messages, which are messages produced by a message control system (MCS), for example, Command and Edit (CANDE) messages.

- Compiler messages, which are messages produced by a language compiler, for example, COBOL74.
- Program-specific messages, which are messages produced by an application system or program, for example, Screen Design Facility Plus (SDF Plus) or Advanced Data Dictionary System (ADDS) messages.

You can translate messages defined in output message arrays by using the MSGTRANS utility. Appendix E, "Messages Available for Localization," lists the Unisys products that contain messages you can translate with MSGTRANS. Appendix E also lists the names of the files that contain the messages for each product. Once you have identified the file or files that you need, refer to the *MSGTRANS Operations Guide* for complete instructions on translating messages.

Unisys products created with SDF Plus do not define messages in output message arrays. Therefore, you cannot use MSGTRANS to translate these messages. These messages are defined in a text library file and are translated by using SDF Plus. You can translate SDF Plus text libraries by using the procedures for translating SDF Plus screens described under "Localizing SDF Plus Screens" earlier in this section.

The one exception is that you can translate the SDF Plus messages contained in the SYSTEM/SDFPLUS/ARCHIVEMANAGER file by using MSGTRANS.

Translating Help Text for Unisys Products

Help text for Unisys products is created with either the Help Utility or the SDF Plus Help utility as follows:

- If the product you are localizing has a screen interface created with IMG or SDF Plus, then the help utility you need to use when translating help text depends on the tool that created the screen interface.

Table 5–6 matches the help utilities with their corresponding screen interfaces and gives examples of the Unisys products that have help text that was created with the help utility.

Table 5–6. Matching Help Utilities to Screen Interfaces

Help Utility	Screen Interface	Used For
Help Utility	IMG	MARC, DSS, Native File Transfer, IDC
SDF Plus Help utility	SDF Plus	ADDS, IC, ISORT, KEYEDIOII, OCM, Remote Database Backup, SDF Plus

- If the product you are localizing does not have a screen interface or was not created with IMG or SDF Plus, then you need to use the Help Utility when translating help text.

To translate help text for Unisys products, you must identify the name of the help text symbol file or help library that contains the help text. Appendix D, "Help Text Available for Localization," lists the products with help text that can be translated and lists the names of the help text symbol files or help libraries for each product.

Complete instructions for translating help text symbol files is provided earlier in this section under "Localizing Help Text." Instructions for translating help libraries is provided under "Localizing SDF Plus Screens" also earlier in this section.

In addition to the information described under "Localizing Help Text," you need to understand the following information that all Unisys help text symbol files contain:

- Copyright information
- Disclaimer information
- Trademark information

The remainder of this discussion describes each of these elements and the parts of these elements that should and should not be translated.

Copyright Information

All Unisys documentation is protected by copyright. The copyright is a legal statement limiting the use and reproduction of the information contained in the document. In the help text symbol files, the copyright information is prefixed by the following two lines:

```
.SECTION "FRONT MATTER"  
.KEYWORD COPYRIGHT
```

The text following these two lines is the Unisys standard copyright information. Do not localize this text.

Disclaimer Information

All Unisys documentation contains a disclaimer. The disclaimer text is prefixed by the line

```
.KEYWORD DISCLAIMER
```

The disclaimer is the Unisys standard disclaimer information. Do not localize the disclaimer.

Trademark Information

Following the disclaimer information is a list of the trademarked terms used in the file. A trademarked term is one that is used to identify a product or a service. The first time a trademarked term is used in a document, you must acknowledge the owner of the term. The trademark information is located after the following two lines:

```
.SECTION "TRADEMARK INFORMATION"  
.KEYWORD TRADEMARKS
```

Each trademarked term is followed by a statement that names the owner of the trademarked term. For example,

DIF (R) is a registered trademark of Lotus Development Corporation.

Do not localize any of the information in this section of the help text symbol file.

Identifying Unisys Products Localized with SDF Plus

Table 5-7 identifies the Unisys products and the corresponding form, text, and help library files for each product that you can localize with SDF Plus. The products are listed in alphabetical order. Files not listed in this table might be translatable with other tools. Check with your Unisys representative to determine the files that are translatable and the appropriate tool for translation.

Each file name identifies the type of library, the directory name, and the library name. The file names have the following format:

<type of library>/<directory name>/1<library name>

The following table explains the components of the file name syntax:

Component	Explanation
<type of library>	Identifies the file as a form library (FORMLIB), a text library (TEXTLIB), or a help library (HELPLIB).
<directory name>	Identifies the directory in the dictionary where the libraries are described.
1	Assigned by SDF Plus as a delimiter. (The number 1 precedes each library name node, but is not part of the library name.)
<library name>	Identifies the SDF Plus library name for the product.

For example, FORMLIB/KEYEDIOII/1KEYEDIOII is a form library. The first node identifies the library as a form library. The second node is the directory name, KEYEDIOII. The third node is the library name, KEYEDIOII, preceded by the digit 1.

An overview of the procedures to localize any of the products listed in Table 5-7 is provided under "Localizing Screens with SDF Plus" earlier in this section. For more complete instructions, refer to the *SDF Plus Installation and Operations Guide*.

Table 5-7. Unisys Products Localized with SDF Plus

Product	Files
COBOL Migration Tool (CMT)	FORMLIB/CMT/1CMT HELPLIB/CMT/1CMT
COBOL74LOADER	FORMLIB/COBOL74LOADER/1COBOL74LOADER HELPLIB/COBOL74LOADER/1COBOL74LOADER
InfoExec ADDS	FORMLIB/ADDS/1ADDS FORMLIB/ADDS2/1ADDS2 FORMLIB/ADDSUTIL/1ADDSUTIL FORMLIB/IEMANAGER/1IEMANAGER HELPLIB/ADDS/1ADDS HELPLIB/ADDS2/1ADDS2 HELPLIB/ADDSUTIL/1ADDSUTIL HELPLIB/IEMANAGER/1IEMANAGER
InfoExec OCM	FORMLIB/OCM/1OCM FORMLIB/DSGL/1DSGL HELPLIB/OCM/1OCM HELPLIB/DSGL/1DSGL
Interactive Configurator (IC)	FORMLIB/ICONFIG/1ICONFIGLIB HELPLIB/ICONFIG/1ICONFIGLIB
ISORT	FORMLIB/ISORT/1ISORT HELPLIB/ISORT/1ISORT
KEYEDIOII utility	FORMLIB/KEYEDIOII/1KEYEDIOII
ODBC Access Drivers	FORMLIB/INFOACCESS/1AUTILITY
Remote Database Backup	FORMLIB/RDBUTILITY/1RDBUTILITY TEXTLIB/RDBUTILITY/1TEXTLIB HELPLIB/RDBUTILITY/1TEXTLIB
SDF Plus	FORMLIB/SDFPLUS/1FEFORMS FORMLIB/SDFPLUS/1FMFORMS FORMLIB/SDFPLUS/1SDFPLUSHELL FORMLIB/SDFPLUS/1UTILITIES TEXTLIB/SDFPLUS/1FETEXTLIB HELPLIB/SDFPLUS/1FEHELP HELPLIB/SDFPLUS/1FPHELP HELPLIB/SDFPLUS/1FMHELP HELPLIB/SDFPLUS/1WSFPHELP

Localizing Workstation Products

Unisys provides workstation products on several software platforms. You can localize some of the Unisys workstation products that are provided for the Microsoft Windows software platforms.

Table 5–8 lists the Unisys workstation products that can be localized and the software platform of the product.

Table 5–8. Unisys Workstation Products That Can Be Localized

Product	Software Platform
Administration Center	Microsoft Windows
CD-ROM Formatter	Microsoft Windows
COMTI for ClearPath MCP	Microsoft Windows
Database Operations Center	Microsoft Windows
End User Center	Microsoft Windows
Enterprise Output Manager for ClearPath MCP	Microsoft Windows
Installation Center	Microsoft Windows
Java Servlet Manager	Microsoft Windows
Logon Agent	Microsoft Windows
Network Database OLE DB Data Provider for ClearPath OS 2200	Microsoft Windows
NX/View	Microsoft Windows
ODBC Access Drivers	Microsoft Windows
Operations Center	Microsoft Windows
Print Center	Microsoft Windows
Programmer's Workbench for ClearPath MCP	Microsoft Windows
System Management Center	Microsoft Windows
Transaction Center	Microsoft Windows
Workstation Dialog Management (WDM)	Microsoft Windows
XlateEBCDIC	Microsoft Windows

Except for SDF Plus, the localization of these workstation products might require products that Unisys does not provide. Contact your Unisys representative or subsidiary about obtaining a localized version of the software platform and the Unisys product that you want.

Making Translated Files Available

Once you have localized the screens, messages, and help text for a specific natural language, you must make the localized versions accessible to the products and to the users of the products. You can make the localized version accessible to the product by renaming the localized version to the appropriate system names, making it public, and reestablishing its privileges with the appropriate system commands.

These localized versions are available when you select a recognized language. Refer to Section 4, "Establishing an MLS Environment," for information on selecting a language.

When you localize products by using IMG, SDF Plus, MSGTRANS, or the Help Utility, any new files or altered files are saved as private files under your usercode. Also, the files no longer have any system privileges established that they might have had on the release media. For example, a localized compiler file is no longer of type compiler after it is localized. To reestablish it as a compiler, you must use the MC (Make Compiler) system command.

You must perform the following steps to make the file visible to non-privileged users and to reestablish its privileges:

1. Rename the files under your usercode to the appropriate system names.

To rename the files, you could use the CANDE *CHANGE* command, as follows:

```
CHANGE NEW/FILE TO *NEW/FILE
```

You can rename a file as a *SYSTEM file only if you are a privileged user. Depending on the way security is defined on your system, you might have to delete a *SYSTEM file of the same name, if one exists, before you can rename the new file.

If you translated screens by using the SDF Plus Compare feature, use the Copy feature to release changes, because the new, private version is not complete. The resulting UIDL file lists the components and attributes of the form library, help library, or text library in a syntactically complete format, similar to the source code for an application program.

2. Establish the appropriate program privileges and security classifications for the new files.

The original files on the release media have certain program privileges and security classifications that the new files also must have in order to be available for system wide use. To reestablish the original settings for each file, you might have to use one or more of the following system commands or CANDE commands:

- CANDE *SECURITY* command
- CP (Control Program) system command
- MC (Make Compiler) system command
- PP:PU (Privileged Program: Privileged Usercode) system command
- PP:T (Privileged Program: Transparent) system command
- PP:SECADMIN (Privileged Program: Security Administrator) system command

- SL (Support Library) system command
- XP (Executable Program) system command

To make your localized software operational, you can use the appropriate commands in the Simple Installation (SI) program. Explanations of the SI program commands are provided in the *Simple Installation (SI) Operations Guide*.

If you do not know the program privileges or security classifications of the original files, ask your system administrator for assistance.

Section 6

Creating Application Systems in an MLS Environment

This section discusses the way to incorporate internationalization features in application systems that consist of application programs and user interfaces. The section contains information that identifies the methods and tools you can use to design and develop localized components for application systems.

Designing Application Programs to Use the Internationalization Features

The following compilers enable you to develop application programs that support coded character sets, ccsversion, natural languages, and business and cultural conventions:

- ALGOL
- C
- COBOL74
- COBOL85
- NEWP
- Pascal
- RPG Compiler

Programs using these compilers can access the internationalization features through calls to the CENTRALSUPPORT library and, in some cases, through constructs provided in the programming language. The following text discusses, in general terms, compiler support for the internationalization features. For specific information on calling CENTRALSUPPORT library procedures from your program or for using other constructs of the programming language, refer to the programming reference manual for the compiler you are using. Refer to Section 8, "Using C Programs in an MLS Environment," when you are using the C compiler.

Note that in some languages you must designate within your program that the internationalization features are to be used.

The following text describes the way to design your program to use the following features:

- Coded character sets and ccsversions
- Natural languages
- Business and cultural conventions

Note: *In the following text, the ALGOL name is used to identify the CENTRALSUPPORT library procedures in the tables. In other programming languages, the name used to refer to the procedure might be different. For example, COBOL74 must call the CNV_DISPLAYMODEL_COB procedure instead of the CNV_DISPLAYMODEL procedure. Refer to the programming reference manual for the language that you are using for the name of the procedure appropriate for that language.*

Providing Support for Coded Character Sets and Ccsversions

Your application program can incorporate features that enable users to enter and manipulate data in any one coded character set and ccsversion. The following constructs within the program can use characters from any one coded character set:

- The contents of string literals, that is, any characters enclosed within the string delimiters defined by the programming language you are using
- Data items containing variable character data
- Comments

For a list of the available coded character sets, refer to Section 9, “Coded Character Sets and Coded character set mapping tables.”

The compilers require that you write all constructs, other than the ones in the preceding list, in the subset of the standard EBCDIC character set defined by the programming language. Refer to the appropriate programming reference manual for more information.

Tables 6-1 through 6-7 provide information on the CENTRALSUPPORT library procedures that support coded character sets and ccsversions. Within each table, the procedures are listed in alphabetical order.

Identifying the Coded Character Set and Ccsversions Available on a Host System

The host system on which your program runs has several coded character sets and ccsversions residing in a file. However, only one coded character set and ccsversion can be active on the system at a time. Table 6-1 lists the CENTRALSUPPORT library procedures that your program can call to determine

- The names and numbers of the coded character set and ccsversions available on the system
- The system default settings for the coded character set and ccsversion
- The availability of a designated coded character set or ccsversion

Table 6-1. Identifying the Available Coded Character Sets and Ccsversions

Procedure Name	Purpose
<p>CCSVSN_NAMES_NUMS</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns the names and numbers of all coded character sets or all ccsversions available on the host system.</p> <p>The names and numbers are listed in two arrays. These arrays are ordered so that the names in the names array correspond to the numbers in the numbers array.</p> <p>With this procedure, you can create a menu that lists the ccsversions on the host system from which a site manager can choose, or you could verify that the ccsversion to be used by a program is available on the host system.</p>
<p>CENTRALSTATUS</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Obtains the values of the default settings for internationalization features on the host system. This procedure returns the names of the default ccsversion, language, convention, and HostCCS. It also returns the number of the default ccsversion.</p> <p>With this procedure, you can inquire about the default settings on the host system.</p>

Table 6-1. Identifying the Available Coded Character Sets and Ccsversions

Procedure Name	Purpose
<p>VALIDATE_NAME_RETURN_NUM</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Verifies that a designated coded character set or ccsversion name is valid on the host system. If the coded character set or the ccsversion is valid, the procedure returns the number for the designated coded character set or ccsversion.</p> <p>With this procedure, you can obtain the ccsversion number needed as a parameter in other calls to CENTRALSUPPORT library procedures.</p>
<p>VALIDATE_NUM_RETURN_NAME</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Verifies that a designated coded character set or ccsversion number is valid on the host system. If the coded character set or the ccsversion is valid, the procedure returns the name of the designated coded character set or ccsversion.</p> <p>Given its number, you can display the name of a coded character set or ccsversion with this procedure.</p>

Obtaining Information about a Coded Character Set or a Ccsversion

To obtain information for a designated coded character set or ccsversion, your program can call one of the CENTRALSUPPORT library procedures listed in Table 6-2. These procedures enable you to format and display data correctly for the coded character set or ccsversion.

Table 6-2. Obtaining Coded Character Set and Ccsversion Information

Procedure Name	Purpose
<p>CCSINFO</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Provides basic information about a designated coded character set, including the number of bits per character (8, 16, or mixed, multibyte), the coding format (for example, ISO or STDEBCDIC), and the space character.</p> <p>With this procedure, you can determine the code format of a file for which the INTMODE attribute is set to a coded character set number.</p> <p>If the coded character set is mixed, multibyte, the following additional information is returned in the array parameter:</p> <ul style="list-style-type: none"> • The number of subccses • The library CCS number of each subccs • The number of bits per character of each subccs • The invocation type of each subccs • The space character of each subccs <p>Note: The caller must pass in a larger array to obtain the additional information. The recommended array size is 30 words. If the caller does not want extension facilities or subccs information, the recommended array size is five words.</p>
<p>VSNINFO</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Returns the following information for a designated ccsversion:</p> <ul style="list-style-type: none"> • The number of the base coded character set to which the ccsversion applies • The escapement information • The characters in the spaces data class for the ccsversion • The array sizes required by other ccsversion elements <p>With this procedure, you can determine the base coded character set for the ccsversion.</p>

Processing Data in an MLS Environment

You can use the CENTRALSUPPORT library procedures listed in Tables 6-3 through 6-5 to enable your program and its users to manipulate textual data. Your program can call these procedures to map, sort, compare, and position text.

Mapping Data for Coded Character Sets

The procedures listed in Table 6-3 use coded character set mapping tables to map characters between two coded character sets.

Eight-bit coded characters sets typically use ALGOL translate tables for mapping. Many CENTRALSUPPORT library procedures store coded character set and ccversion information in this type of mapping table as a way of defining, processing, and mapping 8-bit data. It is not necessary to understand the layout of an ALGOL translate table because the table usually is not visible to your program. A description of ALGOL translate tables is provided in the *ALGOL Programming Reference Manual, Volume 1: Basic Implementation*.

Mixed, multibyte and 16-bit coded character sets must use the CCSTOCCS_TRANS_TEXT_COMPLEX procedure for mapping. This procedure is hereafter referred to as the Complex procedure. Using coded character set numbers as input, this procedure uses mapping tables and translation rules to map the data from one coded character set to another. The Complex procedure also supports mapping for 8-bit coded character sets. For further details, refer to "Complex Procedure" later in this section and to Appendix C, "Using the Complex Procedure."

Table 6-3. Mapping Data from One Coded Character Set to Another

Procedure Name	Purpose
<p>CCSTOCCS_TRANS_TABLE</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Returns a mapping table that maps data encoded in a designated coded character set to data encoded in another designated coded character set. The result is a one-to-one mapping of the characters.</p> <p>You can retain the mapping table in memory and use it with subsequent calls to the TRANS_TEXT_USING_TTABLE procedure. If the program must perform repeated mappings, retaining the table in memory decreases the time required to perform the actual mapping.</p> <p>In an ALGOL or NEWP program, you can also use the translate table directly with the REPLACE syntax.</p> <p>For mixed, multibyte coded character set mappings, use the CCSTOCCS_TRANS_TEXT_COMPLEX procedure.</p>

Complex Procedure

The Complex procedure enables mapping to be spread over multiple calls for processing simple or complex coded character sets. Coded character sets with 16-bit or mixed, multibyte data must be translated through this procedure in CENTRALSUPPORT. The translate tables that support these complex mappings cannot be accessed directly or used by other programs. The procedure maintains static indices to the CENTRALSUPPORT tables, so that initialization overheads can be eliminated on subsequent calls, and maintains previous-operation status information, so that multiple calls can be used to process mixed, multibyte or double-byte source data streams efficiently.

Option Parameter

The Option parameter enables you to optimize how the Complex procedure performs coded character set mapping, both simple and complex. The parameter controls whether the indices to the CENTRALSUPPORT tables are initialized or reinitialized between calls and if the previous-operation status information is used. Table 6-4 lists and describes the values of the Option parameter.

Table 6-4. Option Parameter Values

Values	Effect on Complex Procedure
CS_OPT_INITIAL_COMPLETEV (0)	<p>Initializes the state indices and previous operation status, and then maps the data.</p> <p>Represents the first call for a combination of source and destination coded character sets. Used when the source data stream is complete and the data can be mapped in one call.</p>
CS_OPT_COMPLETEV (5)	<p>Uses existing state indices, ignores previous operation status, and then maps the data.</p> <p>Represents a new source data stream, but is not the first call for a given combination of source and destination coded character sets. Used when the source data stream, which is not part of a prior call, is complete and the data can be mapped in one call.</p>
CS_OPT_INITIAL_HEADV (2)	<p>Initializes the state indices and previous operation status, and then maps the data.</p> <p>Represents the first call for a combination of source and destination coded character sets. Used when the source data stream is incomplete or the destination is not large enough to handle all of the data in one call.</p>
CS_OPT_HEADV (7)	<p>Uses existing state indices, ignores previous operation status, and then maps the data.</p> <p>Represents the beginning of a new source data stream, but is not the first call for a given combination of source and destination coded</p>

Table 6-4. Option Parameter Values

Values	Effect on Complex Procedure
	character sets. Used when the source data stream is incomplete or the destination is not large enough to handle all of the data in one call.
CS_OPT_MIDDLEV (3)	Uses existing state indices and previous operation status, and then maps the data. Used for the second or higher call for a source data stream, but does not represent the last call.
CS_OPT_TAILV (1)	Uses existing state indices and previous operation status, and then maps the data. Used for the second or higher call for a source data stream and does represent the last call.

Refer to Appendix C, "Using the Complex Procedure," for further details and implementation strategies.

Processing Data Based on Ccsversion Information

The procedures listed in Table 6-5 use mapping tables and data classes to process coded character set data according to the rules of a designated ccsversion.

A mapping table is used to map one group of characters to another group of characters or to another representation of the original characters. Many CENTRALSUPPORT library procedures store coded character set and ccsversion information in a type of mapping table called an ALGOL translate table as a way of defining, processing, and mapping data. For example, a mapping table can exist to map lowercase characters to uppercase characters. A *data class* is a group of characters sharing common attributes such as alphabetic, upon which membership tests can be made. A *truthset* is a method of storing the declared set of characters, upon which membership tests can be made. Many CENTRALSUPPORT library procedures store ccsversion information in ALGOL truthset tables as a way to define ccsversion data classes. The internationalization features provide you with access to truthsets that apply to a ccsversion.

Table 6–5. Processing Data According to a Ccsversion

Procedure Name	Purpose
<p>INSPECT_TEXT_USING_TSET</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Compares the input text to the data class to determine whether all the characters in the text are in the data class. This procedure takes a truthset as input. You can retrieve the truthset that determines the data class by calling the VSNTRUTHSET procedure.</p> <p>You can use this procedure to determine if characters are in one of the following data classes:</p> <ul style="list-style-type: none"> • Alphabetic • Numeric • Space • Presentation • Lowercase • Uppercase <p>With this procedure, you could verify that a numeric field contains only digits by comparing characters in the field to the members of the numeric data class.</p>

Table 6–5. Processing Data According to a Ccsversion

Procedure Name	Purpose
<p>TRANS_TEXT_USING_TTABLE</p> <p>Programming Languages:</p> <p>C Pascal</p>	<p>Uses a mapping table to map data according to a designated ccsversion. This procedure takes a mapping table as input. You can retrieve the mapping table by calling the VSNTRANSTABLE procedure. The table you use determines the type of mapping done. All mappings are one-to-one mappings of the characters.</p> <p>You can use this procedure to perform the following types of mappings:</p> <ul style="list-style-type: none"> • Lowercase to uppercase characters • Uppercase to lowercase characters • The digits 0 through 9 to alternate digits • Alternate digits to the digits 0 through 9 • Characters to their character escapement direction values <p>With this procedure, you could map a text string from lowercase letters to uppercase letters to further parse the string.</p>
<p>VSNINSPECT_TEXT</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Compares the input text to a designated ccsversion data class to determine whether the characters in the text are in the data class. You can use this procedure to determine if characters are in one of the following data classes:</p> <ul style="list-style-type: none"> • Alphabetic • Numeric • Space • Presentation <p>With this procedure, you could verify that a numeric field contains only digits by comparing characters entered in the field to members of the numeric data class.</p>

Table 6–5. Processing Data According to a Ccsversion

Procedure Name	Purpose
<p>VSNTRANS_TEXT</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Uses a designated ccsversion to map text.</p> <p>You can use the procedure to perform the following types of mappings:</p> <ul style="list-style-type: none"> • Lowercase to uppercase characters • Uppercase to lowercase characters • The digits 0 through 9 to alternate digits • Alternate digits to the digits 0 through 9 • Characters to their character escapement directions <p>With this procedure, you could map a text string from lowercase letters to uppercase letters, or process an input string by applying the ccsversion escapement rules to it.</p>
<p>VSNTRANSTABLE</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Returns a mapping table for a designated ccsversion. The type of mapping table you request depends on the mapping to be performed.</p> <p>You can use mapping tables to perform the following types of mappings:</p> <ul style="list-style-type: none"> • Lowercase to uppercase characters • Uppercase to lowercase characters • The digits 0 through 9 to alternate digits • Alternate digits to the digits 0 through 9 • Characters to their character escapement directions <p>You can retain the mapping table in memory to be used with subsequent calls to the TRANS_TEXT_USING_TTABLE procedure. If the program must perform repeated mappings, retaining the table in memory decreases the time required to perform the actual mappings.</p> <p>You can improve program performance by using this combination of procedures instead of multiple calls to the VSNTRANS_TEXT procedure. The VSNTRANS_TEXT procedure performs the same function as the combination of the TRANS_TEXT_USING_TTABLE and VSNTRANSTABLE procedures. The procedure attempts to locate the mapping table each time it is called.</p> <p>In an ALGOL or a NEWP program, you can also use the translate table with the REPLACE syntax.</p>

Table 6–5. Processing Data According to a Ccsversion

Procedure Name	Purpose
<p>VSNTRUTHSET</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Returns a data class for the designated ccsversion. The data class contains the characters in a given data class for the ccsversion. Data classes are available for the following data classes:</p> <ul style="list-style-type: none"> • Alphabetic • Numeric • Space • Presentation • Lowercase • Uppercase <p>You can retain the data class in memory to be used with subsequent calls to the INSPECT_TEXT_USING_TSET procedure. If the program must perform repeated inspections, retaining the table in memory decreases the time required to perform the actual inspections.</p> <p>You can improve program performance by using this combination of procedures instead of multiple calls to the VSNINSPECT_TEXT procedure. The VSNINSPECT_TEXT procedure performs the same function as the combination of the INSPECT_TEXT_USING_TSET and VSNTRUTHSET procedures. The procedure attempts to locate the truthset each time it is called.</p> <p>In an ALGOL or a NEWP program, you can also use the truthset with the SCAN syntax.</p>

Comparing and Sorting Text

Historically, text was sorted by using a standard system-provided method that is based on a strict binary comparison of the character values. Within a program, you might also specify a collating sequence to be used for text comparisons.

Often with international coded character sets, the binary method of comparison does not result in a meaningful ordering of the text to be sorted. This situation occurs when the binary ordering of the coded characters does not match the ordering sequence of the alphabet. Your program can call the CENTRALSUPPORT library procedures listed in Table 6-6 to obtain the ordering information of the ccsversion, and to sort or compare text based on this information.

Table 6-6. Comparing and Sorting Text

Procedure Name	Purpose
<p>COMPARE_TEXT_USING_ORDER_INFO</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Compares two strings by using the ordering information from the ccsversion. The ordering information can be obtained by calling the VSNORDERING_INFO procedure.</p> <p>You can choose one of the following comparison methods:</p> <ul style="list-style-type: none"> • Equivalent comparison This method is based on the ordering sequence values (OSVs) of the characters. • Logical comparison This method is based on the OSVs and the priority sequence values (PSVs) of the characters. See the example at the end of the table. <p>With this procedure, you can compare a list of names to determine the correct ordering based on the designated comparison method.</p>
<p>VSNCOMPARE_TEXT</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Compares two strings by using one of three comparison methods for a designated ccsversion. You can select one of the following comparison methods:</p> <ul style="list-style-type: none"> • Binary comparison This method is based on the binary values of the characters. • Equivalent comparison This method is based on the OSVs of the characters. • Logical comparison This method is based on the OSVs and PSVs of the characters. See the example at the end of the table. <p>With this procedure, you can compare a list of names to determine the correct ordering based on the designated comparison method.</p>

Table 6–6. Comparing and Sorting Text

Procedure Name	Purpose
<p>VSNGETORDERINGFOR_ ONE_ TEXT</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns the ordering information of the input text. The ordering information includes the OSVs and PSVs of the characters and any substitution of characters to be made when the input text is sorted. You can choose one of the following types of information:</p> <ul style="list-style-type: none"> • Equivalent ordering information Characters are translated to their OSVs. • Collating information Characters are translated to their OSVs followed by the PSVs. <p>With this procedure, you can transform a text string into its OSVs and perform multiple comparisons within your program.</p>
<p>VSNORDERING_INFO</p> <p>Programming Languages:</p> <p>ALGOL C NEWP Pascal</p>	<p>Returns the ordering information for a designated ccsversion. The ordering information determines the way data is collated for the ccsversion. It includes the OSVs and PSVs of the characters and any substitution of characters to be made when the designated ccsversion ordering is applied to a string of text.</p> <p>You can retain the ordering information in memory to be used with subsequent calls to the COMPARE_TEXT_USING_ORDER_INFO procedure. If the program must collate a large amount of data, retaining the ordering information in memory decreases the time required to perform the actual collating.</p> <p>You can improve program performance by using this combination of procedures instead of multiple calls to the VSNCOMPARE_TEXT procedure. The VSNCOMPARE_TEXT procedure performs the same function as the combination of the COMPARE_TEXT_USING_ORDER_INFO and VSNORDERING_INFO procedures. This procedure attempts to locate the ordering information each time it is called.</p>

In a logical comparison, the OSVs for the entire length of the strings are compared first. If the OSVs are equal for the entire length of the strings, then the PSVs are examined to determine the order.

		OSVs	PSVs
String1	ba	130, 129	02, 02
String2	Be	130, 133	01, 02
String3	be	130, 133	02, 02

Figure 6-1. Logical Comparison Method

String1 has a lower value than String2 because, when the OSVs are compared, the letter *a* (129) is lower in value than the letter *e* (133). Thus, *ba* appears before *Be*. String2 has a lower value than String3 because, after the OSVs are found to be equal, the PSVs indicate that the capital letter *B* (01) precedes the small letter *b* (02).

Positioning Text Based on Escapement Rules

You can use the procedure listed in Table 6-7 to position text according to the escapement rules specified in a designated ccsversion.

Table 6-7. Positioning Characters

Procedure Name	Purpose
VSNESCAPEMENT Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Takes the input text and rearranges it according to the escapement rules of the ccsversion. Both the character advance direction and the character escapement direction are used. If the character advance direction is positive, then the starting position for the text is the leftmost position of the starting character. If the character advance direction is negative, then the starting position for the text is the rightmost position of the last character. From that point on, the character advance direction and the character escapement direction values, in combination, control the placement of each character in relation to the previous character. With this procedure, you could format a bi-directional text string.

Providing Support for Natural Languages

The ability of your program to process data in several coded character sets enhances support for several natural languages. Your program can also provide for display of output messages to users in their natural languages and for interpretation of messages received from the user or another program in a natural language different from the base language of the program.

Determining the Available Languages on the Host System

The CENTRALSUPPORT library procedure described in Table 6-8 enables your program to determine the languages that are bound to the MCP. Your program can use one of these languages, or it can choose a language that is not bound to the MCP.

Table 6-8. Determining the Available Natural Languages

Procedure Name	Purpose
MCP_BOUND_LANGUAGES Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Returns the names of languages that are currently bound to the MCP.

Creating Input and Output Messages That Can Be Localized

In the MLS environment, the messages handled by your application program are grouped into the following categories:

Message Category	Explanation
Output message	A message that an application program displays to the user. Some examples of output messages are error messages and prompts for input. An output message is localized so that it can be displayed in the language of the user.
Input message	A message received by an interactive program either from a user or from another program in response to a prompt for input. The input message might be in a language that the program cannot recognize. In this case, the message must be translated so that it can be understood by the program.

If you code the input and output messages for your program using the OUTPUTMESSAGE ARRAY syntax, you make the localization process easier. When messages are in an output message array, the translator can use MSGTRANS to localize the messages. Translation at the source code level is not necessary.

You can create an output message array by using the OUTPUTMESSAGE ARRAY declaration in an ALGOL or NEWP program. You can then use certain ALGOL or NEWP statements to display output messages or translate input messages. For more information on using these statements, refer to "Processing Messages in an ALGOL or NEWP Program" later in this section.

If your application program is written in C, COBOL74, COBOL85, Pascal, or RPG Compiler, you can still use the ALGOL or NEWP output message array to handle input and output messages. To do so, you must create an ALGOL or NEWP library containing the output message array and the message-handling syntax. You can then call the library from your application program. For more information on this procedure, refer to "Processing Messages in C, COBOL74, COBOL85, Pascal, and RPG Compiler Programs" later in this section.

The following paragraphs describe the procedures for creating and using output message arrays.

Creating an Output Message Array

You can create an output message array by creating an ALGOL program or library that contains OUTPUTMESSAGE ARRAY declarations.

An output message array contains messages to be used by MLS. ALGOL statements within the program return output messages or translate input messages. See the *ALGOL Programming Reference Manual, Volume 1: Basic Implementation* for a description of the procedures for creating and using output message arrays. Additionally, the EXAMPLE/MLS/ALGOL/LIBRARY program on the release media demonstrates the way to create an ALGOL library containing output message arrays.

Example 6-2 shows a sample of the OUTPUTMESSAGE ARRAY declaration called ERRORS. In this example, the message array contains corresponding messages in English and in French. The English messages use the ASeriesNative ccsversion and the French messages use the FRANCE ccsversion.

```
OUTPUTMESSAGE ARRAY ERRORS (  
  ENGLISH  
    <"This help text appears with all English messages "  
      "in MSGTRANS.">  
  
  USEBCDIC  
  
  (  
    10 = <"This help text appears only with English"  
      "message 10 in MSGTRANS.">  
      "Positive integer expected.",  
  
    20 = "Too many parameters."  
  ),  
  FRANCAIS FRANCE (  
    10 = "Demande un entier positif.",  
    20 = "Trop de param-tres."  
  ));
```

Figure 6–2. Sample Output Message Array

Processing Messages in an ALGOL or NEWP Program

The ALGOL and NEWP compilers provide statements that enable you to process the messages in an output message array. The following table lists the statements you can use to display output messages in the appropriate language and ccsversion for the user and to translate input messages into the language understood by the application program. For a complete description of the syntax and use of these statements, refer to the *ALGOL Reference Manual, Vol. 1*.

Statement	Purpose
MESSAGESEARCHER	Searches a designated output message array for the specified message number, and returns the message in the appropriate language and ccsversion.
MLSACCEPT	Displays or prints an output message and causes the program to wait for input, or returns a Boolean value indicating whether or not a message is waiting for the program.
MLSDISPLAY	Displays a message on the system console, and prints the message in the job summary listing.
MLSTRANSLATE	Compares an input message to the messages in a designated output message array. If the input message matches one of the messages in the array, the translated message is returned to the program. If the input message does not match any of the messages in the array, an error is returned. Note that input messages cannot contain parameters. They must be simple EBCDIC or hexadecimal strings.

Processing Messages in C, COBOL74, COBOL85, Pascal, and RPG Compiler Programs

Although the syntax for creating and accessing messages is supplied by the ALGOL and NEWP compilers, you can access localized messages from C, COBOL74, COBOL85, Pascal, and RPG Compiler programs. To access the messages, you must include the output message array and the appropriate syntax for processing the messages in an ALGOL or NEWP library. You then call this library as you would any other library.

Refer to Appendix B, "Examples of Using Libraries to Access Output Message Arrays," for examples of ALGOL and NEWP libraries and of programs calling these libraries.

Accessing Localized CENTRALSUPPORT Library Messages

If your program calls the CENTRALSUPPORT library, you might want to access the text messages that correspond to the possible errors returned by the library. Table 6-9 describes the procedure you use to access these messages. Note that these messages are contained in an output message array in the CENTRALSUPPORT library and can be localized. Refer to Appendix A, "CENTRALSUPPORT Library Messages," for a list of the messages.

Table 6-9. Accessing CENTRALSUPPORT Library Messages

Procedure Name	Purpose
<p>GET_CS_MSG</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns the message associated with a designated error number and language from the CENTRALSUPPORT library. Your program can specify the maximum message length desired. If the returned message is shorter, it is padded with blanks.</p> <p>An entire message consists of three parts:</p> <ul style="list-style-type: none">• The header, which always takes up the first 80 characters of the returned message• The general description, which takes the next 80 characters• The specific description, which has no maximum length

Providing Support for Business and Cultural Conventions

The business and cultural features enable users of an application program to display and receive data according to local conventions. A convention consists of formatting instructions for date, time, numeric, currency, and default page size.

Tables 6-9 through 6-16 describe the procedures your program can call to access information about the convention definitions available on the host system.

Identifying the Convention Definitions Available on a Host System

The host system on which your program runs has several convention definitions residing in a file. Table 6-10 lists the CENTRALSUPPORT library procedures that your program can call to determine

- The names and numbers of the convention definitions available on the system
- The system default settings for the convention definition
- If a designated convention definition is available

Table 6–10. Identifying the Available Convention Definitions

Procedure Name	Purpose
CENTRALSTATUS Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Obtains the values of the default settings for internationalization features on the host system. This procedure returns the names of the default ccsversion, language, convention, and HostCCS. It also returns the number of the default ccsversion. With this procedure, you can inquire about the default settings on the host system.
CNV_NAMES Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Returns the names of the conventions available on the host system. With this procedure, you can create a menu that lists the conventions on the host system from which a site manager can choose, or you could verify that the convention to be used by a program is available on the host system.
CNV_VALIDATENAME Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Verifies that a designated convention is currently defined on the host system. With this procedure, you can verify that the convention established for the program is valid on the host.

Obtaining Information about Convention Definitions

The host system on which your program is run might have both standard and custom convention definitions available. The procedures listed in Table 6-11 describe CENTRALSUPPORT library procedures that your program can call to access

- All the elements in a designated convention
- The formatting templates or symbols defined for dates, times, numeric values, or monetary values

Table 6–11. Obtaining Information about Conventions

Procedure Name	Purpose
CNV_INFO Programming Languages: ALGOL C NEWP Pascal	Returns a description of all the elements in a designated convention definition. This procedure can retrieve all formatting templates for use with subsequent procedure calls.
CNV_SYMBOLS Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Returns the numeric and monetary symbols defined for a designated convention. The symbols in the convention are <ul style="list-style-type: none"> • Numeric positive symbol • Numeric negative symbol • Numeric thousands separator symbol • Numeric decimal symbol • Numeric left enclosure symbol • Numeric right enclosure symbol • Numeric grouping specifications • Monetary positive symbol • Monetary negative symbol • International currency notation • National currency symbol • Monetary thousands separator symbol • Monetary decimal symbol • Monetary left enclosure symbol • Monetary right enclosure symbol • Monetary grouping specifications With this procedure, you can retrieve the international currency notation for use in your program.

Table 6–11. Obtaining Information about Conventions

Procedure Name	Purpose
CNV_TEMPLATE Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	<p>Returns the requested formatting template for a designated convention. You can obtain the template for the following:</p> <ul style="list-style-type: none"> • Long date format • Short date format • Numeric date format • Long time format • Numeric time format • Monetary editing format • Numeric editing format <p>To improve performance, you can use the CNV_TEMPLATE procedure with procedures that have names with a TMP extension. This combination of procedures can be used as an alternative to calling a procedure that does not take a template as input. Procedures that do not take templates as input attempt to locate a template each time the procedure is called.</p>

Formatting Dates According to a Convention

The procedures listed in Table 6-12 enable your program to format dates according to a designated convention or by using a template defined within the program. The procedures also enable a program to process formatted dates.

Table 6-12. Formatting Dates According to a Convention

Procedure Name	Purpose
CNV_CONVERTDATE Programming Languages: ALGOL C NEWP Pascal	Converts a formatted numeric date, passed as a parameter to the procedure, to the format YYYYMMDD. The numeric date must be formatted according to the numeric date format template defined in the designated convention.
CNV_DISPLAYMODEL Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Returns either the date or the time display model defined for the designated convention. The components of the model are translated to the designated language.
CNV_FORMATDATE Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Formats a date that has the form YYYYMMDD. The date is passed as a parameter to the procedure, with a designated convention and language. The date can be formatted using the long, short, or numeric date format defined in the convention. If the convention is Belgium, which has a short date template !1w!, !0d! !1n!, !0yyy!, the language is English, and the input date is 19901015, the short date is then formatted Mon, 15 Oct, 1990.
CNV_FORMATDATETMP Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Formats a numeric date, passed as a parameter to the procedure, with a designated template and language. The template can be retrieved by calling the CNV_TEMPLATE procedure. If the date template is !w!, !d! !n! !yyyy!, the input date is 19900315, and the language is French, the long date is then formatted Lundi, 15 Mars 1990.

Table 6–12. Formatting Dates According to a Convention

Procedure Name	Purpose
<p>CNV_SYSTEMDATETIME</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns either the system date, the system time, or both, formatted according to the designated convention and language. You can choose from the following types of templates:</p> <ul style="list-style-type: none"> • Long date and long time • Long date and numeric time • Short date and long time • Short date and numeric time • Numeric date and long time • Numeric date and numeric time • Long date only • Short date only • Long time only • Numeric time only
<p>CNV_SYSTEMDATETIMETMP</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns the system date, the system time, or both, in the designated language, formatted according to a template passed as a parameter to the procedure. The template can be retrieved by calling the CNV_TEMPLATE procedure.</p>

Formatting Times According to a Convention

The procedures listed in Table 6-13 enable your program to format times according to a designated convention or by using a template defined within the program. The procedures also enable a program to process formatted times.

Table 6-13. Formatting Times According to a Convention

Procedure Name	Purpose
CNV_CONVERTTIME Programming Languages: ALGOL C NEWP Pascal	Converts a formatted numeric time, passed as a parameter to the procedure, to the format HHMMSSPPPP. The numeric time must be formatted according to the numeric time format template defined in the designated convention.
CNV_DISPLAYMODEL Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Returns either the date or the time display model defined for the designated convention. The component designators for the model are translated to the designated language.
CNV_FORMATTIME Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Formats a time with the form HHMMSSPPPP. The time is passed as a parameter to the procedure, with a designated convention and language. The time can be formatted using the long or numeric time template defined in the convention. If the convention is Sweden, which has a numeric time template !0t!.!0m!, the language is English, and the input time is 1155300000, the numeric time is then formatted as 11.55.
CNV_FORMATTIMETMP Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Formats a time with the form HHMMSSPPPP, passed as a parameter to the procedure, according to a designated template and language. The template can be retrieved by calling the CNV_TEMPLATE procedure. If the time template is !0t!:!0m!:0s!, the language is English, and the input time is 1255016256, the numeric time is formatted as 12:55:01.

Table 6–13. Formatting Times According to a Convention

Procedure Name	Purpose
<p>CNV_SYSTEMDATETIME</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns either the system date, the system time, or both, formatted with the designated convention template and language. You can choose from the following types of templates:</p> <ul style="list-style-type: none"> • Long date and long time • Long date and numeric time • Short date and long time • Short date and numeric time • Numeric date and long time • Numeric date and numeric time • Long date only • Short date only • Long time only • Numeric time only
<p>CNV_SYSTEMDATETIMETMP</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Returns the system date, the system time, or both, in the designated language, formatted according to a template passed as a parameter to the procedure.</p> <p>The template can be retrieved by calling the CNV_TEMPLATE procedure.</p>

Formatting Numeric Data According to a Convention

The procedure described in Table 6-14 enables your program to convert a formatted numeric value to a real number.

Table 6–14. Formatting Numeric Data According to a Convention

Procedure Name	Purpose
CNV_CONVERTNUMERIC Programming Languages: ALGOL C NEWP Pascal	Converts a string containing digits and numeric symbols to a real number.

Formatting Monetary Data According to a Convention

The procedures listed in Table 6-15 enable your program to format a monetary value according to a designated convention or by using a template defined within the program. The procedures also enable your program to convert a formatted monetary value to a real number.

Table 6–15. Formatting Monetary Data According to a Convention

Procedure Name	Purpose
CNV_CONVERTCURRENCY Programming Languages: ALGOL C NEWP Pascal	Converts a string containing digits and monetary symbols to a real number.
CNV_CURRENCYEDIT Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Formats a monetary value, passed as a parameter to the procedure, according to the monetary editing format template defined in the designated convention. If the convention is Denmark, which has a template !n[-]c[kr.]t[:0,3]d[,]#!, and the input amount is 1234567.89, the currency is then formatted as kr.1.234.567,89.

Table 6–15. Formatting Monetary Data According to a Convention

Procedure Name	Purpose
<p>CNV_CURRENCYEDIT_DOUBLE</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler</p>	<p>Formats a monetary value, passed as a parameter to the procedure, according to the monetary editing format template defined in the designated convention.</p> <p>If the convention is Denmark, which has a template !n[-]c[kr.]t[:0,3]d[.]#!, and the input amount is 123456789 with a precision value of 2, the currency is then formatted as kr.1.234.567,89.</p>
<p>CNV_CURRENCYEDITTMP</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal</p>	<p>Formats a monetary value, passed as a parameter to the procedure, according to a template also passed as a parameter. The template is retrieved with the CNV_TEMPLATE procedure.</p> <p>If the input amount of the monetary value is 1234567.89 and the template is !n[-]c[\$]t[:0,3]d[.]#! the monetary value is then formatted as 1,234,567.89.</p>
<p>CNV_CURRENCYEDITTMP_DOUBLE</p> <p>Programming Languages:</p> <p>ALGOL C COBOL74 COBOL85 NEWP Pascal</p>	<p>Formats a monetary value, passed as a parameter to the procedure, according to a template also passed as a parameter. The template is retrieved with the CNV_TEMPLATE procedure.</p> <p>If the input amount of the monetary value is 123456789 with a precision value of 2 and the template is !n[-]c[\$]t[:0,3]d[.]#! the monetary value is then formatted as 1,234,567.89.</p>

Determining the Default Page Length and Width

You can use the procedure described in Table 6-16 to determine the default settings for page length (lines per page) and page width (characters per line) on the host system.

Table 6-16. Determining the Default Page Length and Width

Procedure Name	Purpose
CNV_FORMSIZE Programming Languages: ALGOL C COBOL74 COBOL85 NEWP Pascal RPG Compiler	Returns the default lines-per-page and characters-per-line values, defined in a designated convention, for presentation devices such as printers. You might use this information to format program output according to the default paper size used at the site.

Adding, Modifying, and Deleting Conventions

You can use the procedures listed in Table 6-17 to add, modify, or delete custom convention definitions.

Table 6-17. Adding, Modifying, and Deleting Conventions

Procedure Name	Purpose
CNV_ADD Programming Languages: ALGOL C NEWP Pascal	Adds a new convention definition to the SYSTEM/CONVENTIONS file. The new definition is available immediately.
CNV_DELETE Programming Languages: ALGOL C NEWP Pascal	Deletes an existing convention definition from the SYSTEM/CONVENTIONS file. The convention is deleted after the next halt/load. Note that this procedure can be used to delete only those convention definitions that you have created. You cannot delete the standard convention definitions provided on the release media.
CNV_MODIFY Programming Languages: ALGOL C NEWP Pascal	Modifies an existing convention definition in the SYSTEM/CONVENTIONS file. The modified convention becomes effective after the next halt/load. Note that this procedure can be used to modify only those convention definitions that you have created. You cannot modify the standard convention definitions provided on the release media.

Designing User Interfaces That Can Be Localized

A user interface encompasses the components within an application system that the application user sees and interacts with. The components of the user interface can include forms, menus, messages, and online help. Creating messages as part of an application program is discussed under “Creating Input and Output Messages That Can Be Localized” earlier in this section. The following text discusses the way to create the other components of the user interface so that they can be localized.

In the MLS environment, SDF Plus is a tool that enables you to create a user interface that uses the internationalization features. The user interface you create with SDF Plus is called a user interface system (UIS). This UIS is composed of a form library and, optionally, a text library and one or more help libraries. The form library contains the forms displayed to the user and the instructions for processing the UIS. The text library contains the messages and other text displayed to the user by the UIS. The help library contains information that is displayed online when the user requests help on a prompt, field, menu choice, or form. All these components can be localized.

The following text provides general information on the SDF Plus features as they relate to the MLS environment. For a more detailed overview of SDF Plus concepts, refer to the *Screen Design Facility Plus (SDF Plus) Technical Overview*. For instructions on using the SDF Plus features described here, refer to the *SDF Plus Installation and Operations Guide*.

Providing Support for Coded Character Sets

SDF Plus supports the use of both single-octet and double-octet characters. This feature enables you to design a UIS based on any of the available coded character sets, plus Kanji. Like the compilers, SDF Plus places certain restrictions on the use of alternate coded character sets. You can use alternate coded character sets for the contents of most string literals.

You must use the standard characters for the following UIS components:

- Entity names
- Constructs of the SDF Plus expression syntax
- Constructs of the SDF Plus Help utility syntax

By creating the UIS with SDF Plus, you can provide support for displaying text that uses alternate coded character sets. You can also handle validation of user input when alternate coded character sets are used.

The user interface components that can be localized include the following components:

- Literal text appearing on the form image. This text includes titles, prompts, menu choices, and explanatory text. These components can be either painted directly on the form image as literal text fields, or defined as alphanumeric output-only fields.
- Values defined for validation of alphanumeric, Boolean, and double-octet fields.

- Field values and variable values defined for substitution tables.
- Mnemonics and return values for form menus.
- Menu choice text for action menus on mainframe terminals.
- Text displayed as validation errors, online help text, or other types of messages.
- Default values for global variables and for transaction field attributes.

Providing Support for Ccsversions

SDF Plus enables you to declare a `ccsversion` to be associated with the UIS. You can designate a `ccsversion` for the form library, the text library, and any help libraries associated with the UIS.

When your UIS is run, the `ccsversion` you designated is compared to the default `ccsversion` on the host system. If they are different, SDF Plus displays a warning message. You still have the option of running the UIS; however, because character mapping might differ between the `ccsversions`, you can get unexpected results.

SDF Plus also uses information from the `ccsversion` to

- Map lowercase characters to uppercase characters.
- Convert alternate digits to standard digits for processing of numeric fields.

Providing Support for Natural Languages

SDF Plus enables you to declare a language as an attribute of a UIS. This language is used to select the appropriate form library, help libraries, and text library at run time based on the language setting of the user.

For more information on creating and localizing the components of a UIS, refer to the *SDF Plus Installation and Operations Guide*.

Providing Support for Business and Cultural Conventions

SDF Plus enables you to designate specific formats to be used when dates or times are displayed. Alternatively, you can designate that dates and time be formatted according to the convention specified by the user.

Section 7

Configuring Unisys Products for Internationalization

Data communication and networking products provide the communications link between a host system and remote devices such as terminals and workstations, or between various host systems.

If you are using a character set other than USEBCDIC, you need to configure the data communications network between the host system and terminals to send and receive data for the character set you are using. For example, to send accented characters between the host system and a PC, you must first establish a coded character set mapping table to map characters between the host and the PC.

Table 7–1 describes the products that have international features for data communications and for networking between a host system and terminals or workstations.

Table 7–1. Comparison of International Features in Data Communications and Networks

DataCom/Network Product	Notes
Client Access Services	Handles data translation when file names, message text, and certain file data are transferred between the MCP environment and internationalized versions of Windows.
CP 2000, CPDLP, MACP100, MACP150, AND CPDLAN (CPDLAN includes ICP22, ICP26, ICP10A, and MAICP when they are configured as built in communication hosts rather than an ICP).	Enables you to reference a system mapping table or create a custom 8-bit mapping table with the translation table feature. These datacom products also support code extension.
HLCN	Enables you to select an 8-bit mapping table easily, but you cannot customize the table for special needs.
NSP/DCDLP, EDCDLP/DCHA	Uses the code extension feature.

Table 7-1. Comparison of International Features in Data Communications and Networks

DataCom/Network Product	Notes
TCP/IP File Transfer Protocol	Permits translation between any pair of 8-bit coded character sets for which translation is provided by CENTRALSUPPORT.
TCP/IP Telnet Services	Permits translation between any pair of 8-bit coded character sets for which translation is provided by CENTRALSUPPORT.

Using the Translation Table and Code Extension Features

The translation table feature enables you to create an 8-bit mapping table between the characters on the host system and the PC. You can create any 8-bit table you want because the tables can be completely customized. The translation table feature is separate from the code extension feature, although many times the features are used together for internationalization. You can specify each feature on a terminal-by-terminal basis. That is, terminals configured with a CP 2000, for example, can use different translation tables and different code extension schemes.

The LAN-Connected Workstations (LCW) protocol is one example of an open systems network that you can localize using the translation table feature. In the Network Administration Utility (NAU), follow the process described in Table 7-2 to send 8-bit data between a PC-based workstation and a host using the LCW protocol. These steps apply to any device that supports code extension or to any terminal for translation tables.

Table 7-2. Configuring BNA Version 2 Networks for Internationalization

Step	Action
1	For each table you need, define a network translation table on the TERMINAL TRANSLATION TABLE LIST by referencing the system mapping tables.
2	Reference the desired network translation table names on the TRANSLATION ASSIGNMENT LIST that is defined for each communication host.
3	Assign the appropriate TranslationTableName and CodeExtensionMode TERMINAL ATTRIBUTES on each terminal or terminal profile by using the TERMINAL ATTRIBUTES (3/4) screen or TERMINAL ATTRIBUTES PROFILE (3/4) screen.

Result

The NAU generates the appropriate OI commands in the communication host initialization file.

For More Information

- The *BNA Version 2 Network Software Implementation Guide, Volume 2: Configuration* provides the complete procedure for the process described in Table 7–2.
- Section 9, “Coded Character Sets and Coded Character Set Mapping Tables,” describes the system mapping tables.

Coded Character Set Mapping with FTP

File Transfer Protocol provides mapping between 8-bit coded character sets during file transfers. FTP implementations map file data to and from the coded character set corresponding to the representation type you use in the transfer.

ClearPath NX and A Series FTP provides the following mapping features for both text and image transfers:

Inbound File Transfers

When you want FTP to receive a file from a remote destination, you can specify the coded character set in which the incoming data stream is encoded and the coded character set in which the disk file data is to be encoded. If CENTRALSUPPORT supports mapping between these two 8-bit coded character sets, the transfer will be performed with the requested mapping. Further, you can specify a third coded character set, which is the value that will be used as the physical EXTMODE of the disk file.

For example, data can be mapped from Latin1ISO to Latin1EBCDIC, but the disk file can be created with EXTMODE=OCTETSTRING.

Outbound File Transfers

When you want FTP to send a file to a remote destination, you can specify the coded character set in which the outgoing data stream is to be encoded and the coded character set in which the disk file data is assumed to be encoded. This second coded character set need not be the coded character set that is represented by the EXTMODE of the disk file. By default, the EXTMODE of the disk file will provide the source coded character set for the requested mapping. However, you are also allowed to specify a substitute coded character set if the file's EXTMODE is one which does not support mapping, such as OCTETSTRING.

For example, if the data in a disk file is encoded as LatinCyrillicEBCDIC, but the EXTMODE of the file is EBCDIC, you can request that the data be mapped from LatinCyrillicEBCDIC to LatinCyrillicISO before transmission. These features are fully described in the *TCP/IP Distributed Systems Services (DSS) Operations Guide*.

Configuring HLCN for Internationalization

Table 7–3 show the steps for configuring HLCN to send 8-bit data between a PC-based workstation and a host. You can use this procedure when you are changing the currently used coded character set value on the PC to the one compatible with the one established on the host system. If you are setting the initial value on the PC, you must use the procedures described in the *Host LAN Connection (HLCN) Installation and Configuration Guide* for the NA HLCNTS ADD/MODIFY TERMINAL/TEMPLATE command.

Table 7–3. Configuring HLCN for Internationalization

Step	Action
1	Select a ccsversion with the SYSTEMOPTIONS system command.
2	Halt/load the system to establish the ccsversion as the system default.
3	Change the currently used coded character set on the PC to the one compatible with the one you established on the host system by entering the following command. ?ATTR CHARSET = <coded character set name> Result: The system loads a coded character set mapping table.

Using Code Extension

NSP/DCDLP and DCDLP/DCHA data communication products primarily use a 7-bit line protocol. However, your site might want to use 8-bit coded character sets.

Poll/select customarily uses the high-order bit of a character for parity, making this bit unavailable for full character definition. As a result, characters that use the high-order bit for definition cannot be transported between devices.

To accommodate the use of 8-bit coded character sets, the Protocols 05 release (the datacom release) includes an extension to the poll/select protocol called *code extension*.

What Is Code Extension?

Code extension is a technique that enables you to transport 8-bit character data through the 7-bit line protocol. Code extension uses special characters to mark the beginning and end of a character or of a string of characters that requires 8 bits per character for full definition.

Who Can Use Code Extension?

In the Protocols 05 release, code extension is implemented for systems that use the enhanced data communications data link processor (EDCDLP) and for Micro A systems that use the data communications host adapter (DCHA) operating with at least a level 76 firmware.

You can also implement code extension by using the *SYSTEM/SOURCENDLII file on systems that use network support processors/line support processors (NSP/LSP). However, the performance of your datacom might be substantially degraded. The implementation of code extension for systems that use NSP/LSP is not part of the Protocols 05 release.

To take advantage of code extension, terminals and printers must be capable of handling 8-bit characters. The terminal must also be capable of recognizing and transmitting the special code-extension characters. The T 27 terminal is capable of handling code extension and supports several coded character sets.

What Are the Code Extension Special Characters?

With code extension, you can choose to use either of the following pairs of characters to mark the beginning and end of each 8-bit character or string of 8-bit characters:

Beginning Character	End Character
SO (shift out)	SI (shift in)
ESC SO (escape shift out)	ESC SI (escape shift in)

Currently, some sites use the SO and SI control characters for code extension. However, some devices use these characters to indicate reverse video and underlining. To allow the use of code extension with devices that already use the SO and SI control characters, the system provides the alternative ESC SO and ESC SI control characters.

You can determine the pair of control characters you want to use on a device-by-device basis. However, you cannot use one control character from each pair simultaneously; that is, you can use the SO control character only with the SI control character, and you can use the ESC SO control character only with the ESC SI control character.

ET and T27 terminals normally use the SO and SI characters to control reverse video and underlining. To recognize alternate codes (such as the ESC 3 and the ESC 4 codes or the ESC' . and the ESC' / codes) as controls for reverse video and underlining, do one of the following:

- Alter your application programs.
- Alter the adapter control in your *SYMBOL/SOURCENDLII file.
- Alter the Editor in your *SYMBOL/SOURCENDLII file.

Altering the Adapter Control to Recognize New Control Characters

If you want to use the SO and SI control characters as your shift out and shift in characters, but you still want the ability to use reverse video and underlining for a terminal, you must alter the Adapter Control of the poll/select algorithm.

Add the following code to the Adapter Control to use the ESC 3 and ESC 4 control characters in place of the SO and SI control characters for reverse video and underlining:

```
TRANSMIT CB.XMTADDR, STX, TEXT UNTIL ETX & SO & SI;
WHILE NOT ENDTEXT DO
  BEGIN
    FETCH C;
    CASE C
      BEGIN
        SO:
          TRANSMIT ESC, 4, TEXT UNTIL ETX & SO & SI;
        SI:
          TRANSMIT ESC, 3, TEXT UNTIL ETX & SO & SI;
      END;    % CASE C
    END;
  END;
```

How Does the Code Extension Process Work?

Two parts to the code extension process exist. They are

- The process on the terminal or printer
- The process on the EDCDLP or DCHA

The following explanation of the code extension process is based on the hardware configuration illustrated in Figure 7-1. The code extension process on a printer is identical to the process on a terminal. And, the process on the DCHA is identical to the process on the EDCDLP. Also, for the purposes of the explanation, the SO and SI control characters indicate code extension.

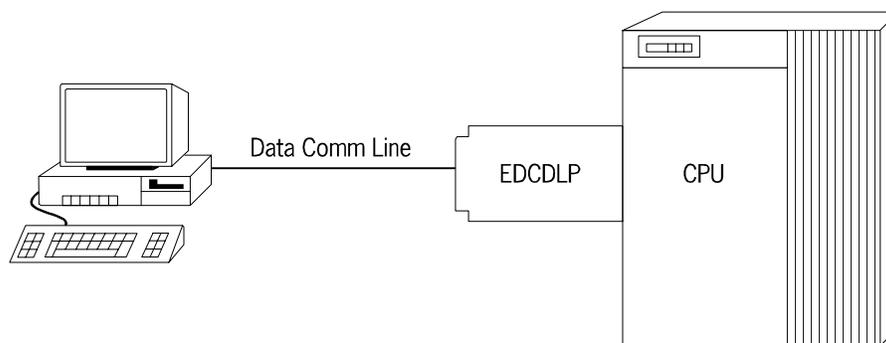


Figure 7-1. Example Hardware Configuration for Code Extension

Sending Data from the Terminal to the Mainframe

The following information explains the code extension process for transferring data from a terminal to the mainframe. The process occurs in two parts. The first part occurs at the terminal, and the second part occurs at the EDCDLP.

The Process at the Terminal for Transmitting Characters

The process illustrated in Figure 7-2 occurs at the terminal when you press the XMT (transmit) key. The process is repeated once for each character transmitted.

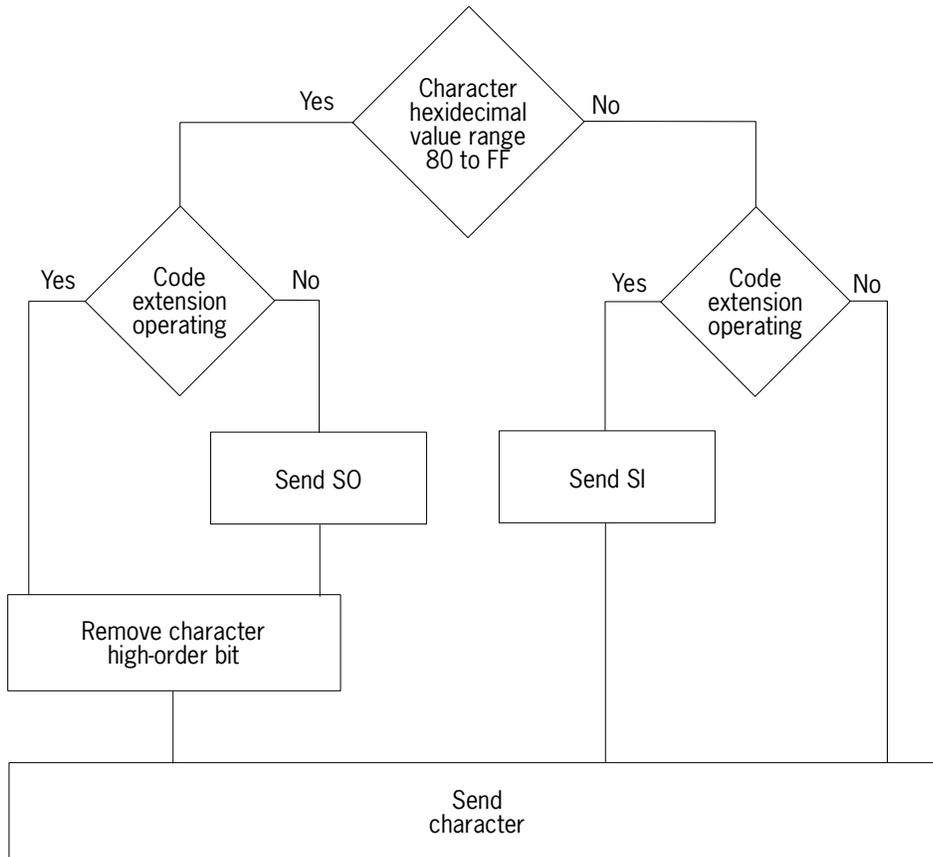


Figure 7-2. Transmitting Data: The Process at the Terminal

The Process at the EDCDLP for Receiving Characters

When data is sent from the terminal to the mainframe, the process illustrated in Figure 7-3 occurs at the EDCDLP. The process is repeated once for each character received.

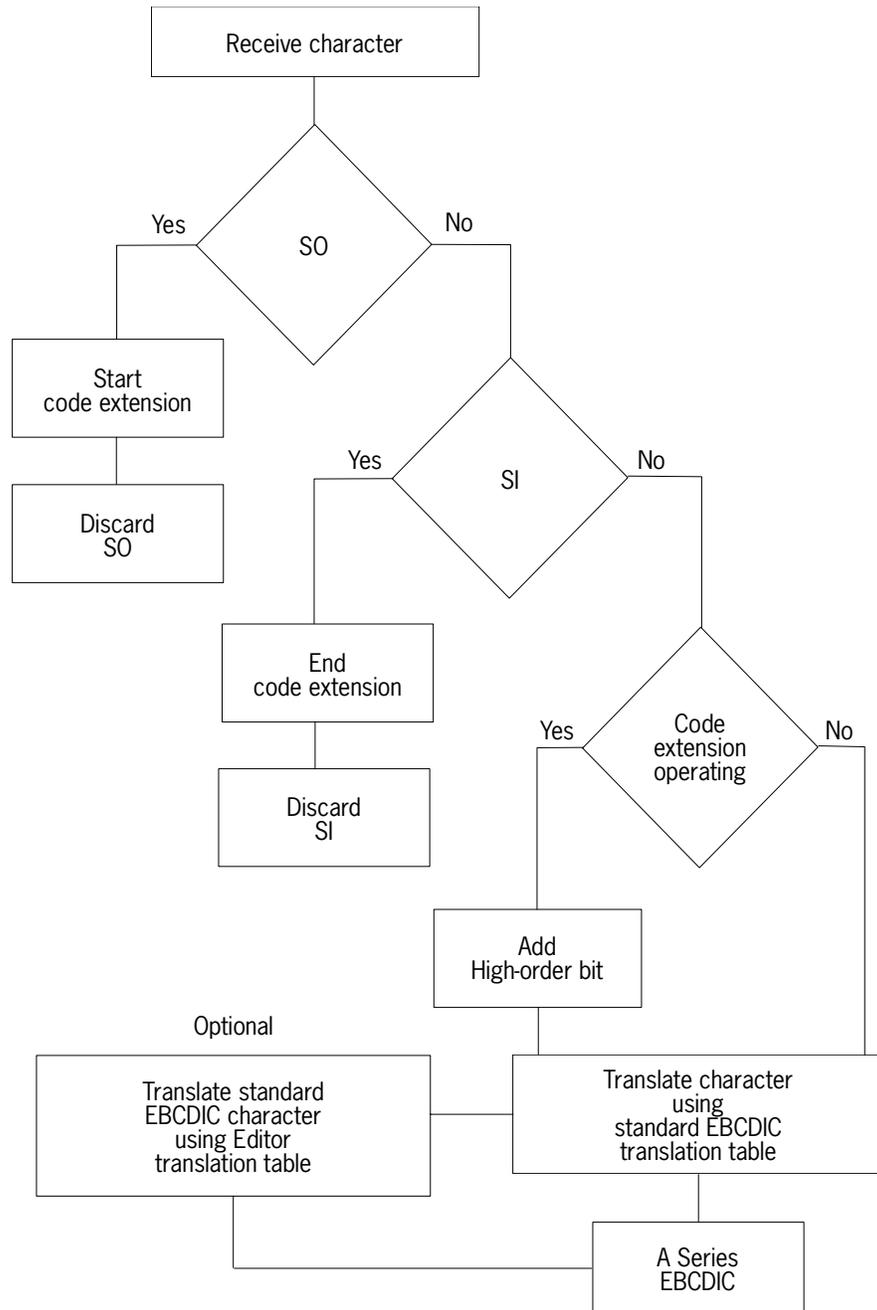


Figure 7-3. Receiving Data: The Process at the EDCDLP

The standard EBCDIC and Editor coded character set mapping tables used in the code extension process are described under “Using Coded Character Set Mapping Tables” later in this section.

Sending Data from the Mainframe to the Terminal

The following information explains the code extension process for transferring data from a terminal to the mainframe. The process occurs in two parts. The first part occurs at the EDCDLP and the second part occurs at the terminal.

The Process at the EDCDLP for Transmitting Characters

Figure 7–4 illustrates the process that occurs at the EDCDLP when data is transferred from the mainframe to a terminal. The process is repeated once for each character transmitted.

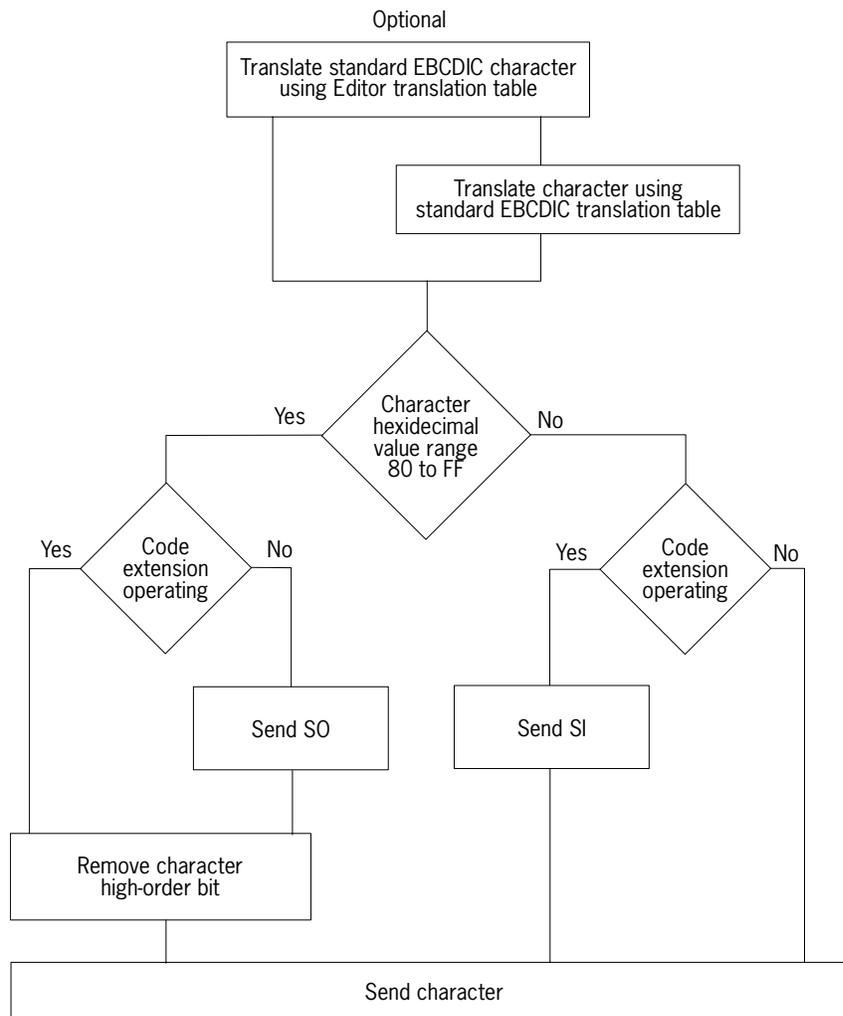


Figure 7–4. Transmitting Data: The Process at the EDCDLP

The Process at the Terminal for Receiving Data

The process illustrated in Figure 7-5 occurs at the terminal once for each character received.

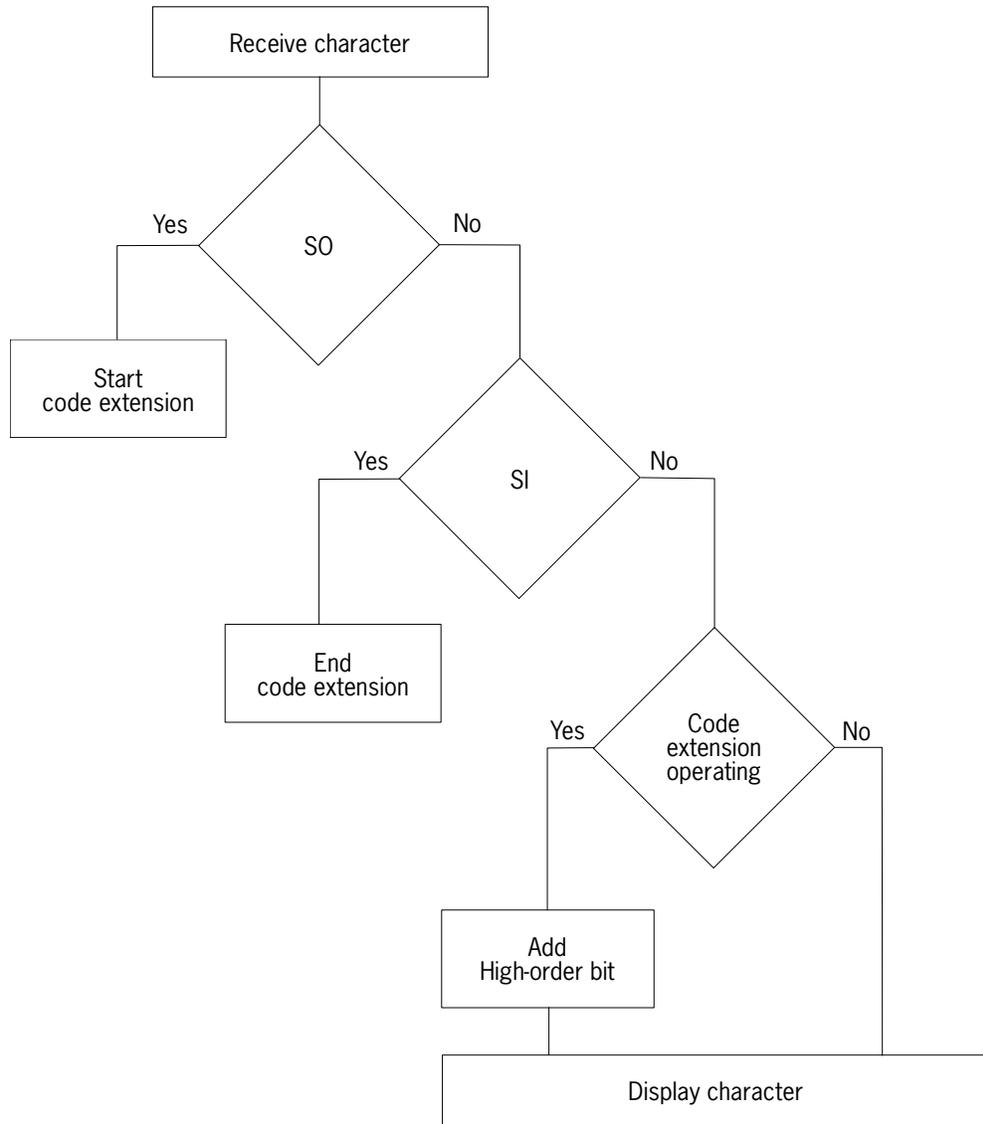


Figure 7-5. Receiving Data: The Process at the Terminal

Using Coded Character Set Mapping Tables

When data is transferred between devices and the mainframe, each character can be mapped using one or more 8-bit coded character set mapping tables. The coded character set mapping tables are defined in the file *SYMBOL/SOURCENDLII.

The number of coded character set mapping tables you require depends on the number of coded character sets in use on a single datacom line. If all the devices on a single datacom line and the mainframe use the same coded character set, you need not use any coded character set mapping tables. If all the devices on a single datacom line use the same coded character set, you need use only one coded character set mapping table. However, if more than one coded character set is in use on a datacom line, you must use at least two coded character set mapping tables.

Note: *Using code extension, you can use characters in the hexadecimal range 80 through FF. However, to avoid possible confusion with control characters, you should limit the characters you use with code extension to the hexadecimal range A0 through FE.*

Using Code Extension with One Coded Character Set Mapping Table

Only one coded character set mapping table is required for code extension if all the devices on a single datacom line use the same coded character set. This coded character set mapping table is used in the Adapter Control of the NDII algorithm.

In the Protocols 05 release, the standard EBCDIC mapping table in the *SYMBOL/SOURCENDLII file provided for code extension is called ASCIIIX. The ASCIIIX mapping table does the following:

- Maps ASCII characters in the hexadecimal range 00 through 7F to their standard EBCDIC counterparts
- Maps ASCII characters in the hexadecimal range A0 through FF to normally unused positions in the standard EBCDIC table
- The ASCII hexadecimal value A0 is mapped to the first empty space in the standard EBCDIC table. The ASCII hexadecimal value A1 is mapped to the second empty space in the standard EBCDIC table, and so on.

Figure 7–6 illustrates the ASCIIIX mapping table.

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE			SP	&	-		C2	C9	D0	D8	{	}	\	0
1	SOH	DC1			A0	A9	/	BA	a	j	D1	D9	A	J		1
2	STX	DC2		SYN	A1	AA	B2	BB	b	k	s	DA	B	K	S	2
3	ETX	DC3			A2	AB	B3	BC	c	l	t	DB	C	L	T	3
4					A3	AC	B4	BD	d	m	u	DC	D	M	U	4
5	HT		LF		A4	AD	B5	BE	e	n	v	DD	E	N	V	5
6		BS	ETB		A5	AE	B6	BF	f	o	w	DE	F	O	W	6
7	DEL		ESC	EOT	A6	AF	B7	C0	g	p	x	DF	G	P	X	7
8		CAN			A7	B0	B8	C1	h	q	y	E0	H	Q	Y	8
9		EM			A8	B1	B9	`	i	r	z	E1	I	R	Z	9
A					[]		:	C3	CA	D2	E2	E8	EE	F4	FA
B	VT				.	\$,	#	C4	CB	D3	E3	E9	EF	F5	FB
C	FF	FS		DC4	<	*	%	@	C5	CC	D4	E4	EA	F0	F6	FC
D	CR	GS	ENQ	NAK	()	_	'	C6	CD	D5	E5	EB	F1	F7	FD
E	SO	RS	ACK		+	;	>	=	C7	CE	D6	E6	EC	F2	F8	FE
F	SI	US	BEL	SUB	!	^	?	"	C8	CF	D7	E7	ED	F3	F9	FF

Figure 7–6. ASCIIIX Mapping Table

The mapping provided by the ASCIIIX mapping table might not be compatible with the coded character set operating on your mainframe. To change the mapping process, you must either modify the ASCIIIX table or add a new coded character set mapping table.

If you create a new coded character set mapping table, use the ASCIIIX mapping table as a model. You must then associate the new coded character set mapping table with the required datacom lines by changing the associated class attribute code in the Configuration module of the *SYMBOL/SOURCENDLII file. You can also edit the class attribute code by using the Interactive Datacomm Configurator (IDC).

Using Code Extension with Multiple Coded Character Set Mapping Tables

If the printers and terminals on a single datacom line do not all use the same coded character set, the code extension process must use at least two coded character set mapping tables. One coded character set mapping table is used in all cases in the Adapter Control. The second coded character set mapping table is defined on a device-by-device basis and accessed in the Editor for each station on the line.

Figure 7-7 illustrates the mapping process if two terminals on the same datacom line use different coded character sets.

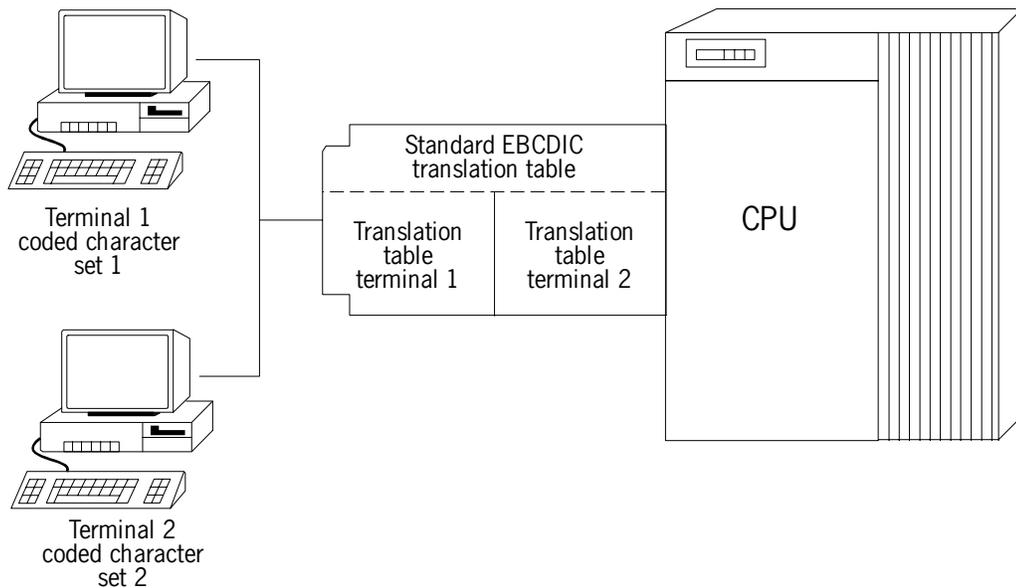


Figure 7-7. Mapping Process for Multiple Coded Character Sets

You must define the coded character set mapping tables for each terminal or printer that uses code extension. The standard EBCDIC mapping table is the ASCII table. The coded character set mapping tables in the Editor must take into account the functionality of the ASCII mapping table, the coded character set in use on the system, and the coded character set in use on the device.

To create coded character set mapping tables and define the printers and terminals that must use a particular coded character set mapping table, use the following procedure:

1. Access the file *SYMBOL/SOURCENDLII.
2. Locate the mapping tables in the *SYMBOL/SOURCENDLII file by doing a search for the text string *LOWERTOUPPERCASE*.
3. Add the coded character set mapping tables you require.
4. Add any required new enumerations to the enumeration type CHAR_SET_TYPE before the translate tables to equate a unique number with each coded character set mapping table. The enumeration declaration has the following syntax:

```
— ENUMERATION — CHAR_SET_TYPE — = — ( —<declaration>— ); —————|
```

<declaration>

```
— |<ccs>— : —<integer>—|—————|
```

The <ccs> element defines the coded character set used by the station. The <integer> element is the number you want to assign to the coded character set.

5. Add a global station include declaration of the same type as the enumeration declaration. Global station include declarations are located after the enumeration declaration but before the first coded character set mapping table. Following is a global station include declaration:

```
CHAR_SET_TYPE CHAR_SET:= STANDARD
```

6. Use the following case statement syntax to modify the input and output editors. The case statement enables the editors to recognize the coded character set mapping tables required by each station.

```
— CASE — STATION.CHAR_SET — BEGIN —<process>— END; —————|
```

<process>

```
— |<ccs>— : — SKIP —————|
      | TRANSLATE(SOURCE, —<t-table>— ); —|
```

The following table explains the elements of the case statement:

Element	Explanation
<ccs>	Names the coded character set in use on the terminal or printer.
SKIP	Requests no mapping.
<t-table>	Identifies the coded character set mapping table associated with the coded character set.

Configuring Unisys Products for Internationalization

7. Assign coded character set mapping tables to stations either by altering the station definition in the Configuration module or by using IDC. For more information on IDC, refer to the *Interactive Datacomm Configurator (IDC) Operations Guide*.
8. End your edit session and save the *SYMBOL/SOURCENDLII file.
9. Compile the *SYMBOL/SOURCENDLII file by using the following statement in CANDE:

```
C *SYMBOL/SOURCENDLII as $SYSTEM/NIF
```

10. Post-compile the *SYSTEM/NIF file by using the following procedure in CANDE:
 - a. Use the following statement to create a file and to start a Utility Editor session:

```
MAKE <file name>;U ED
```

For example, to create and edit the file NPC/CARD, use the following statement:

```
MAKE NPC/CARD;U ED
```

- b. Add the following two compiler control records to the file:

```
$ RESET LIST CODE
```

```
$ RESET ERRORLIST
```

- c. End your edit session and save the file.
 - d. Use the following statements to post-compile the new *SYSTEM/NIF file to generate the \$NEW/NIF file:

```
C NPC/CARD AS $NEW/NIF WITH NPC;  
C FILE HOST = SYSTEM/NIF;  
C FILE SOURCE = SYMBOL/SOURCENDLII
```

The elements of the statement are explained in the following table:

Element	Explanation
NPC/CARD	The file you created containing the compiler control cards
\$NEW/NIF	A temporary name for the new Network Information File (NIF) produced by post-compiling the *SYSTEM/NIF file in step 10
SYSTEM/NIF	The file you created by compiling the *SYMBOL/SOURCENDLII file in step 9
SYMBOL/SOURCENDLII	The file compiled in step 9

For other methods of post-compiling an NDLL file, refer to the *Network Definition Language II (NDLII) Programming Reference Manual*.

11. Use the following system command to replace the existing NIF with the new NIF NEW/NIF:

```
"ID NEW";
"ID:QT"
```

Note that the file name you give in the ID command does not include the /NIF node.

For more information on the ID command, refer to the *System Commands Reference Manual*.

Examples

Example 1

Figure 7–8 illustrates the code required to assign numbers to the coded character sets for ASERIESEBCDIC, Latin1EBCDIC, and Latin2EBCDIC.

```
ENUMERATION CHAR_SET_TYPE =           % Coded character set.
(STANDARD : 0,                         % ASERIESEBCDIC
 LATIN1   : 1,                         % Latin1EBCDIC
 LATIN2   : 2,                         % Latin2EBCDIC
);
```

Figure 7–8. Declaring Enumerations

Example 2

Figure 7–9 illustrates the code defining the mapping tables ASCIIX, LATIN1TOLATIN1, and LATIN1TOLATIN2. The ASCIIX translate table maps ASCII to standard EBCDIC. The LATIN1TOLATIN1 translate table is used for a host using Latin1EBCDIC and a device using ASCII Latin1. The LATIN1TOLATIN2 mapping table is used for a host using Latin1EBCDIC and a device using ASCII Latin2.

```
TRANSLATETABLE ASCIIX (7) % EXTENDED EBCDIC TO ASCII 7
OUTPUT (ASCII, 4"415877A0B8DC" TO 4"A0B0C0D0E0F0",
        4"425978A1B9DD" TO 4"A1B1C1D1E1F1",
        4"436280AABADE" TO 4"A2B2C2D2E2F2",
        4"44638AABBBDF" TO 4"A3B3C3D3E3F3",
        4"45648BACBCEA" TO 4"A4B4C4D4E4F4",
        4"46658CADBDEB" TO 4"A5B5C5D5E5F5",
        4"47668DAEBEEC" TO 4"A6B6C6D6E6F6",
        4"48678EAFBFED" TO 4"A7B7C7D7E7F7",
        4"49688FBOCAEE" TO 4"A8B8C8D8E8F8",
        4"516990B1CBEF" TO 4"A9B9C9D9E9F9",
        4"52719AB2CCFA" TO 4"AABACADAEAF",
        4"53729BB3CDFB" TO 4"ABBECDBEBFB",
        4"54739CB4CEFC" TO 4"ACBCCDCEFC",
        4"55749DB5CFFD" TO 4"ADBECDDDFD",
        4"56759EB6DAFE" TO 4"AEBECEDEEEFE",
```

Configuring Unisys Products for Internationalization

```
4"57769FC7DBFF" TO 4"AFBFCDFEFFF"),
    % EXTENDED ASCII (80-9F EXCLUDED) TO
    % EBCDIC

INPUT (ASCII, 4"A0B0C0D0E0F0" TO 4"415877A0B8DC",
    4"A1B1C1D1E1F1" TO 4"425978A1B9DD",
    4"A2B2C2D2E2F2" TO 4"436280AABADE",
    4"A3B3C3D3E3F3" TO 4"44638AABBBDF",
    4"A4B4C4D4E4F4" TO 4"45648BACBCEA",
    4"A5B5C5D5E5F5" TO 4"46658CADBDEB",
    4"A6B6C6D6E6F6" TO 4"47668DAEBEEC",
    4"A7B7C7D7E7F7" TO 4"48678EAFBFED",
    4"A8B8C8D8E8F8" TO 4"49688FB0CAEE",
    4"A9B9C9D9E9F9" TO 4"516990B1CBEF",
    4"AABACADAEAF" TO 4"52719AB2CCFA",
    4"ABBECBDBEBFB" TO 4"53729BB3CDFB",
    4"ACBCCDCEFC" TO 4"54739CB4CEFC",
    4"ADBECDDDED" TO 4"55749DB5CFFD",
    4"AEBECEDEEEFE" TO 4"56759EB6DAFE",
    4"AFBFCDFEFFF" TO 4"57769FB7DBFF");

TRANSLATETABLE LATIN1TOLATIN1(8) % Host LATIN1EBCDIC to
    % STANDARD EBCDIC to be outbound
    % for ASCII LATIN1 device.

OUTPUT (EBCDIC, 4"419064AC448C" TO 4"415877A0B8DC",
    4"AA8F65694549" TO 4"425978A1B9DD",
    4"B0EA62ED42CD" TO 4"436280AABADE",
    4"B1FA66EE46CE" TO 4"44638AABBBDF",
    4"9FBE63EB43CB" TO 4"45648BACBCEA",
    4"B2A067EF47CF" TO 4"46658CADBDEB",
    4"BBB69EEC9CCC" TO 4"47668DAEBEEC",
    4"B5B368BF48E1" TO 4"48678EAFBFED",
    4"BD9D74805470" TO 4"49688FB0CAEE",
    4"B4DA71FD51DD" TO 4"516990B1CBEF",
    4"9A9B72FE52DE" TO 4"52719AB2CCFA",
    4"8A8B73FB53DB" TO 4"53729BB3CDFB",
    4"BAB778FC5BDC" TO 4"54739CB4CEFC",
    4"CAB875AD558D" TO 4"55749DB5CFFD",
    4"AFB976AE568E" TO 4"56759EB6DAFE",
    4"BC8B775957DF" TO 4"57769FB7DBFF"),
    % Inbound from ASCII LATIN1 device
    % STANDARD EBCDIC to
    % host Latin1EBCDIC.

INPUT (EBCDIC, 4"415877A0B8DC" TO 4"419064AC448C",
    4"425978A1B9DD" TO 4"AA8F65694549",
    4"436280AABADE" TO 4"B0EA62ED42CD",
    4"44638AABBBDF" TO 4"B1FA66EE46CE",
    4"45648BACBCEA" TO 4"9FBE63EB43CB",
    4"46658CADBDEB" TO 4"B2A067EF47CF",
    4"47668DAEBEEC" TO 4"BBB69EEC9CCC",
    4"48678EAFBFED" TO 4"B5B368BF48E1",
    4"49688FB0CAEE" TO 4"BD9D74805470",
```

```

4"516990B1CBEF" TO 4"B4DA71FD51DD",
4"52719AB2CCFA" TO 4"9A9B72FE52DE",
4"53729BB3CDFB" TO 4"8A8B73FB53DB",
4"54739CB4CEFC" TO 4"BAB778FC5BDC",
4"55749DB5CFFD" TO 4"CAB875AD558D",
4"56759EB6DAFE" TO 4"AFB976AE568E",
4"57769FB7DBFF" TO 4"BC8B775957DF");

TRANSLATETABLE LATIN1TOLATIN2(8)  % Host Latin1EBCDIC to
                                % STANDARD EBCDIC to be outbound
                                % for ASCII LATIN2 device.
    OUTPUT (EBCDIC, 4"419064AC448C" TO 4"415800A000DC",
            4"AA8F65694549" TO 4"00007800B900",
            4"B0EA62ED42CD" TO 4"00008000BA00",
            4"B1FA66EE46CE" TO 4"000000AB00DF",
            4"9FBE63EB43CB" TO 4"45648BACBCEA",
            4"B2A067EF47CF" TO 4"000000000000",
            4"BBB69EEC9CCC" TO 4"000000AE00EC",
            4"B5B368BF48E1" TO 4"48678EAFBFED",
            4"BD9D74805470" TO 4"490000000000",
            4"B4DA71FD51DD" TO 4"00009000CB00",
            4"9A9B72FE52DE" TO 4"000000B200FA",
            4"8A8B73FB53DB" TO 4"00009B00CD00",
            4"BAB778FC5BDC" TO 4"000000B400FC",
            4"CAB875AD558D" TO 4"55009DB5CFFD",
            4"AFB976AE568E" TO 4"00000000DA00",
            4"BC8B775957DF" TO 4"000000B700FF"),
            % Inbound from ASCII LATIN2 device
            % STANDARD EBCDIC to
            % host Latin1EBCDIC.
    INPUT  (EBCDIC, 4"415877A0B8DC" TO 4"419000AC008C",
            4"425978A1B9DD" TO 4"000065004500",
            4"436280AABADE" TO 4"000062004200",
            4"44638AABBBDF" TO 4"000000EE00CE",
            4"45648BACBCEA" TO 4"9FBE63EB43CB",
            4"46658CADBDEB" TO 4"000000000000",
            4"47668DAEBEEC" TO 4"000000EC00CC",
            4"48678EAFBFED" TO 4"B50068BF48E1",
            4"49688FB0CAEE" TO 4"BD0000000000",
            4"516990B1CBEF" TO 4"000071005100",
            4"52719AB2CCFA" TO 4"000000FE00DE",
            4"53729BB3CDFB" TO 4"000073005300",
            4"54739CB4CEFC" TO 4"000000FC00DC",
            4"55749DB5CFFD" TO 4"CA0075AD558D",
            4"56759EB6DAFE" TO 4"000000005600",
            4"57769FB7DBFF" TO 4"0000005900B3");

```

Figure 7–9. Creating Coded Character Set Mapping Tables

Example 3

Figure 7–10 illustrates the code you must add to the input and output editors to ensure that ASCII, LATIN1TOLATIN1, and LATIN1TOLATIN2 are used as mapping tables.

```
CASE STATION.CHAR_SET
BEGIN
  STANDARD:
    SKIP;
  LATIN1:
    TRANSLATE(SOURCE, LATIN1TOLATIN1);
  LATIN2:
    TRANSLATE(SOURCE, LATIN1TOLATIN2);
END;  % CASE STATION.CHAR_SET
```

Figure 7–10. Recognizing Coded Character Set Mapping Tables

Requesting Code Extension

To request code extension for a terminal, a printer, or other device, you must complete certain tasks depending on the algorithm you are using. The following list describes the tasks you must complete when requesting code extension for poll/select, enhanced poll/select, or any other algorithm:

- For poll/select or enhanced poll/select, you must alter the CODE_EXT_MODE station variable in the *SYMBOL/SOURCENDLII file or use IDC to alter the CODE_EXT_MODE variable for each station. CODE_EXT_MODE is a new local STATION INCLUDE variable in poll/select and in enhanced poll/select in the Protocols 05 release.

To alter the CODE_EXT_MODE station variable in the *SYMBOL/SOURCENDLII file, use the NDLL <station initialize attribute> syntax in the station definition portion of the Configuration module. For more information on NDLL and the <station initialize attribute> syntax, refer to the *NDLL Reference Manual*.

- For other algorithms, you must declare the CODE_EXT_MODE station variable by using NDLL to add the appropriate declaration to the *SYMBOL/SOURCENDLII file.

Use the <include declaration> syntax with the <station part> syntax in NDLL to declare the CODE_EXT_MODE station variable for an algorithm. For more information on NDLL and the <include declaration> syntax, refer to the *NDLL Reference Manual*.

The following paragraphs explain the way to set code extension for any algorithm.

Setting Code Extension

Once you have requested code extension for an algorithm, you can set the CODE_EXT_MODE station variable to one of the following three settings:

Setting	Explanation
EXT_NONE	Indicates the station does not use code extension. This is the default value.
EXT_SO	Indicates code extension using the control characters SO and SI to mark the beginning and end of data transfers that require 8 bits to define each character.
EXT_ESC_SO	Indicates code extension using the control characters ESC SO and ESC SI to mark the beginning and end of data transfers that require 8 bits to define each character.

Refer to the *Data Communications Protocols Guide* for more information concerning these three settings of the CODE_EXT_MODE station variable.

Transporting Kanji and Other Double-Octet Characters

If you are transporting Kanji characters or other double-octet characters, you must use the start-of-double-octet (SDO) and end-of-double-octet (EDO) characters to enclose strings of double-octet characters. If you want to expand your character set, you can use code extension with the SDO and EDO characters.

The SDO and EDO characters are synonyms for the start-of-Kanji (SOK) and end-of-Kanji (EOK) characters, not replacements. (The SDO and EDO characters have the same code values that the SOK and EOK characters have.)

Configuring Client Access Services for Internationalization

Client Access Services handles data translations when file names, message text, and certain file data are transferred between the MCP environment and internationalized versions of Windows.

The administrator must ensure that the MCP environment ccsversion, datacom subsystem translation tables, and workstation coded character sets are properly configured so that correct character translation occurs between the MCP environment and the workstations.

Which Coded Character Sets Are Used?

The coded character sets that are used for translation are determined by the following conditions:

- If SW7 is TRUE when *SYSTEM/NXSERVICES/SERVER starts, then the workstations are assumed to be using the Shift-JIS ASCII coded character set and the MCP host is assumed to be using the JapanEBCDIC coded character set. You can modify the *SYSTEM/NXSERVICES/SERVER code file with WFL to set SW7 to TRUE by default.
- If the SYSOPS ccsversion is ASERIESNATIVE, then the workstations are assumed to be using CODEPAGE850 for DOS data and CODEPAGE1252 for Windows data. The MCP host is assumed to be using EBCDIC and employs the standard ASCII TO EBCDIC and EBCDIC TO ASCII translation tables in the datacom system. The terminal emulators are assumed to be using Latin1ISO.
- If the SYSOPS ccsversion is other than ASERIESNATIVE, refer to the following table to verify which coded character sets are in use.

Table 7-4. Ccsversions and Their Character Sets

Ccsversion	DOS Coded Character Set	Windows Coded Character Set	MCP Coded Character Set
BRAZILIAN	850	1252	Latin1EBCDIC
CANADAEBCDIC	850	1252	Latin1EBCDIC
CZECHOSLOVAKIA	852	1250	Latin2EBCDIC
FRANCE	850	1252	Latin1EBCDIC
HUNGARIAN	852	1250	Latin2EBCDIC
LATINGREEK	851	1253	LatinGREEKEBCDIC
NORWAY	850	1252	Latin1EBCDIC
POLISH	852	1250	Latin2EBCDIC
ROMANIAN	852	1250	Latin2EBCDIC
RUSSIAN	866	1251	LatinCYRILLICEBCC
SPANISH	850	1252	Latin1EBCDIC
SWEDISH1	850	1252	Latin1EBCDIC
SWISS	850	1252	Latin1EBCDIC
TURKISH	857	1254	Latin5EBCDIC

Note: In the MCP environment, file titles are stored and message text is displayed using the MCP coded character set. It is assumed that the datacom subsystem translation tables and the terminal emulator coded character set are configured correctly to display the MCP coded character set.

Which Files Are Translated?

The following kinds of files and data are translated:

- File titles, message text, share names, and share comments are always translated.
- Files created by workstations are stored with MAXRECSIZE=1, FRAMESIZE=8, and EXTMODE=ASCII. No translation is done when workstation files are written or read.
- Files that do not have one character record are treated as MCP files. FILEKIND is used to locate the text portion of the record. If FILEKIND indicates a non-text file or a code file, then the open is not allowed.

How Are Translations Performed?

Translations are performed under the following conditions:

- If EXTMODE is EBCDIC or SINGLE and SW7 is TRUE, then data is translated from JapanEBCDIC to Shift-JIS.
- If EXTMODE is EBCDIC or SINGLE, SW7 is TRUE, and the SYSOPS ccsversion is ASERIESNATIVE, then data is translated with EBCDICTOASCII. This process reverses the translation that occurred when the data entered the system via the standard datacom translation tables.
- If EXTMODE is set to match the MCP coded character set and SW7 is FALSE, then data is translated from the MCP coded character set to the Windows coded character set. Note that, in this case, Latin1EBCDIC and Latin5EBCDIC are assumed to match.

When EXTMODE is any variant of ASCII or OCTETSTRING, the data is returned to the workstation without translation.

Refer to the *Cleint Access Services Administration Guide* for further details.

Section 8

Using C Programs in an MLS Environment

Two ways exist for a C program to access and use the internationalization features that an MLS environment provides:

- C programs can access and use the CENTRALSUPPORT library procedures. For more information, refer to the *C Programming Reference Manual, Volume 2: Headers and Functions*.
- C programs can use locales associated with the <locale.h> standard header.

This section describes locales and the way they can affect C programs in an MLS environment, provides the steps to create or modify a locale, and explains the effects of locale components on the library functions in a C program.

Understanding Locales

A central element of the internationalization features provided by the C programming language is the concept of a *locale*. The implementation of C locales defines a locale with four *locale components* that designate the ccsversion, language, convention, and spaces set that are used within a C program. The CENTRALSUPPORT system library is accessed by the C program to obtain information about these four locale components. Refer to “Understanding the LOCALES Array” later in this section for more information about the locale components. You can also refer to the discussion of the <locale.h> standard header in the *C Reference Manual* for more information on locales and locale-sensitive library functions.

The C library function, `setlocale`, is used in a C program to select the locale that is to be used for the C program. The `setlocale` function also designates the aspects of the C program, known as *locale categories*, that are affected by the locale that is chosen. Refer to “Understanding Locale Categories” later in this section for more information.

Before C programs can take full advantage of the internationalization functionality provided by the C programming language, either the subsidiary or the local site administrator must create additional locales or modify one of the predefined locales supplied with the C compiler. For instance, the predefined NATIVE locale should be customized to reflect the standards and functional requirements that are desired for the native environment of the local system.

Refer to “Understanding the Predefined Locales” later in this section for more information on the predefined locales. Refer to “Creating or Modifying Locales” later in this section for the instructions on creating and modifying locales.

Understanding Locale Categories

A locale category delimits the portion of a C program that is affected by a particular locale. Locale categories are divided into five program aspects as follows:

- Collating of strings
- Character handling
- Date and time formatting
- Monetary formatting
- Numeric formatting

A C program can dynamically select the desired locale for one of the five locale categories by using the `setlocale` function declared in the `<locale.h>` standard header. The C program can change locales during processing by calling the `setlocale` function and providing different names for the locale. This capability is useful if, for example, you want to print a report that uses different monetary formatting conventions. The syntax needed to include the `<locale.h>` header file is shown by the following statement:

```
#include <locale.h>
```

This syntax provides the following declaration of the `setlocale` function to the C program:

```
char *setlocale (int category, const char * locale);
```

The *category* argument designates the locale category that is affected by the locale designated in the *locale* argument. Table 8–1 lists the seven values for the category argument, which are defined as C macros in the <locale.h> standard header, and describes the portion of the C program that is affected by each category:

Table 8–1. Locale Categories

Category	Portion of Category Affected
LC_ALL	All five locale categories
LC_CHECK_LOCALEDEF	All five locale categories Use the LC_CHECK_LOCALEDEF category solely for debugging purposes when you are creating or modifying locales. If a problem exists with the locale definition and you use the LC_CHECK_LOCALEDEF category, explanatory messages are displayed and the C program is terminated. For more information, refer to “Testing Locales” later in this section.
LC_COLLATE	Collating of strings
LC_CTYPE	Character handling
LC_MONETARY	Monetary formatting
LC_NUMERIC	Numeric formatting
LC_TIME	Date and time formatting

The following example shows the way you can designate the locale category and the locale in a `setlocale` function. In this example, the JAPANESE locale is selected for the monetary formatting locale category in the C program.

```
setlocale (LC_MONETARY, "JAPANESE");
```

At program start-up, the C compiler designates the predefined C locale as the current locale for all the locale categories. This designation remains in effect until the `setlocale` function is used to change the locale setting. The predefined C locale is explained in Table 8–2.

Understanding the Predefined Locales

The C compiler contains two predefined locales: C and NATIVE. Table 8–2 describes the predefined C and NATIVE locales. For information on defining additional locales, refer to “Creating or Modifying Locales” later in this section.

Table 8–2. Predefined Locales

Locale Name	Explanation
C	<p>All C library functions that are locale-sensitive execute as defined in the American National Standards Institute (ANSI) draft for the C Programming Language standard, X3.159-1989. The locale-sensitive functions are described later in this section under “Effects of Locale Components on C Library Functions.”</p> <p>The C locale is the only locale that does not access and use the CENTRALSUPPORT system library. Therefore, the definitions of the locale components do not affect program results when the C locale is used.</p> <p>At program start-up, the C compiler designates the C locale as the current locale for all the locale categories. This designation remains in effect until the setlocale function is used to change the locale setting.</p>
NATIVE	<p>The NATIVE locale is referred to as the default native environment in the C programming language. A C program can select the NATIVE locale by passing a null string in the setlocale function as shown in the following statement:</p> <pre data-bbox="532 1146 808 1171">setlocale (LC_ALL, "");</pre> <p>Since the NATIVE locale is the default native environment for a system, the local site administrator should customize the four locale components of this locale for the particular standards and functional requirements of the site. When a C program uses the customized NATIVE locale, the C library functions that are locale-sensitive perform according to the characteristics that are native to the local site.</p> <p>As supplied on the release media, the four locale components of the NATIVE locale are defined as follows:</p> <ul data-bbox="500 1465 1247 1633" style="list-style-type: none"> • The ccsversion component is defined to be ASERIESNATIVE. • The language component is defined to be ENGLISH. • The convention component is defined to be ASERIESNATIVE. • The spaces set component is defined to be the decimal numbers "64,12,37,13,5,11". <p>The string of numbers defined for the spaces set component designates the following characters as white space characters: space, form feed, line feed, carriage return, horizontal tab, and vertical tab.</p>

Creating or Modifying Locales

You can create a locale or modify a locale for use with any C program. The C locale definitions are part of the SLICESUPPORT library in the *SYSTEM/SLICESUPPORT file. The following list describes how to create or modify a locale:

- To create a locale, you must add it to the SLICESUPPORT library. Use the MSGTRANS utility to add the name of the new locale to the LOCALES array in the *SYSTEM/SLICESUPPORT file and then defining the following four locale components: ccsversion, language, convention, and spaces set.
- To modify a locale, you must change one or more of the locale components already defined in the *SYSTEM/SLICESUPPORT file.

The following paragraphs explain more about the way to create and modify locales.

Understanding the LOCALES Array

The *SYSTEM/SLICESUPPORT file contains the LOCALES array, which defines each locale and its four locale components. You can use MSGTRANS to create or modify locales in the LOCALES array, which is defined as an output message array. However, unlike most output message arrays, the LOCALES array does not contain user-oriented messages to be translated to various native languages. Rather, each defined language corresponds to a particular locale and each message corresponds to one of the four components of that locale. Once a language (locale) has been set up in the LOCALES array, the language name that is used for the locale can be used in a C program to refer to that locale.

For example, when you use MSGTRANS to create the four messages (locale components) for the Spanish language, you actually create a new locale, which is called *SPANISH*. A C program can invoke the SPANISH locale by executing the following statement:

```
setlocale (LC_ALL, "SPANISH");
```

The LOCALES array contains four messages that define the locale components. When you use MSGTRANS to create a new locale or to modify an existing locale, you are editing messages 0 through 3 of the LOCALES array. The contents of the messages in the LOCALES array are described as follows:

- If you are creating a locale, the messages in the LOCALES array are blank. You must define each of the four messages with the appropriate value. For example, you must enter the name of a convention as the text of message 2.
- If you are modifying a locale, the messages in the LOCALES array contain the names of the current locale components. For example, message 1 might contain the text *DEUTSCH*, which designates the language for the locale.

Note that when you are modifying a locale, you do not have to change all the locale components—you change only the components you want to change.

Note: The *ccsversion*, *language*, and *convention* that are defined for a locale must be valid for the site that is to use the locale. If a locale that is defined with invalid components is used in a *setlocale* function, the *setlocale* function returns a null pointer and the current locale does not change.

Table 8–3 lists each of the four locale components, the corresponding message number in the `LOCALES` array that defines the locale component, and an explanation of the way the locale component is used.

Table 8–3. Locale Components in the `LOCALES` Array

Locale Component	Message Number	Explanation
Ccsversion	0	Designates the <i>ccsversion</i> for the locale. For example, you can enter <code>SWISS</code> as the text of message 0 to designate that <code>SWISS</code> is the <i>ccsversion</i> for the locale.
Language	1	Designates the <i>language</i> for the locale. For example, you can enter <code>DEUTSCH</code> as the text of message 1 to designate that <code>Deutsch</code> is the <i>language</i> for the locale.
Convention	2	Designates the <i>convention</i> for the locale. For example, you can enter <code>SWITZERLAND</code> as the text of message 2 to designate that <code>Switzerland</code> is the <i>convention</i> for the locale.
Spaces set	3	Designates the white space characters for the locale. The <i>spaces set</i> locale component is a string of decimal numbers enclosed in quotation marks (<code>"</code>). Each decimal number represents a character that is considered a white space character.

Table 8-3. Locale Components in the LOCALES Array

Locale Component	Message Number	Explanation
Spaces set (Cont.)	3	<p>For example, if the ccsversion of the locale is ASERIESNATIVE, you can enter "64,13" as the text of message 3 to designate the space and carriage return as white space characters. To obtain the decimal number that corresponds to a character you want to use in the spaces set string, complete the following steps:</p> <ol style="list-style-type: none"> 1. Determine the coded character set that is being used for the ccsversion of the locale. Refer to Table 12-1, "Ccsversion in the Current Release," for this information. 2. Find the table for this coded character set in Sections 9, 10, or 11. 3. Use the table column value and table row value for the character to obtain the corresponding decimal number. The following equation represents the decimal number of any character in the coded character set tables: $(\text{column number} * 16) + (\text{row number})$ <p>For example, assume that the character you want is displayed in column 08 and row 13 of the coded character set table. The decimal value equation is the following: $(8 * 16) + 13$ </p> <p>This equation yields the value 141, which is the decimal number that corresponds to the character in column 08 and row 13 of the coded character set table.</p>

Using MSGTRANS to Create or Modify a Locale

The following steps briefly explain the way to use MSGTRANS to add or modify a locale in the LOCALES array of the *SYSTEM/SLICESUPPORT file. Read the *MSGTRANS Operations Guide* for detailed information on using MSGTRANS.

1. Type the following statement in any CANDE session or in the Action field of any MARC screen and transmit:

```
RUN $SYSTEM/MSGTRANS ON <pack>
```

Use the syntax *ON <pack>* to identify the pack on which the MSGTRANS program resides. If MSGTRANS is on your default pack, you can omit this syntax from the statement.

MSGTRANS displays the Start Session screen illustrated in Figure 8–1.

```
What object file is to be edited? [ ]
To exit the program, enter ABORT here. [ ]

-----

HELP      For help at any time, press the SPCFY key. Information will be
          displayed about the field or caption where the cursor is
          located.

REFRESH   To refresh the screen at any time, type the following sequence:
          CTRL 0 0 XMT

-----

Unisys Message Translation Utility, Version 39.022.018
```

Figure 8–1. MSGTRANS Start Session Screen

2. Type **SYSTEM/SLICESUPPORT* in the What Object File Is To Be Edited? field and transmit.

MSGTRANS displays the Home screen illustrated in Figure 8–2.

3. Type *EDIT* in the Enter Command field and move to the next field.
4. Type the name of the locale, which is equivalent to the language you want, in the Enter the Language Name . . . field. The locale name you enter identifies the locale that you want to create or modify. For example, to create a French locale, enter *FRANCAIS* in this field. If you want to edit the *NATIVE* locale, type *NATIVE* in this field. Transmit.

MSGTRANS displays the Edit screen. If, for example, you have specified Francais as the language you want to edit, the first physical message in the file *SYSTEM/SLICESUPPORT is shown, as illustrated in Figure 8–3.

```
The messages are being loaded from the object file
-----
EDIT      Displays messages so you can view, edit, copy, or delete them.
INCLUDE   Compares messages, or adds messages from an external file.
LANGUAGES Displays languages defined for each message array.
PRINT     Sends messages to a line printer or a disk file.
RECESS    Ends the session and saves all messages to a recovery file.
SAVE      Saves messages and, optionally, starts a binding job.
BYE/END   Displays the Bye screen where you can end the session.

Enter Command: [      ]
-----
Enter the language name for the messages you want to edit: [      ]
To add or change the ccsversion, enter the ccsversion:    [      ]
To display comparison messages, enter the language name:  [      ]
```

Figure 8–2. MSGTRANS Home Screen

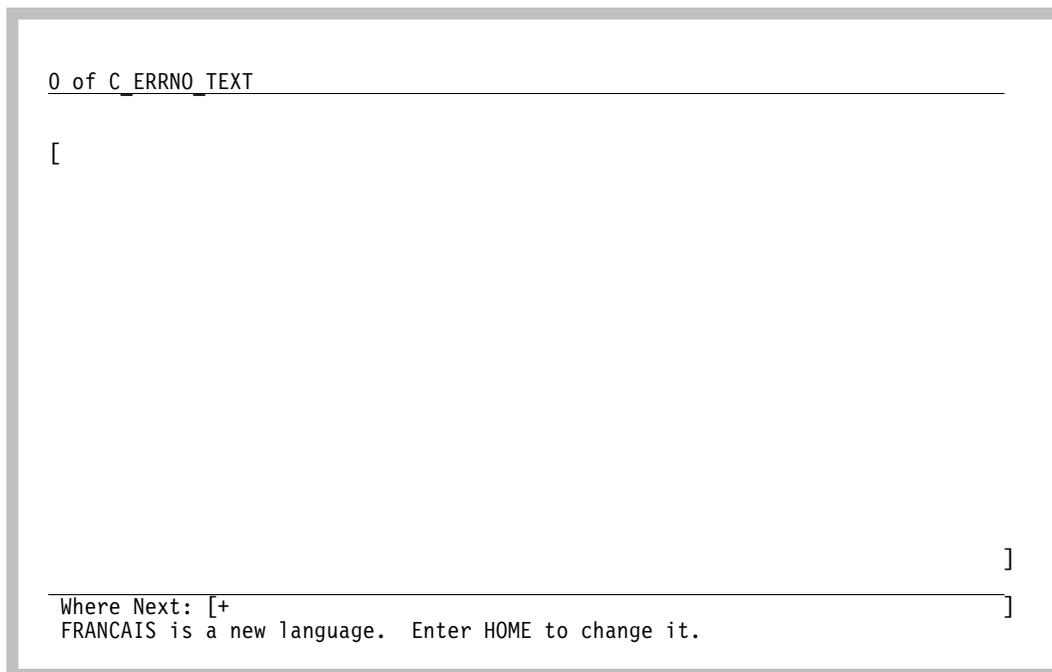


Figure 8-3. MSGTRANS Edit Screen—Displaying the First Physical Message

5. Type *FIRST OF LOCALES* in the Where Next field and transmit.

MSGTRANS refreshes the Edit screen and displays message 0 of the LOCALES array as illustrated in Figure 8-4. If you are creating a new locale, the Edit screen does not show any message text for the messages because the text has not been defined yet. If you are modifying a new locale, the Edit screen shows the message text for the messages.

Figure 8-4 shows the way the Edit screen looks when you are creating a new locale.

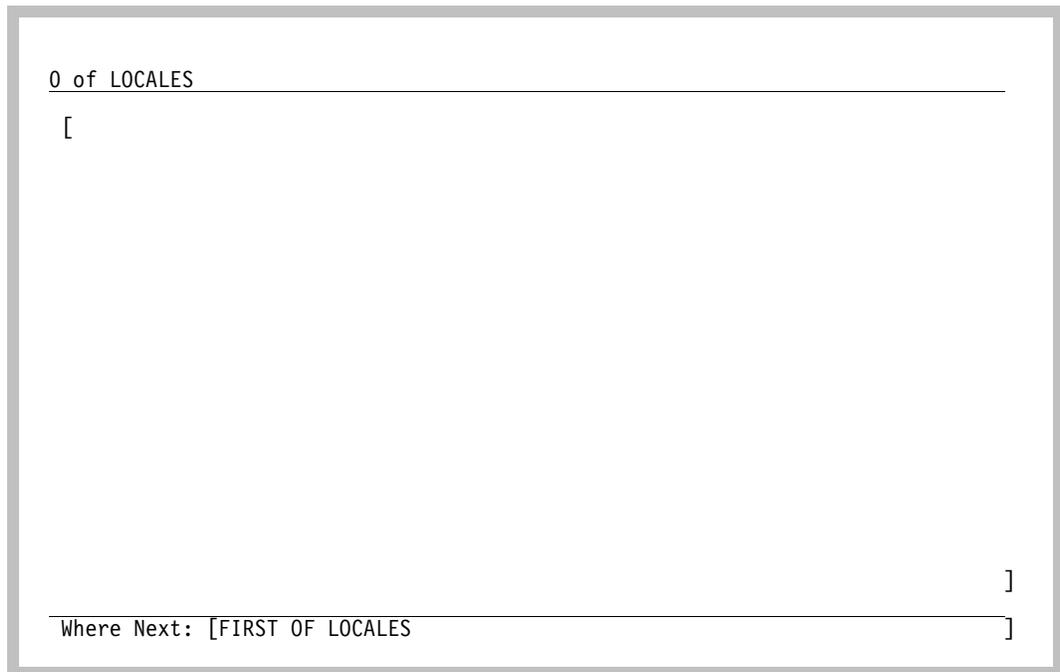


Figure 8–4. MSGTRANS Edit Screen—Displaying Message 0 of the LOCALES Array

6. Type the name of the ccsversion for the locale in the edit workspace. The ccsversion name must be bounded by quotation marks ("). For example, enter *"FRANCE"* to use the France ccsversion for the locale. Move to the next field.
7. Type *+* in the Where Next field. Transmit.
MSGTRANS displays the next message in the LOCALES array, which is message 1.
8. Type the name of the language for the locale in the edit workspace. The language name must be bounded by quotation marks ("). For example, enter *"FRANCAIS"* to use the Francais language for the locale. Move to the next field.
9. Type *+* in the Where Next field. Transmit.
MSGTRANS displays the next message in the LOCALES array, which is message 2.
10. Type the name of the convention for the locale in the edit workspace. The convention name must be bounded by quotation marks ("). For example, enter *"FranceListing"* to use the FranceListing convention for the locale. Move to the next field.

11. Type `+` in the Where Next field. Transmit.

MSGTRANS displays the last message in the LOCALES array, which is message 3.

12. Type the decimal numbers that correspond to the characters that are to be considered as white space characters for the locale in the edit workspace. The decimal numbers must be bounded by quotation marks ("). For example, if the `ccsversion` is `ASERIESNATIVE`, enter "`64,12,37`" to designate the space, form feed, and line feed characters as white space characters for the locale.

Refer to the discussion of the spaces set locale component in Table 8–3 for a description of the way to obtain the decimal number that corresponds to a character you want to use.

Move to the next field.

13. If you want to add or modify another locale, enter `HOME` in the Where Next field and transmit. Repeat all steps in this procedure, beginning with step 3.

If you want to save your changes and exit MSGTRANS, enter `SAVE` in the Where Next field and transmit.

MSGTRANS displays the Save screen illustrated in Figure 8–5.

```
-----  
Select OBJECT or SOURCE: [OBJECT]  
Enter file title for saved messages: [ ]  
  
What languages are to be saved? Mark an X by each language that is wanted.  
[X] ENGLISH  
[X] C  
[X] NATIVE  
[X] FRANCAIS  
  
To bind the saved messages file with the original object file, enter a file  
title for the resulting new object file: [ ]  
  
-----  
To return to the Home screen without saving, enter ABORT here: [ ]
```

Figure 8–5. MSGTRANS Save Screen

14. Bypass the Select OBJECT or SOURCE field. MSGTRANS pre-fills this field with the word `OBJECT`.
15. Type a name in the Enter File Title . . . field. This title is the name you want to give to the saved messages file that contains the changes you made to the LOCALES array. For example, you might name a French message file `FRANCAISMSGs`.
Move to the next field.

16. Bypass the *What Languages Are To Be Saved?* field. By default, MSGTRANS saves the messages in all languages.
17. Type a file name, for example *NEW/SLICESUPPORT*, in the *To Bind the Saved Messages . . .* field and transmit.

MSGTRANS binds the saved messages file to the file you designate in the *To Bind the Saved Messages . . .* field. The resulting file is a combination of the **SYSTEM/SLICESUPPORT* file and the changes that you made to the *LOCALES* array. The file is saved under the usercode from which you are running MSGTRANS.

Refer to “Making the New or Changed Locales Available” later in this section for information on the way to make this file accessible to C programs.

MSGTRANS then displays the Restart screen illustrated in Figure 8–6.

```
The messages are being saved.
A job named MLS_BIND is being initiated to perform the bind.

-----

What now?

    END      Ends the current session.
    HOME     Displays the Home screen and continues the current session.
    RESTART  Starts a new session with a different object file.

Enter Command: [      ]

-----
```

Figure 8–6. MSGTRANS Restart Screen

18. In the *Enter Command* field, type *END* and transmit to exit MSGTRANS.

Testing Locales

The implementation of the locales available to a C program is dependent upon the definitions of the `ccsversion`, `language`, `convention`, and `spaces set` components of the locale as well as the proper installation of the corresponding elements in the `CENTRALSUPPORT` system library.

If the `setlocale` function is unable to change the current locale setting to the requested locale, the `setlocale` function returns a null pointer to the C program. No error messages are issued. To make it easier to test various locales and to determine the source of a problem in a locale setup, the `LC_CHECK_LOCALEDEF` category is available. When the `setlocale` function is called with the name of a locale to be checked and the category is `LC_CHECK_LOCALEDEF`, the `setlocale` function displays informative messages when it finds any problems in the locale definition. The C program also is terminated when any problems are found and the `LC_CHECK_LOCALEDEF` category is used.

The `LC_CHECK_LOCALEDEF` category detects problems such as the following:

- Use of a locale name not found in the `LOCALES` array
- Use of a `ccsversion`, `language`, or `convention` in the `LOCALES` array that is not recognized by the `CENTRALSUPPORT` system library
- Use of an incorrectly defined monetary template in a `convention`
- The return of an error code from the `CENTRALSUPPORT` system library

Some of these errors could represent problems in the installation of the `CENTRALSUPPORT` system library.

After you have defined a new locale, you should test the new locale. The following sample C program illustrates the way you could test the `SPANISH` locale:

```
#include <stdio.h>
#include <locale.h>
main () {
    if (setlocale(LC_CHECK_LOCALEDEF, "SPANISH") == NULL) {
        printf("The setlocale function returned NULL. \n");
    } else {
        printf("The setlocale function returned successfully. \n");
    }
}
```

If the message "The setlocale function returned NULL." is displayed or the program terminates with an error message, the `SPANISH` locale is not defined properly or the `CENTRALSUPPORT` system library is not installed correctly. Check the C program output for any error messages from the `setlocale` function that could help you identify the problem.

Making the New or Changed Locales Available

After you exit MSGTRANS, you must update the *SYSTEM/SLICESUPPORT file and make the system recognize the new *SYSTEM/SLICESUPPORT file so that C programs can access the new or changed locales. The following steps explain the tasks you must complete:

1. Rename the file whose name you designated in the To Bind the Saved Messages... field of the MSGTRANS Save screen to the name *SYSTEM/SLICESUPPORT. You can use the CHANGE command in CANDE to rename the file.

For example, if you titled the new bound messages file as NEW/SLICESUPPORT, you enter the following command in CANDE to change the name of the new file:

```
CHANGE NEW/SLICESUPPORT TO *SYSTEM/SLICESUPPORT
```

Note that you can rename the new file as *SYSTEM/SLICESUPPORT only if you are a privileged user. Depending on the way system security is defined, you might have to delete the old *SYSTEM/SLICESUPPORT file before you can rename the new file.

2. Establish the new *SYSTEM/SLICESUPPORT file for the SLICESUPPORT library.

Use the SL (System Library) system command to establish the file for the SLICESUPPORT library. To do so, enter the following command on a supervisory printer output (SPO) console:

```
SL SLICESUPPORT=*SYSTEM/SLICESUPPORT
```

The SL (System Library) command instructs the system to use the latest version of the *SYSTEM/SLICESUPPORT file for requests to the SLICESUPPORT library. For more information on the SL (System Library) system command, refer to the *System Commands Reference Manual*.

The locales you added or modified are now available to C programs running in an MLS environment.

Effects of Locale Components on C Library Functions

Some of the standard C library functions are affected by the current locale setting for a given locale category. For example, the locale setting for the numeric formatting category affects the `atof` function that is declared in the `<stdlib.h>` standard header.

Note: *The predefined C locale is unaffected by the four locale components defined in the `LOCALES` array. Therefore, the following discussion does not apply to the predefined C locale.*

The standard headers whose functions are affected by the locale settings are as follows:

- Character handling `<ctype.h>`
- Localization `<locale.h>`
- Input/output `<stdio.h>`
- General utilities `<stdlib.h>`
- String handling `<string.h>`
- Date and time `<time.h>`

Note that not all functions declared in these standard headers are affected; only locale-sensitive functions are affected by the locale setting. Refer to the discussion of the particular standard header later in this section for a complete list of the functions that are affected.

The remainder of this section describes the way the functions are affected by the locale setting. The functions are arranged under the standard header that declares each function. For more information on the standard headers or functions, refer to the *C Reference Manual*.

Character Handling <ctype.h>

When you designate the LC_CTYPE value or the LC_ALL value for the category argument of the setlocale function, the character-handling functions that you declare in the <ctype.h> standard header are affected by the ccsversion and spaces set components of the locale that you designate in the same setlocale function call.

Table 8–4 lists the functions affected by the ccsversion and spaces set components of the locale and explains the way the functions operate. Note that the only character-handling functions not affected by a locale are the isdigit function and the isxdigit function.

Table 8–4. Effects of Locale Components on Character-Handling Functions

Function	Explanation
isalnum	Tests for any character that is in the alphabetic data class of the ccsversion for the current locale or in the range from 0 through 9.
isalpha	Tests for any character in the alphabetic data class of the ccsversion for the current locale.
iscntrl	Tests for any character that is not in the presentation set of the ccsversion for the current locale.
isgraph	Tests for any character in the presentation set of the ccsversion for the current locale except a blank (' ').
islower	Tests for any character in the lowercase data class of the ccsversion for the current locale.
isprint	Tests for any character in the presentation set of the ccsversion for the current locale including a blank (' ').
ispunct	Tests for any character in the presentation set of the ccsversion for the current locale that is not in the alphabetic data class, not in the range from 0 through 9, and not a blank (' ').
isspace	Tests for any character in the set of white space characters defined for the spaces set string for the current locale.
isupper	Tests for any character in the uppercase data class of the ccsversion for the current locale.
tolower	Uses the uppercase-to-lowercase mapping table of the ccsversion for the current locale to map an uppercase character to the corresponding lowercase character. If no corresponding lowercase character can be found in the table, the character remains unchanged.
toupper	Uses the lowercase-to-uppercase mapping table of the ccsversion for the current locale to map a lowercase character to the corresponding uppercase character. If no corresponding uppercase character can be found in the table, the character remains unchanged.

Localization <locale.h>

When you designate the LC_NUMERIC value, the LC_MONETARY value, or the LC_ALL value for the category argument of the setlocale function, the localeconv function that you declare in the <locale.h> standard header is affected by the convention component of the locale that you designate in the same setlocale function call.

Table 8–5 lists the members affected by the convention component of the locale, lists the CENTRALSUPPORT library procedure that is used to obtain the contents of the member, and describes the contents of the member.

Table 8–5. Effects of Locale Components on Locale Function Members

Member	Procedure Used	Contents of Member
currency_symbol	CNV_SYMBOLS	The national currency symbol defined in the convention of the locale.
decimal_point	CNV_SYMBOLS	The numeric decimal symbol defined in the convention of the locale.
frac_digits	CNV_INFO	The number of decimal digits defined in the convention of the locale.
grouping	CNV_SYMBOLS	The numeric grouping specification defined in the convention of the locale, converted to the C grouping string format.
int_curr_symbol	CNV_SYMBOLS	The international currency symbol notation defined in the convention of the locale.
int_frac_digits	CNV_INFO	The number of international decimal digits defined in the convention of the locale.
mon_decimal_point	CNV_SYMBOLS	The monetary decimal symbol defined in the convention of the locale.
mon_grouping	CNV_SYMBOLS	The monetary grouping specification defined in the convention of the locale, converted to the C grouping string format.
mon_thousands_sep	CNV_SYMBOLS	The monetary thousands separator symbol defined in the convention of the locale.

Table 8–5. Effects of Locale Components on Locale Function Members

Member	Procedure Used	Contents of Member
n_cs_precedes	CNV_INFO	<p>The value 1 or 0, which designates whether the monetary negative symbol precedes or does not precede a monetary value.</p> <p>The n_cs_precedes member is unavailable if the negative symbol, the national currency symbol, or the placement of the monetary value is not defined in the convention of the locale.</p> <p>Note: The value of this member is derived from the monetary template defined in the convention of the locale. If the monetary template is not properly defined in the convention of the locale, then this member is given a value of CHAR_MAX, which indicates that the value is not available. CHAR_MAX is defined in the C header file <limits.h> as the maximum value for the object of type character.</p>
n_sep_by_space	CNV_INFO	<p>The value 1 or 0, which designates whether the monetary negative symbol is separated from a monetary value by one or more spaces.</p> <p>The n_sep_by_space member is unavailable if the negative symbol, the national currency symbol, or the placement of the monetary value is not defined in the convention of the locale.</p> <p>Note: The value of this member is derived from the monetary template defined in the convention of the locale. If the monetary template is not properly defined in the convention of the locale, then this member is given a value of CHAR_MAX, which indicates that the value is not available. CHAR_MAX is defined in the C header file <limits.h> as the maximum value for the object of type character.</p>
negative_sign	CNV_SYMBOLS	<p>The monetary negative symbol defined in the convention of the locale.</p>

Table 8-5. Effects of Locale Components on Locale Function Members

Member	Procedure Used	Contents of Member
p_cs_precedes	CNV_INFO	<p>The value 1 or 0, which designates whether the monetary positive symbol precedes or does not precede a monetary value.</p> <p>The p_cs_precedes member is unavailable if the positive symbol, the national currency symbol, or the placement of the monetary value is not defined in the convention of the locale.</p> <p>Note: The value of this member is derived from the monetary template defined in the convention of the locale. If the monetary template is not properly defined in the convention of the locale, then this member is given a value of CHAR_MAX, which indicates that the value is not available. CHAR_MAX is defined in the C header file <limits.h> as the maximum value for the object of type character.</p>
p_sep_by_space	CNV_INFO	<p>The value 1 or 0, which designates whether the monetary positive symbol is separated from a monetary value by one or more spaces.</p> <p>The p_sep_by_space member is unavailable if the positive symbol, the national currency symbol, or the placement of the monetary value is not defined in the convention of the locale.</p> <p>Note: The value of this member is derived from the monetary template defined in the convention of the locale. If the monetary template is not properly defined in the convention of the locale, then this member is given a value of CHAR_MAX, which indicates that the value is not available. CHAR_MAX is defined in the C header file <limits.h> as the maximum value for the object of type character.</p>

Table 8-5. Effects of Locale Components on Locale Function Members

Member	Procedure Used	Contents of Member
p_sign_posn	CNV_INFO	<p>A value indicating the position of the positive symbol in relation to the national currency symbol and the monetary value. The value can be from 1 through 4 as follows:</p> <ul style="list-style-type: none"> • The value 1 designates that the positive symbol precedes both the national currency symbol and the monetary value, for example, +\$43.52 or +43.52\$. • The value 2 designates that the positive symbol succeeds both the national currency symbol and the monetary value, for example, \$43.52+ or 43.52\$+. • The value 3 designates that the positive symbol immediately precedes the national currency symbol, for example, 435+F. • The value 4 designates that the positive symbol immediately succeeds the national currency symbol, for example, \$+43.52. <p>The p_sign_posn member is unavailable if the positive symbol, the national currency symbol, or the placement of the monetary amount is not defined in the monetary template of the convention in the locale.</p> <p>Note: The value of this member is derived from the monetary template defined in the convention of the locale. If the monetary template is not properly defined in the convention of the locale, then this member is given a value of CHAR_MAX, which indicates that the value is not available. CHAR_MAX is defined in the C header file <limits.h> as the maximum value for the object of type character.</p>
positive_sign	CNV_SYMBOLS	The monetary positive symbol defined in the convention of the locale.
thousands_sep	CNV_SYMBOLS	The numeric thousands separator symbol defined in the convention of the locale.

Table 8-5. Effects of Locale Components on Locale Function Members

Member	Procedure Used	Contents of Member
n_sign_posn	CNV_INFO	<p>A value indicating the position of the negative symbol in relation to the national currency symbol and the monetary value. The value can be from 0 through 4 as follows:</p> <ul style="list-style-type: none">• The value 0 designates that a negative monetary value is surrounded with enclosure symbols. The monetary left enclosure symbol and the monetary right enclosure symbol are defined as left parenthesis (()) and right parenthesis ()), respectively.• The value 1 designates that the negative symbol precedes both the national currency symbol and the monetary value, for example, -\$65.12 or -65.12\$.• The value 2 designates that the negative symbol succeeds both the national currency symbol and the monetary value, for example, \$65.12- or 65.12\$-.

Table 8-5. Effects of Locale Components on Locale Function Members

Member	Procedure Used	Contents of Member
n_sign_posn (Cont.)	CNV_INFO (Cont.)	<ul style="list-style-type: none"> • The value 3 designates that the negative symbol immediately precedes the national currency symbol, for example, 651-F. • The value 4 designates that the negative symbol immediately succeeds the national currency symbol, for example, \$-65.12. The n_sign_posn member is unavailable in the following two cases: <ul style="list-style-type: none"> • If the negative symbol, the national currency symbol, or the placement of the monetary amount is not defined in the monetary template of the convention in the locale. • If the monetary left enclosure symbol and the monetary right enclosure symbol are defined in the convention as a symbol other than left parenthesis (()) and right parenthesis ()), respectively. <p>Note: The value of this member is derived from the monetary template defined in the convention of the locale. If the monetary template is not properly defined in the convention of the locale, then this member is given a value of CHAR_MAX, which indicates that the value is not available. CHAR_MAX is defined in the C header file <limits.h> as the maximum value for the object of type character.</p>

Input/Output <stdio.h> and General Utilities <stdlib.h>

When you designate the LC_CTYPE value, the LC_NUMERIC value, or the LC_ALL value for the category argument of the setlocale function, the formatted input/output functions and the string conversion functions that you declare in the <stdio.h> standard header and the <stdlib.h> standard header, respectively, are affected by the convention and spaces set components of the locale that you designate in the same setlocale function call.

The effects of assigning a locale other than C to the LC_NUMERIC category are:

- The numeric decimal symbol defined in the convention of the locale is used instead of a decimal point (.).
- The characters designated in the spaces set of the locale are used to determine the characters that are considered white space characters by these functions.

These two effects apply to both the formatted input/output functions and the string conversion functions. These functions are listed in the following table.

Formatted Input/Output Functions		String Conversion Functions	
fprintf	sscanf	atof	strtod
fscanf	vfprintf	atoi	strtol
printf	vprintf	atol	strtoul
scanf	vsprintf		
sprintf			

String Handling <string.h>

When you designate the LC_COLLATE value or the LC_ALL value for the category argument of the setlocale function, then the strcoll function and the strxfrm function that you declare in the <string.h> standard header are affected by the ccsversion component of the locale that you designate in the same setlocale function call.

Table 8–6 describes how the strcoll function and the strxfrm function are affected when the LC_COLLATE value or the LC_ALL value is used in a setlocale function call.

Table 8-6. Effects of Locale Components on String-Handling Functions

Function	Explanation
strcoll	The two strings passed to the strcoll function are compared using the collating sequence in the ccsversion defined for the current locale. The strings are compared using a logical comparison that is based on the ordering sequence values and the priority sequence values of the characters. The collating sequence is defined by the ordering sequence value (OSV) table and by the priority sequence value (PSV) table of the ccsversion.
strxfrm	A string of text passed to the strxfrm function is transformed into another string that contains ordering information for the original string of text. This new string can be used with other strings of ordering information to perform a logical comparison for a designated ccsversion. This process is beneficial when a C program needs to compare the same strings multiple times, but you do not want to transform the strings each time.

Date and Time <time.h>

When you designate the LC_TIME value or the LC_ALL value for the category argument of the setlocale function, the strftime function that you declare in the <time.h> standard header is affected by the convention and language components of the locale that you designate in the same setlocale function call.

The effect of assigning a locale other than C to the LC_TIME category is that some of the time-formatting control characters that are used in the strftime function are satisfied by the internationalized date and time information provided by the CENTRALSUPPORT system library. For example, when the %b control character is used in the strftime function, the C program accesses the CENTRALSUPPORT system library to obtain the abbreviated version of the month name.

The following table lists the time-formatting control characters that access the CENTRALSUPPORT system library when the LC_TIME value or the LC_ALL value is used in a setlocale function call.

Note that none of the other time-formatting control characters that are used in the strftime function use the CENTRALSUPPORT system library.

Time-Formatting Control Characters	Description
%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name
%c	Date/time string

Time-Formatting Control Characters	Description
%p	AM/PM equivalent
%x	Date string
%X	Time string

Section 9

Coded Character Sets and Coded Character Set Mapping Tables

This section introduces the coded character sets and provides the following information:

- How to identify the 8-bit, double-byte, and mixed, multibyte coded character sets supported in the current release
- How to identify the 8-bit, double-byte, and mixed, multibyte coded character set mapping tables supported in the current release
- How to obtain decimal and hexadecimal numbers for characters within any of the 8-bit coded character set tables

You can find 8-bit coded character sets arranged alphabetically in Sections 10 through 12. Double-byte and mixed, multibyte coded character sets are in Section 13. Ccsversions are in Section 14 and standard convention definitions are in Section 15.

Identifying Coded Character Sets on the Release Media

Table 9–1 alphabetically lists the coded character sets supported in the release media. Each coded character set is identified by its common name, its coded character set number (standard ccsnumber), and its library coded character set number (library ccsnumber). The table also gives the coding format associated with the coded character set.

Use the following rules when referring to Table 9–1:

- EBCDIC is accepted as a synonym for ASERIESEBCDIC. Both ASERIESEBCDIC and EBCDIC refer to the same coded character set and, therefore, share the same library ccsnumber.
- CODEPAGE identifies a coded character set on a personal computer.
- ISO 8825 defines ISOVISIBLESTRING and IA5STRING. Both ISOVISIBLESTRING and IA5STRING contain characters from the ISO 646 character set (IRV). Refer to ISO 646 for the coding format.

Note: *Coded character sets produced late in the development of a release may not appear in this Guide until the next revision.*

Table 9–1. Coded Character Sets Supported in the Current Release

Common Name	Standard Ccsnumber	Library Ccsnumber	Coding Format
Arabic20EBCDIC (Previously named Version 20 EBCDIC)	8CS11.01	34	ASERIESEBCDIC
Arabic20ISO (Previously named Version 20 ASCII)	8CS11.00	35	ISO
ArabicV201B	8CS00.00	52	ISO
ArabicV202D	8CS00.00	53	ISO
ASCII	8CS1.00	5	ISO
ASERIESEBCDIC or EBCDIC	8CS1.01	4	ASERIESEBCDIC
ASTASCIIGB2312	816CS00.00	112	ISOMB
ASTASCIIKSC5601	816CS00.00	106	ISOMB
ASTASCIUUTL	816CS00.00	109	ISOMB
ASTUTL	16CS00.00	83	DoubleByte
ASUTL	16CS00.00	82	DoubleByte

Table 9-1. Coded Character Sets Supported in the Current Release

Common Name	Standard Ccsnumber	Library Ccsnumber	Coding Format
CANSupplEBCDIC	8CS51.01	16	STDEBCDIC
CANSupplISO	8CS51.00	17	ISO
CODEPAGE1250	8CS58.03	33	PC (Windows)
CODEPAGE1251	8CS57.03	32	PC (Windows)
CODEPAGE1252	8CS60.03	37	PC (Windows)
CODEPAGE1253	8CS00.00	45	PC (Windows)
CODEPAGE1254	8CS00.00	43	PC (Windows)
CODEPAGE1256	8CS00.00	55	PC (Windows)
CODEPAGE437	8CS59.03	36	PC
CODEPAGE850	8CS49.00	18	PC
CODEPAGE851	8CS43.00	21	PC
CODEPAGE852	8CS27.03	28	PC
CODEPAGE857	8CS00.00	44	PC
CODEPAGE864	8CS00.00	54	PC
CODEPAGE866	8CS56.03	31	PC
CODEPAGE932	816CS00.00	102	PCMB
CODEPAGE936	816CS00.00	113	PCMB
CODEPAGE949	816CS00.00	107	PCMB
CODEPAGE950	816CS00.00	110	PCMB
EBCDICGB2312	816CS00.00	111	EBCDICMB
EBCDICKSC5601	816CS00.00	105	EBCDICMB
EBCDICUTL	816CS00.00	108	EBCDICMB
EUCJp	816CS00.00	103	EUCMB
FrenchArabicE	8CS46.01	22	STDEBCDIC
FrenchArabicISO	8CS46.00	23	ISO
HungarianBTOS	8CS00.00	46	BTOS
IA5STRING	8CS0.01	10	ISO
IBM297	8CS62.01	39	STDEBCDIC
IBMSwedenEBCDIC	8CS00.00	51	STDEBCDIC
ISOVISIBLESTRING	8CS0.00	9	ISO

Table 9-1. Coded Character Sets Supported in the Current Release

Common Name	Standard Ccsnumber	Library Ccsnumber	Coding Format
JapanEBCDIC	8CS00.00	41	ASERIESEBCDIC
JapanEBCDICJBIS8	816CS00.00	100	EBCDICMB
JapanV24	8CS00.00	49	None
JapanV24JB1S8	16CS00.00	114	EBCDICMB
JBIS7	16CS00.00	81	DoubleByte
JBIS8	16CS00.00	80	DoubleByte
JISASCII	8CS00.00	42	ISO
JISASCIIJBIS7	816CS00.00	101	ISOMB
Latin1EBCDIC	8CS26.01	12	STDEBCDIC
Latin1ISO	8CS26.00	13	ISO
Latin2EBCDIC	8CS27.01	26	STDEBCDIC
Latin2ISO	8CS27.00	27	ISO
Latin5EBCDIC	8CS29.01	14	STDEBCDIC
Latin5ISO	8CS29.00	15	ISO
Latin9EBCDIC	8CS00.00	47	STDEBCDIC
Latin9ISO	8CS00.00	48	ISO
LatinCyrillicEBC	8CS55.01	29	STDEBCDIC
LatinCyrillicISO	8CS55.00	30	ISO
LatinGreekBTOS	8CS41.00	25	BTOS
LatinGreekEBCDIC	8CS30.01	19	STDEBCDIC
LatinGreekISO	8CS30.00	20	ISO
LetsJ	816CS00.00	104	2200MB
LocalEBCDIC	8CS00.00	50	ASERIESEBCDIC
MACROMAN	8CS61.03	38	PC
NorwayBTOS	8CS50.00	24	BTOS
UCS2	16CS00.00	85	UCSBMP
UCS2NT	16CS00.00	84	UCSBMPNT

Coded Character Set Mapping Tables

Tables 9–2, 9–3, and 9–4 list the 8-bit, double-byte, and mixed, multibyte coded character set mapping tables supplied in the current release.

Each coded character set mapping table maps data from the coded character set listed in the first column (the source) to data in the coded character set listed in the second column (the destination). Column 3, the source library ccsnumber, identifies the integer value of the INTMODE and EXTMODE file attributes for the source coded character set. Column 4, the destination library ccsnumber, identifies the integer value of the INTMODE and EXTMODE file attributes for the destination coded character set. These coded character set numbers are used as input to the CCSTOCCS_TRANS_TEXT procedure for translating data in 8-bit coded character sets and to the CCSTOCCS_TRANS_TEXT_COMPLEX procedure for translating data in double-byte and mixed, multibyte coded character sets.

Coded character set mapping tables are paired so that you can go from one coded character set to another coded character set and back again. For example, the ASERIESEBCDIC to Latin1EBCDIC mapping table is paired with the Latin1EBCDIC to ASERIESEBCDIC mapping table.

Since a mapping table does not exist for every pair of coded character sets, you might have to call the CCSTOCCS_TRANS_TEXT or CCSTOCCS_TRANS_TEXT_COMPLEX procedure more than once to map the source coded character set to the destination coded character set you want. If you pass two library ccsnumbers that do not have a corresponding coded character set mapping table, the procedures return a DATA NOT FOUND error (4002) as the result. If complex mapping is required and you call on a procedure that does not have complex mapping capability, the CS_COMPLEX_TRAN_REQV error (4004) is returned.

While each coded character set mapping table supports one-to-one mapping from coded character set to coded character set, it is possible for two or more characters to map into the same character. For example, if the source coded character set supports the letter *a* with a tilde (~), the letter *a* with a grave accent (`), and the letter *a* with an acute accent (´), but the destination coded character set does not support any of the variations of the letter *a*, all variations of the letter *a* in the source coded character set might map into the Latin *a* in the destination coded character set. If you then return data to its original coded character set, you need to check each occurrence of the letter *a* to see if it needs a tilde, a grave accent, or an acute accent.

A similar situation can occur if the source coded character set supports the diphthong *ae* and the destination coded character set does not support the diphthong. In this case, original data containing the diphthong *ae* would be mapped to the letter *e*. When this letter *e* is mapped to the original coded character set, it would be mapped to the letter *e*—not to the diphthong *ae*.

Coded Character Sets and Coded Character Set Mapping Tables

Table 9–2 contains the 8-bit coded character set mapping tables available in the current release.

Table 9–2. Eight-Bit CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
Arabic20EBCDIC	Arabic20ISO	34	35
Arabic20EBCDIC	ASCII	34	5
Arabic20EBCDIC	ArabicV201B	34	52
Arabic20EBCDIC	ArabicV202D	34	53
Arabic20EBCDIC	CODEPAGE864	34	54
Arabic20EBCDIC	CODEPAGE1256	34	55
Arabic20ISO	Arabic20EBCDIC	35	34
ASCII	Arabic20EBCDIC	5	34
ASCII	ASERIESEBCDIC	5	4
ASCII	CANSupplEBCDIC	5	16
ASCII	CODEPAGE850	5	18
ASCII	Latin1EBCDIC	5	12
ASCII	Latin2EBCDIC	5	26
ASCII	Latin5EBCDIC	5	14
ASCII	Latin9EBCDIC	5	47
ASCII	LatinCyrillicEBC	5	29
ASCII	LatinGreekEBCDIC	5	19
ASCII	LocalEBCDIC	5	50
ASERIESEBCDIC	ASCII	4	5
ASERIESEBCDIC	CODEPAGE437	4	36
ASERIESEBCDIC	CODEPAGE1252	4	37
ASERIESEBCDIC	Latin1EBCDIC	4	12
CANSupplEBCDIC	ASCII	16	5
CANSupplEBCDIC	CANSupplISO	16	17
CANSupplISO	CANSupplEBCDIC	17	16
CODEPAGE1250	Latin2EBCDIC	33	26
CODEPAGE1250	LocalEBCDIC	33	50
CODEPAGE1251	LatinCyrillicEBC	32	29

Table 9-2. Eight-Bit CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
CODEPAGE1251	LocalEBCDIC	32	50
CODEPAGE1252	ASERIESEBCDIC	37	4
CODEPAGE1252	Latin1EBCDIC	37	12
CODEPAGE1252	Latin5EBCDIC	37	14
CODEPAGE1252	Latin9EBCDIC	37	47
CODEPAGE1252	LocalEBCDIC	37	50
CODEPAGE1253	LatinGreekEBCDIC	45	19
CODEPAGE1253	LocalEBCDIC	45	50
CODEPAGE1254	Latin5EBCDIC	43	14
CODEPAGE1254	LocalEBCDIC	43	50
CODEPAGE1256	Arabic20EBCDIC	55	34
CODEPAGE437	ASERIESEBCDIC	36	4
CODEPAGE437	Latin1EBCDIC	36	12
CODEPAGE437	LocalEBCDIC	36	50
CODEPAGE850	ASCII	18	5
CODEPAGE850	Latin1EBCDIC	18	12
CODEPAGE850	Latin5EBCDIC	18	14
CODEPAGE851	LatinGreekEBCDIC	21	19
CODEPAGE851	LatinGreekISO	21	20
CODEPAGE852	Latin2EBCDIC	28	26
CODEPAGE857	Latin5EBCDIC	44	14
CODEPAGE864	Arabic20EBCDIC	54	34
CODEPAGE866	LatinCyrillicEBC	31	29
HungarianBTOS	Latin2EBCDIC	46	26
IBM297	Latin1EBCDIC	39	12
IBM297	Latin5EBCDIC	39	14
IBMSwedenEBCDIC	Latin1ISO	51	13
JapanEBCDIC	JISASCII	41	42
JapanEBCDIC	JapanV24	41	49
JapanV24	JapanEBCDIC	49	41

Table 9-2. Eight-Bit CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
JapanV24	JISASCII	49	42
JISASCII	JapanEBCDIC	42	41
JISASCII	JapanV24	42	49
Latin1EBCDIC	ASCII	12	5
Latin1EBCDIC	ASERIESEBCDIC	12	4
Latin1EBCDIC	CODEPAGE437	12	36
Latin1EBCDIC	CODEPAGE850	12	18
Latin1EBCDIC	CODEPAGE1252	12	37
Latin1EBCDIC	IBM297	12	39
Latin1EBCDIC	Latin1ISO	12	13
Latin1EBCDIC	LocalEBCDIC	12	50
Latin1EBCDIC	MACROMAN	12	38
Latin1EBCDIC	NorwayBTOS	12	24
Latin1ISO	IBMSwedenEBCDIC	13	51
Latin1ISO	Latin1EBCDIC	13	12
Latin1ISO	Latin5EBCDIC	13	14
Latin1ISO	Latin9EBCDIC	13	47
Latin2EBCDIC	ASCII	26	5
Latin2EBCDIC	CODEPAGE852	26	28
Latin2EBCDIC	CODEPAGE1250	26	33
Latin2EBCDIC	HungarianBTOS	26	46
Latin2EBCDIC	Latin2ISO	26	27
Latin2EBCDIC	LocalEBCDIC	26	50
Latin2ISO	Latin2EBCDIC	27	26
Latin5EBCDIC	ASCII	14	5
Latin5EBCDIC	CODEPAGE850	14	18
Latin5EBCDIC	CODEPAGE857	14	44
Latin5EBCDIC	CODEPAGE1252	14	37
Latin5EBCDIC	CODEPAGE1254	14	43
Latin5EBCDIC	IBM297	14	39

Coded Character Sets and Coded Character Set Mapping Tables

Table 9-2. Eight-Bit CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
Latin5EBCDIC	Latin1ISO	14	13
Latin5EBCDIC	Latin5ISO	14	15
Latin5EBCDIC	LocalEBCDIC	14	50
Latin5EBCDIC	MACROMAN	14	38
Latin5ISO	Latin5EBCDIC	15	14
Latin9EBCDIC	ASCII	47	5
Latin9EBCDIC	CODEPAGE1252	47	37
Latin9EBCDIC	Latin1ISO	47	13
Latin9EBCDIC	Latin9ISO	47	48
Latin9EBCDIC	LocalEBCDIC	47	50
Latin9ISO	Latin9EBCDIC	48	47
LatinCyrillicEBC	ASCII	29	5
LatinCyrillicEBC	CODEPAGE866	29	31
LatinCyrillicEBC	CODEPAGE1251	29	32
LatinCyrillicEBC	LatinCyrillicISO	29	30
LatinCyrillicEBC	LocalEBCDIC	29	50
LatinCyrillicISO	LatinCyrillicEBC	30	29
LatinGreekBTOS	LatinGreekEBCDIC	25	19
LatinGreekEBCDIC	ASCII	19	5
LatinGreekEBCDIC	CODEPAGE851	19	21
LatinGreekEBCDIC	CODEPAGE1253	19	45
LatinGreekEBCDIC	LatinGreekBTOS	19	25
LatinGreekEBCDIC	LatinGreekISO	19	20
LatinGreekEBCDIC	LocalEBCDIC	19	50
LatinGreekISO	CODEPAGE851	20	19
LatinGreekISO	LatinGreekEBCDIC	20	21
LocalEBCDIC	ASCII	50	5
LocalEBCDIC	Codepage437	50	36
LocalEBCDIC	Codepage1250	50	33
LocalEBCDIC	Codepage1251	50	32

Table 9–2. Eight-Bit CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
LocalEBCDIC	Codepage1252	50	37
LocalEBCDIC	Codepage1253	50	45
LocalEBCDIC	Codepage1254	50	43
LocalEBCDIC	Latin1EBCDIC	50	12
LocalEBCDIC	Latin2EBCDIC	50	26
LocalEBCDIC	Latin5EBCDIC	50	14
LocalEBCDIC	Latin9EBCDIC	50	47
LocalEBCDIC	LatinCyrillicEBC	50	29
LocalEBCDIC	UCS2	50	85
MACROMAN	Latin1EBCDIC	38	12
MACROMAN	Latin5EBCDIC	38	14
NorwayBTOS	Latin1EBCDIC	24	12

Table 9–3 lists the double-byte mapping tables available in the current release. This set of mappings supports the conversion of pure 16-bit character sets through the use of the CCSTOCCS_TRANS_TEXT_COMPLEX procedure.

Table 9–3. Double-Byte CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
JBIS7	JBIS8	81	80
JBIS8	JBIS7	80	81

Coded Character Sets and Coded Character Set Mapping Tables

Table 9–4 lists the mixed, multibyte mappings available in the current release through the use of the CCSTOCCS_TRANS_TEXT_COMPLEX procedure.

Table 9–4. Mixed, Multibyte CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
Arabic20EBCDIC	UCS2	34	85
ASCII	UCS2	5	85
ASERIESEBCDIC	UCS2	4	85
ASERIESEBCDIC	UCS2NT	4	84
ASTASCIIGB2312	EBCDICGB2312	112	111
ASTASCIIGB2312	UCS2NT	112	84
ASTASCIKSC5601	EBCDICKSC5601	106	105
ASTASCIKSC5601	UCS2NT	106	84
CANSupplEBCDIC	UCS2	16	85
CODEPAGE932	JapanEBCDICJBIS8	102	100
CODEPAGE932	JapanV24JBIS8	102	114
CODEPAGE932	JBIS8	102	80
CODEPAGE932	UCS2NT	102	84
CODEPAGE936	EBCDICGB2312	113	111
CODEPAGE936	UCS2NT	113	84
CODEPAGE949	EBCDICKSC5601	107	105
CODEPAGE949	UCS2NT	107	84
CODEPAGE950	UCS2NT	110	84
EBCDICGB2312	ASTASCIIGB2312	111	112
EBCDICGB2312	CODEPAGE936	111	113
EBCDICGB2312	UCS2	111	85
EBCDICGB2312	UCS2NT	111	84
EBCDICKSC5601	ASTASCIKSC5601	105	106
EBCDICKSC5601	CODEPAGE949	105	107
EBCDICKSC5601	UCS2	105	85

Table 9–4. Mixed, Multibyte CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
EBCDICKSC5601	UCS2NT	105	84
EUCJp	JapanEBCDICJBIS8	103	100
EUCJp	JapanV24JBIS8	103	114
EUCJp	JBIS8	103	80
EUCJp	UCS2NT	103	84
IBMSwedenEBCDIC	UCS2	51	85
JapanEBCDICJBIS8	CODEPAGE932	100	102
JapanEBCDICJBIS8	EUCJp	100	103
JapanEBCDICJBIS8	JapanEBCDIC <i>See note at the end of this table.</i>	100	41
JapanEBCDICJBIS8	JapanV24JBIS8	100	114
JapanEBCDICJBIS8	JISASCIJBIS7	100	101
JapanEBCDICJBIS8	LetsJ	100	104
JapanEBCDICJBIS8	UCS2	100	85
JapanEBCDICJBIS8	UCS2NT	100	84
JapanV24JBIS8	CODEPAGE932	114	102
JapanV24JBIS8	EUCJp	114	103
JapanV24JBIS8	JapanEBCDICJBIS8	114	100
JapanV24JBIS8	JISASCIJBIS7	114	101
JapanV24JBIS8	LetsJ	114	104
JapanV24JBIS8	UCS2	114	85
JapanV24JBIS8	UCS2NT	114	84
JBIS8	CODEPAGE932	80	102
JBIS8	EUCJp	80	103
JISASCIJBIS7	JapanEBCDICJBIS8	101	100
JISASCIJBIS7	JapanV24JBIS8	101	114
JISASCIJBIS7	UCS2NT	101	84

Table 9–4. Mixed, Multibyte CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
Latin1EBCDIC	UCS2	12	85
Latin2EBCDIC	UCS2	26	85
Latin5EBCDIC	UCS2	14	85
Latin9EBCDIC	UCS2	47	85
LatinCyrillicEBC	UCS2	29	85
LatinGreekEBCDIC	UCS2	19	85
LetsJ	JapanEBCDICJBIS8	104	100
LetsJ	JapanV24JBIS8	104	114
LetsJ	UCS2	104	85
LetsJ	UCS2NT	104	84
LocalEBCDIC	UCS2	50	85
UCS2	IBMSwedenEBCDIC	85	51
UCS2	Arabic20EBCDIC	85	34
UCS2	ASCII	85	5
UCS2	ASERIESEBCDIC	85	4
UCS2	CANSupplEBCDIC	85	16
UCS2	EBCDICGB2312	85	111
UCS2	EBCDICKSC5601	85	105
UCS2	JapanEBCDICJBIS8	85	100
UCS2	JapanV24JBIS8	85	114
UCS2	Latin1EBCDIC	85	12
UCS2	Latin2EBCDIC	85	26
UCS2	Latin5EBCDIC	85	14
UCS2	Latin9EBCDIC	85	47
UCS2	LatinCyrillicEBC	85	29
UCS2	LatinGreekEBCDIC	85	19
UCS2	LetsJ	85	104

Table 9–4. Mixed, Multibyte CCS Mapping Tables in the Current Release

Source Coded Character Set Name	Destination Coded Character Set Name	Source INTMODE and EXTMODE Integer Value	Destination INTMODE and EXTMODE Integer Value
UCS2	LocalEBCDIC	85	50
UCS2NT	ASERIESEBCDIC	84	4
UCS2NT	ASTASCIIGB2312	84	112
UCS2NT	ASTASCIKSC5601	84	106
UCS2NT	CODEPAGE932	84	102
UCS2NT	CODEPAGE936	84	113
UCS2NT	CODEPAGE949	84	107
UCS2NT	CODEPAGE950	84	110
UCS2NT	EBCDICGB2312	84	111
UCS2NT	EBCDICKSC5601	84	105
UCS2NT	EUCJp	84	103
UCS2NT	JapanEBCDICJBIS8	84	100
UCS2NT	JapanV24JBIS8	84	114
UCS2NT	JISASCIJBIS7	84	101
UCS2NT	LetsJ	84	104

Note: The JAPANEBCDICJBIS8 mixed, multibyte coded character set can be mapped to the 8-bit JapanEBCDIC coded character set, sometimes using two 8-bit characters. This mixed, multibyte to multiple 8-bit subCCS mapping enables you to replace double-byte Kanji characters with one or two 8-bit Katakana characters.

Decimal and Hexadecimal Numbers

The following information explains how to obtain decimal and hexadecimal numbers for characters in any of the 8-bit coded character set tables in Sections 10 through 12.

You can obtain the binary number or the octal number of a character by converting the decimal number or the hexadecimal number of the character to a binary number or an octal number.

Obtaining Decimal Numbers for 8-Bit Characters

To obtain the decimal number that corresponds to an 8-bit character in an 8-bit coded character set table, use the column value and row value for the character in the table. The following equation represents the decimal number of the character:

$$(\text{column number} * 16) + (\text{row number})$$

For example, assume that the character you want is displayed in column 08 and row 13 of an 8-bit coded character set table. The decimal value equation is the following:

$$(8 * 16) + 13$$

This equation yields the value 141, the decimal number that corresponds to the character in column 08 and row 13 of the 8-bit coded character set table.

Obtaining Hexadecimal Numbers for 8-Bit Characters

To obtain the hexadecimal number that corresponds to an 8-bit character in an 8-bit coded character set, use the column value and row value for the character in the table.

The column value represents the first digit of the hexadecimal number and the row value represents the second digit of the hexadecimal number. For example, assume that the character you want is displayed in column 05 and row 02 of the 8-bit coded character set table. The hexadecimal number corresponding to the character is 52.

Note that the column values and row values 10 through 15 correspond to the letters A through F. So, a character that is displayed in column 10 and row 14 of the 8-bit coded character set table has a hexadecimal number of AE.

Section 10

Eight-Bit Coded Character Sets: Arabic20EBCDIC to CANSupplISO

This section includes coded character set tables arranged alphabetically from Arabic20EBCDIC to CANSupplISO. You can find additional 8-bit coded character sets in Sections 11 and 12. Double-byte and mixed, multibyte coded character sets are in Section 13. Ccsversions are in Section 14 and standard convention definitions are in Section 15.

Refer to Section 9, “Coded Character Sets and Coded Character Set Mapping Tables,” for a complete list of the coded character sets and mapping tables supported in the current release, and for how to obtain decimal and hexadecimal numbers for characters in any 8-bit table.

How to Read Coded Character Set Tables

In the coded character set tables on the following pages, read the information in the columns from left to right and the information in the rows from top to bottom. A blank box—that is, a box without any symbols or characters—means no assigned value is in that code position. A shaded box means that the position is reserved.

In the columnar listings that accompany each coded character set table, read the information as follows:

- Column 1 notes the code position by its column and row. For example, the notation 04/15 means the fourth column, fifteenth row.
- Column 2 gives the ISO name for the character. For example, *Latin a* is the lowercase letter *a*; *question mark* is the name for the symbol *?*.
- Column 3 shows the character symbol. The heading for column 3, “SYM,” stands for symbol.

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ت	ز	ع	ن	·	{	}	\	0
01	SOH	DC1		PU1	ASP	»	/	ة	a	j	~	ٲ	A	J	APC	1
02	STX	DC2	BPH	SYN	!	لا	:	ث	b	k	s	ٲ	B	K	S	2
03	ETX	DC3	NBH	STS		+	=	ج	c	l	t	ٲ	C	L	T	3
04	ST	OSC	IND	CCH		ٲ	ع	ح	d	m	u	ٲ	D	M	U	4
05	HT	NEL	LF	MW	€	-	؟	خ	e	n	v	ٲ	E	N	V	5
06	SSA	BS	ETB	SPA	%	·	ئ	د	f	o	w	ٲ	F	O	W	6
07	DEL	ESA	ESC	EOT		/	ا	ذ	g	p	x	ٲ	G	P	X	7
08	EPA	CAN	HTS	SOS		,	ى	ر	h	q	y	ٲ	H	Q	Y	8
09	R1	EM	HTJ		«	!	ب	`	i	r	z	ٲ	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	س	غ	ٲ					
11	VT	SS3	PLD	CSI	.	\$,	#	ش	ف	و					
12	FF	IS4	PLU	DC4	<	*	%	@	ط	ق	ي					
13	CR	IS3	ENQ	NAK	()	_	'	ظ	ك	=					
14	SO	IS2	ACK	PM	+	;	>	=	ط	ل						
15	SI	IS1	BEL	SUB	!	^	?	"	ظ	م						MD

Figure 10-1. Arabic20EBCDIC Code Table (8CS11.01)

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	ASP	·	ئ	ص		
01	SOH	DC1	!	1	A	Q	α	q		PU1	!	١	ا	ض		
02	STX	DC2	"	2	B	R	b	r	BPH	PU2		٢	ى	ط		
03	ETX	DC3	#	3	C	S	c	s	NBH	STS		٣	ب	ظ		
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	€	٤	ت	ع		
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	%	٥	ة	غ		
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA		٦	ث	ف		
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA		٧	ج	ق		
08	BS	CAN	{	8	H	X	h	x	HTS	SOS	«	٨	ح	ك		
09	HT	EM	}	9	I	Y	i	y	HTJ		»	٩	خ	ل		
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	لا	,	د	م		
11	VT	ESC	+	;	K	[k	{	PLD	CSI	+	:	ذ	ن		
12	FF	FS	,	<	L	\	l		PLU	ST	‘	:	ر	ه		
13	CR	GS	-	=	M]	m	}	R1	OSC	-	=	ز	و		
14	SO	RS	.	>	N	^	n	~	SS2	PM	·	ء	س	ي		
15	SI	US	Ú	?	O	_	o	DEL	SS3	APL	/	؟	ش	-		

Figure 10-2. Arabic20ISO Code Table (8CS11.00)

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	07/15	Delete	DEL
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Shift out	SO
00/15	00/15	Shift in	SI

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

Figure 10-3. Arabic20EBCDIC and Arabic20ISO Column Listing Columns 0 and 1

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledgement	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substituted character	SUB

**Figure 10-4. Arabic20EBCDIC and Arabic20ISO Column Listing
Columns 2 and 3**

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ASCII	ISO Name	SYM
04/00	02/00	Space character	SP
04/01	10/00	Arabic space character	ASP
04/02	10/01	Arabic exclamation mark	!
04/03			
04/04			
04/05	10/04	Euro sign	€
04/06	10/05	Arabic percent sign	%
04/07			
04/08			
04/09	10/08	Left-pointing double angle quotation mark	«
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ASCII	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	10/09	Right-pointing double angle quotation mark	»
05/02	10/10	Arabic letter lam-alef	ﻻ
05/03	10/11	Arabic plus sign	+
05/04	10/12	Arabic comma	,
05/05	10/13	Arabic hyphen-minus	-
05/06	10/14	Arabic full stop	.
05/07	10/15	Arabic solidus	/
05/08	11/10	Arabic numeric comma	,
05/09	11/11	Arabic semi-colon	:
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

**Figure 10-5. Arabic20EBCDIC and Arabic20ISO Column Listing
Columns 4 and 5**

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ASCII	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	11/12	Arabic colon	:
06/03	11/13	Arabic equal sign	=
06/04	11/14	Arabic letter hamza	ء
06/05	11/15	Arabic question mark	؟
06/06	12/00	Arabic letter yeh with hamza above	يٰ
06/07	12/01	Arabic letter alef	ا
06/08	12/02	Arabic letter alef maksura	ى
06/09	12/03	Arabic letter beh	ب
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ASCII	ISO Name	SYM
07/00	12/04	Arabic letter teh	ت
07/01	12/05	Arabic letter teh marbuta	ة
07/02	12/06	Arabic letter theh	ث
07/03	12/07	Arabic letter jeem	ج
07/04	12/08	Arabic letter hah	ح
07/05	12/09	Arabic letter khah	خ
07/06	12/10	Arabic letter dal	د
07/07	12/11	Arabic letter tahl	ذ
07/08	12/12	Arabic letter reh	ر
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

**Figure 10-6. Arabic20EBCDIC and Arabic20ISO Column Listing
Columns 6 and 7**

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ASCII	ISO Name	SYM
08/00	12/13	Arabic letter zain	ز
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	12/14	Arabic letter seen	س
08/11	12/15	Arabic letter sheen	ش
08/12	13/00	Arabic letter sad	ص
08/13	13/01	Arabic letter dad	ض
08/14	13/02	Arabic letter tah	ط
08/15	13/03	Arabic letter zah	ظ

EBCDIC	ASCII	ISO Name	SYM
09/00	13/04	Arabic letter ain	ع
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	13/05	Arabic letter ghain	غ
09/11	13/06	Arabic letter feh	ف
09/12	13/07	Arabic letter qaf	ق
09/13	13/08	Arabic letter kaf	ك
09/14	13/09	Arabic letter lam	ل
09/15	13/10	Arabic letter meem	م

Figure 10-7. Arabic20EBCDIC and Arabic20ISO Column Listing Columns 8 and 9

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ASCII	ISO Name	SYM
10/00	13/11	Arabic letter noon	ن
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	13/12	Arabic letter heh	ه
10/11	13/13	Arabic letter waw	و
10/12	13/14	Arabic letter yeh	ي
10/13	13/15	Arabic tatweel	=
10/14			
10/15			

EBCDIC	ASCII	ISO Name	SYM
11/00	11/00	Arabic-Indic digit zero	٠
11/01	11/01	Arabic-Indic digit one	١
11/02	11/02	Arabic-Indic digit two	٢
11/03	11/03	Arabic-Indic digit three	٣
11/04	11/04	Arabic-Indic digit four	٤
11/05	11/05	Arabic-Indic digit five	٥
11/06	11/06	Arabic-Indic digit six	٦
11/07	11/07	Arabic-Indic digit seven	٧
11/08	11/08	Arabic-Indic digit eight	٨
11/09	11/09	Arabic-Indic digit nine	٩
11/10			
11/11			
11/12			
11/13			
11/14			
11/15			

Figure 10-8. Arabic20EBCDIC and Arabic20ISO Column Listing Columns 10 and 11

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ASCII	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10			
12/11			
12/12			
12/13			
12/14			
12/15			

EBCDIC	ASCII	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10			
13/11			
13/12			
13/13			
13/14			
13/15			

Figure 10-9. Arabic20EBCDIC and Arabic20ISO Column Listing Columns 12 and 13

Arabic20EBCDIC (8CS11.01) and Arabic20ISO (8CS11.00) Coded Character Sets

EBCDIC	ASCII	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	09/15	Application program command	APC
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10			
14/11			
14/12			
14/13			
14/14			
14/15			

EBCDIC	ASCII	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10			
15/11			
15/12			
15/13			
15/14			
15/15		Mark deleted	MD

Figure 10–10. Arabic20EBCDIC and Arabic20ISO Column Listing Columns14 and 15

ASCII Coded Character Set (8CS1.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p								
01	SOH	DC1	!	1	A	Q	a	q								
02	STX	DC2	"	2	B	R	b	r								
03	ETX	DC3	#	3	C	S	c	s								
04	EOT	DC4	\$	4	D	T	d	t								
05	ENQ	NAK	%	5	E	U	e	u								
06	ACK	SYN	&	6	F	V	f	v								
07	BEL	ETB	'	7	G	W	g	w								
08	BS	CAN	(8	H	X	h	x								
09	HT	EM)	9	I	Y	i	y								
10	LF	SUB	*	:	J	Z	j	z								
11	VT	ESC	+	;	K	[k	{								
12	FF	FS	,	<	L	\	l									
13	CR	GS	-	=	M]	m	}								
14	SO	RS	.	>	N	^	n	~								
15	SI	US	/	?	O	_	o	DEL								

Figure 10-11. ASCII Code Table (8CS1.00)

Notes:

- Although the ASCII coded character set is defined as a 7-bit coded character set, ClearPath and A Series systems treat ASCII as though ASCII were an 8-bit coded character set whose top bit is always zero (0).
- The file separator (FS), group separator (GS), record separator (RS), and unit separator (US) correspond to the information separator 4 (IS4), information separator 3 (IS3), information separator 2 (IS2), and information separator 1 (IS1), respectively.

ASCII Coded Character Set (8CS1.00)

Col/Row	ISO Name	Sym
00/00	Null	NUL
00/01	Start of heading	SO H
00/02	Start text	STX
00/03	End text	ETX
00/04	End of transmission	EOT
00/05	Enquiry	EN Q
00/06	Acknowledge	ACK
00/07	Bell	BEL
00/08	Backspace	BS
00/09	Horizontal tabulation	HT
00/10	Line feed	LF
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

Col/Row	ISO Name	Sym
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04	Device control 4	DC4
01/05	Negative acknowledge	NAK
01/06	Synchronous idle	SYN
01/07	End of transmission block	ETB
01/08	Cancel	CAN
01/09	End medium	EM
01/10	Substituted character	SUB
01/11	Escape	ESC
01/12	File separator	FS
01/13	Group separator	GS
01/14	Record separator	RS
01/15	Unit separator	US

**Figure 10-12. ASCII Column Listing
Columns 0 and 1**

ASCII Coded Character Set (8CS1.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 10-13. ASCII Column Listing
Columns 2 and 3**

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 10-14. ASCII Column Listing
Columns 4 and 5**

ASCII Coded Character Set (8CS1.00)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Delete	DEL

**Figure 10-15. ASCII Column Listing
Columns 6 and 7**

ASERIESEBCDIC Coded Character Set (8CS1.01)

This coded character set is also known as the EBCDIC coded character set.

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE			SP	&	-						{	}	\	0
01	SOH	DC1					/		a	j	~		A	J	APC	1
02	STX	DC2		SYN					b	k	s		B	K	S	2
03	ETX	DC3							c	l	t		C	L	T	3
04									d	m	u		D	M	U	4
05	HT		LF						e	n	v		E	N	V	5
06		BS	ETB						f	o	w		F	O	W	6
07	DEL		ESC	EOT					g	p	x		G	P	X	7
08		CAN							h	q	y		H	Q	Y	8
09		EM						`	i	r	z		I	R	Z	9
10					[]		:								
11	VT		SDO		.	\$,	#								
12	FF	FS	EDO	DC4	<	*	%	@								
13	CR	GS	ENQ	NAK	()	_	'								
14	SO	RS	ACK		+	;	>	=								
15	SI	US	BEL	SUB	!	^	?	"								MD

Figure 10–16. ASERIESEBCDIC Code Table (8CS1.01)

Note: The file separator (FS), group separator (GS), record separator (RS), and unit separator (US) correspond to the information separator 4 (IS4), information separator 3 (IS3), information separator 2 (IS2), and information separator 1 (IS1), respectively. The end of Kanji (EOK) and start of Kanji (SOK) correspond to the end double octet (EDO) and start double octet (SDO), respectively.

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
00/00	Null	NUL
00/01	Start of heading	SO H
00/02	Start text	STX
00/03	End text	ETX
00/04		
00/05	Horizontal tabulation	HT
00/06		
00/07	Delete	DEL
00/08		
00/09		
00/10		
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

Col/Row	ISO Name	Sym
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04		
01/05		
01/06	Backspace	BS
01/07		
01/08	Cancel	CAN
01/09	End medium	EM
01/10		
01/11		
01/12	File separator	FS
01/13	Group separator	GS
01/14	Record separator	RS
01/15	Unit separator	US

**Figure 10-17. ASERIESEBCDIC Column Listing
Columns 0 and 1**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
02/00		
02/01		
02/02		
02/03		
02/04		
02/05	Line feed	LF
02/06	End-of-transmission block	ETB
02/07	Escape	ESC
02/08		
02/09		
02/10		
02/11	Start double octet	SDO
02/12	End double octet	EDO
02/13	Enquiry	ENQ
02/14	Acknowledge	ACK
02/15	Bell	BEL

Col/Row	ISO Name	Sym
03/00		
03/01		
03/02	Synchronous file	SYN
03/03		
03/04		
03/05		
03/06		
03/07	End of transmission	EOT
03/08		
03/09		
03/10		
03/11		
03/12	Device control 4	DC4
03/13	Negative acknowledge	NAK
03/14		
03/15	Substituted character	SUB

**Figure 10-18. ASERIESEBCDIC Column Listing
Columns 2 and 3**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
04/00	Space character	SP
04/01		
04/02		
04/03		
04/04		
04/05		
04/06		
04/07		
04/08		
04/09		
04/10	Left square bracket	[
04/11	Full stop	.
04/12	Less-than sign	<
04/13	Left parenthesis	(
04/14	Plus sign	+
04/15	Exclamation mark	!

Col/Row	ISO Name	Sym
05/00	Ampersand	&
05/01		
05/02		
05/03		
05/04		
05/05		
05/06		
05/07		
05/08		
05/09		
05/10	Right square bracket]
05/11	Dollar sign	\$
05/12	Asterisk	*
05/13	Right parenthesis)
05/14	Semicolon	;
05/15	Circumflex accent	^

**Figure 10-19. ASERIESEBCDIC Column Listing
Columns 4 and 5**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
06/00	Hyphen-minus	-
06/01	Solidus	/
06/02		
06/03		
06/04		
06/05		
06/06		
06/07		
06/08		
06/09		
06/10	Vertical line	
06/11	Comma	,
06/12	Percent sign	%
06/13	Low line	_
06/14	Greater-than sign	>
06/15	Question mark	?

Col/Row	ISO Name	Sym
07/00		
07/01		
07/02		
07/03		
07/04		
07/05		
07/06		
07/07		
07/08		
07/09	Grave accent	`
07/10	Colon	:
07/11	Number sign	#
07/12	Commercial at	@
07/13	Apostrophe	'
07/14	Equal sign	=
07/15	Quotation mark	"

**Figure 10-20. ASERIESEBCDIC Column Listing
Columns 6 and 7**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
08/00		
08/01	Latin small letter a	a
08/02	Latin small letter b	b
08/03	Latin small letter c	c
08/04	Latin small letter d	d
08/05	Latin small letter e	e
08/06	Latin small letter f	f
08/07	Latin small letter g	g
08/08	Latin small letter h	h
08/09	Latin small letter i	i
08/10		
08/11		
08/12		
08/13		
08/14		
08/15		

Col/Row	ISO Name	Sym
09/00		
09/01	Latin small letter j	j
09/02	Latin small letter k	k
09/03	Latin small letter l	l
09/04	Latin small letter m	m
09/05	Latin small letter n	n
09/06	Latin small letter o	o
09/07	Latin small letter p	p
09/08	Latin small letter q	q
09/09	Latin small letter r	r
09/10		
09/11		
09/12		
09/13		
09/14		
09/15		

**Figure 10-21. ASERIESEBCDIC Column Listing
Columns 8 and 9**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
10/00		
10/01	Tilde	~
10/02	Latin small letter s	s
10/03	Latin small letter t	t
10/04	Latin small letter u	u
10/05	Latin small letter v	v
10/06	Latin small letter w	w
10/07	Latin small letter x	x
10/08	Latin small letter y	y
10/09	Latin small letter z	z
10/10		
10/11		
10/12		
10/13		
10/14		
10/15		

Col/Row	ISO Name	Sym
11/00		
11/01		
11/02		
11/03		
11/04		
11/05		
11/06		
11/07		
11/08		
11/09		
11/10		
11/11		
11/12		
11/13		
11/14		
11/15		

**Figure 10-22. ASERIESEBCDIC Column Listing
Columns 10 and 11**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
12/00	Left curly bracket	{
12/01	Latin capital letter A	A
12/02	Latin capital letter B	B
12/03	Latin capital letter C	C
12/04	Latin capital letter D	D
12/05	Latin capital letter E	E
12/06	Latin capital letter F	F
12/07	Latin capital letter G	G
12/08	Latin capital letter H	H
12/09		
12/10		
12/11		
12/12		
12/13		
12/14		
12/15		

Col/Row	ISO Name	Sym
13/00	Right curly bracket	}
13/01	Latin capital letter J	J
13/02	Latin capital letter K	K
13/03	Latin capital letter L	L
13/04	Latin capital letter M	M
13/05	Latin capital letter N	N
13/06	Latin capital letter O	O
13/07	Latin capital letter P	P
13/08	Latin capital letter Q	Q
13/09	Latin capital letter R	R
13/10		
13/11		
13/12		
13/13		
13/14		
13/15		

**Figure 10-23. ASERIESEBCDIC Column Listing
Columns 12 and 13**

ASERIESEBCDIC Coded Character Set (8CS1.01)

Col/Row	ISO Name	Sym
14/00	Reverse solidus	\
14/01	Application program command	APC
14/02	Latin capital letter S	S
14/03	Latin capital letter T	T
14/04	Latin capital letter U	U
14/05	Latin capital letter V	V
14/06	Latin capital letter W	W
14/07	Latin capital letter X	X
14/08	Latin capital letter Y	Y
14/09	Latin capital letter Z	Z
14/10		
14/11		
14/12		
14/13		
14/14		
14/15		

Col/Row	ISO Name	Sym
15/00	Digit 0	0
15/01	Digit 1	1
15/02	Digit 2	2
15/03	Digit 3	3
15/04	Digit 4	4
15/05	Digit 5	5
15/06	Digit 6	6
15/07	Digit 7	7
15/08	Digit 8	8
15/09	Digit 9	9
15/10		
15/11		
15/12		
15/13		
15/14		
15/15	Mark deleted	MD

**Figure 10-24. ASERIESEBCDIC Column Listing
Columns 14 and 15**

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

Row	Column																
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
00	NUL	DLE		DCS	SP	&	-	()	°	μ	ç	{	}	\	0	
01	SOH	DC1		PU1	NBSP	é	/	É	a	j	~	£	A	J	■	1	
02	STX	DC2	BPH	SYN	â	ê	Â	Ê	b	k	s	¥	B	K	S	2	
03	ETX	DC3	NBH	STS	‘	ë	’	Ë	c	l	t	•	C	L	T	3	
04	ST	OSC	IND	CCH	à	è	À	È	d	m	u	©	D	M	U	4	
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v	§	E	N	V	5	
06	SSA	BS	ETB	SPA	“	î	”	Î	f	o	w	¶	F	O	W	6	
07	APC	ESA	ESC	EOT	≤	ï	≥	Ï	g	p	x	¼	G	P	X	7	
08	EPA	CAN	HTS	SOS	ç	—	Ç	l	h	q	y	½	H	Q	Y	8	
09	RI	EM	HTJ		ñ	≠	Ñ	`	i	r	z	¾	I	R	Z	9	
10	SS2	PU2	VTS	SCI	[]		:	«	ª	¡	¬	SHY	¹	²	³	
11	VT	SS3	PLD	CSI	.	\$,	#	»	º	¿	¡	ô	û	Ô	Û	
12	FF	IS4	PLU	DC4	<	*	%	@	⌞	⌟	⌠	⌡	⁴	ü	⁵	Ü	
13	CR	IS3	ENQ	NAK	()	-	·	⌞	⌟	⌠	⌡	⁶	ù	⁷	Û	
14		IS2	ACK	PM	+	;	>	=	⌞	⌟	⌠	⌡	´	ó	ú	Ó	Ú
15		IS1	BEL	SUB	!	^	?	"	±	+	®	-	⁸	⁹	⁰	DEL	

Figure 10–25. CANSupplEBCDIC Code Table (8CS51.01)

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	⁰	À	Û	à	û
01	SOH	DC1	!	1	A	Q	a	q		PU1	i	¹	Á	Ü	á	ü
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	"	²	Â	®	â	©
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	£	³	Ç	§	ç	—
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	¢	⁴	È	ŋ	è	
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	¥	⁵	É	μ	é	⌞
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA	±	⁶	Ê	ª	ê	⌋
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA	´	⁷	Ë	º	ë	⌏
08	BS	CAN	(8	H	X	h	x	HTS	SOS	(⁸	Í	´	í	⌐
09	HT	EM)	9	I	Y	i	y	HTJ)	⁹	Î	´	î	⌑
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	½	¼	Ï	“	ï	⌒
11	VT	ESC	+	;	K	[k	{	PLD	CSI	+	¾	Ñ	”	ñ	⌓
12	FF	FS	,	<	L	\	l		PLU	ST	.	≤	Ó	«	ó	⌔
13	CR	GS	-	=	M]	m	}	RI	OSC	SHY	≠	Ô	»	ô	⌕
14	SO	RS	.	>	N	^	n	~	SS2	PM	•	≥	Ù	°	ù	⌗
15	SI	US	/	?	O	_	o	DEL	SS3	APC	˘	¿	Ú	!	ú	■

Figure 10-26. CANSupplISO Code Table (8CS51.00)

CANSuppIEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Shift out	SO
00/15	00/15	Shift in	SI

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
½	½	Device control 2	DC2
01/03	01/03	Device control 3	DC3
¼	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

Figure 10–27. CANSuppIEBCDIC and CANSupplISO Column Listing Columns 0 and 1

CANSupplIEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledgement	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substituted character	SUB

Figure 10-28. CANSupplIEBCDIC and CANSupplISO Column Listing Columns 2 and 3

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	14/02	Latin small letter a with circumflex accent	â
04/03	13/08	Left single quotation mark	‘
04/04	14/00	Latin small letter a with grave accent	à
04/05	14/01	Latin small letter a with acute accent	á
04/06	13/10	Left double quotation mark	“
04/07	11/12	Less-than or equal-to sign	≤
04/08	14/03	Latin small letter c with cedilla	ç
04/09	14/11	Latin small letter n with tilde	ñ
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	14/05	Latin small letter e with acute accent	é
05/02	14/06	Latin small letter e with circumflex accent	ê
05/03	14/07	Latin small letter e with diaeresis	ë
05/04	14/04	Latin small letter e with grave accent	è
05/05	14/08	Latin small letter i with acute accent	í
05/06	14/09	Latin small letter i with circumflex accent	î
05/07	14/10	Latin small letter i with diaeresis	ï
05/08	15/03	Box drawing light horizontal	—
05/09	11/13	Not equal-to sign	≠
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

Figure 10–29. CANSupplEBCDIC and CANSupplISO Column Listing Columns 4 and 5

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	12/02	Latin capital letter A with circumflex accent	Â
06/03	13/09	Right single quotation mark	'
06/04	12/00	Latin capital letter A with grave accent	À
06/05	12/01	Latin capital letter A with acute accent	Á
06/06	13/11	Right double quotation mark	"
06/07	11/14	Greater-than or equal-to sign	≥
06/08	12/03	Latin capital letter C with cedilla	Ç
06/09	12/11	Latin capital letter N with tilde	Ñ
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	10/08	Superscript left parenthesis	(
07/01	12/05	Latin capital letter E with acute accent	É
07/02	12/06	Latin capital letter E with circumflex accent	Ê
07/03	12/07	Latin capital letter E with diaeresis	Ë
07/04	12/04	Latin capital letter E with grave accent	È
07/05	12/08	Latin capital letter I with acute accent	Í
07/06	12/09	Latin capital letter I with circumflex accent	Î
07/07	12/10	Latin capital letter I with diaeresis	Ï
07/08	15/04	Box drawing light vertical	
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

Figure 10-30. CANSupplEBCDIC and CANSupplISO Column Listing Columns 6 and 7

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
08/00	10/09	Superscript right parenthesis)
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	13/12	Left-pointing double angle quotation mark	«
08/11	13/13	Right-pointing double angle quotation mark	»
08/12	15/05	Box drawing light up and right	└
08/13	15/09	Box drawing light vertical and right	├
08/14	15/11	Box drawing light vertical and left	┤
08/15	10/06	Plus-minus sign	±

EBCDIC	ISO	ISO Name	SYM
09/00	13/14	Degree sign	°
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	13/06	Feminine ordinal indicator	ª
09/11	13/07	Masculine ordinal indicator	º
09/12	15/07	Box drawing light down and left	┘
09/13	10/12	Cedilla	¸
09/14	15/13	Box drawing light vertical and horizontal	┼
09/15	10/11	Superscript plus sign	+

Figure 10-31. CANSupplEBCDIC and CANSupplISO Column Listing Columns 8 and 9

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
10/00	13/05	Micro sign	μ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	10/01	Inverted exclamation mark	¡
10/11	11/15	Inverted question mark	¿
10/12	15/06	Box drawing light up and left	┌
10/13	15/10	Box drawing light horizontal and up	└
10/14	15/12	Box drawing light horizontal and down	┘
10/15	13/02	Registered trademark sign	®

EBCDIC	ISO	ISO Name	SYM
11/00	10/04	Cent sign	¢
11/01	10/03	Pound sign	£
11/02	10/05	Yen sign	¥
11/03	10/14	Middle dot	•
11/04	15/02	Copyright sign	©
11/05	13/03	Paragraph sign	§
11/06	13/04	Pilcrow sign	¶
11/07	11/10	Vulgar fraction one quarter	¼
11/08	10/10	Vulgar fraction one half	½
11/09	11/11	Vulgar fraction three quarters	¾
11/10	15/14	Not sign	¬
11/11	13/15	Broken bar	⌋
11/12	15/08	Box drawing light down and right	┐
11/13	10/02	Diaeresis	¨
11/14	10/07	Acute accent	´
11/15	10/15	Superscript minus sign	⁻

Figure 10–32. CANSupplEBCDIC and CANSupplISO Column Listing Columns 10 and 11

CANSuppIEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hyphen	SHY
12/11	14/13	Latin small letter o with circumflex accent	ô
12/12	11/04	Superscript 4	4
12/13	11/06	Superscript 6	6
12/14	14/12	Latin small letter o with acute accent	ó
12/15	11/08	Superscript 8	8

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	11/01	Superscript 1	1
13/11	15/00	Latin small letter u with circumflex accent	û
13/12	15/01	Latin small letter u with diaeresis	ü
13/13	14/14	Latin small letter u with grave accent	ù
13/14	14/15	Latin small letter u with acute accent	ú
13/15	11/09	Superscript 9	9

Figure 10-33. CANSuppIEBCDIC and CANSupplISO Column Listing Columns 12 and 13

CANSupplEBCDIC (8CS51.01) and CANSupplISO (8CS51.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	15/15	Full block	■
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	11/02	Superscript 2	²
14/11	12/13	Latin capital letter O with circumflex accent	Ô
14/12	11/05	Superscript 5	⁵
14/13	11/07	Superscript 7	⁷
14/14	12/12	Latin capital letter O with acute accent	Ó
14/15	11/00	Superscript 0	⁰

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	11/03	Superscript 3	³
15/11	13/00	Latin capital letter U with circumflex accent	Û
15/12	13/01	Latin capital letter U with diaeresis	Ü
15/13	12/14	Latin capital letter U with grave accent	Ù
15/14	12/15	Latin capital letter U with acute accent	Ú
15/15	07/15	Delete	DEL

Figure 10–34. CANSupplEBCDIC and CANSupplISO Column Listing Columns 14 and 15

CANSuppIEBCDIC (8CS51.01) and CANSuppIISO (8CS51.00) Coded Character Sets

Section 11

Eight-Bit Coded Character Sets: CODEPAGE437 to CODEPAGE1256

This section includes coded character sets arranged numerically from CODEPAGE437 to CODEPAGE1254. These coded character sets apply for use with personal computers. You can find additional 8-bit coded character sets in Sections 10 and 12. Double-byte and mixed, multibyte coded character sets are in Section 13. Ccsversions are in Section 14 and standard convention definitions are in Section 15.

Refer to Section 9, “Coded Character Sets and Coded Character Set Mapping Tables,” for a complete list of the coded character sets and mapping tables supported in the current release, and for how to obtain decimal and hexadecimal numbers for characters in any 8-bit table.

How to Read Coded Character Set Tables

In the coded character set tables on the following pages, read the information in the columns from left to right and the information in the rows from top to bottom. A blank box—that is, a box without any symbols or characters—means no assigned value is in that code position. A shaded box means that the position is reserved.

In the columnar listings that accompany each coded character set table, read the information as follows:

- Column 1 notes the code position by its column and row. For example, the notation 04/15 means the fourth column, fifteenth row.
- Column 2 gives the ISO name for the character. For example, *Latin a* is the lowercase letter *a*; *question mark* is the name for the symbol *?*.
- Column 3 shows the character symbol. The heading for column 3, “SYM,” stands for symbol.

CODEPAGE437 Coded Character Set (8CS49.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	Ç	É	á	☐	ℓ	⊥	α	≡
01	☺	◀	!	1	A	Q	a	q	ü	æ	í	☐	⊥	⊥	β	±
02	☺	↕	"	2	B	R	b	r	é	Æ	ó	☐	⊥	⊥	Γ	≥
03	♥	!!	#	3	C	S	c	s	â	ô	ú		†	ℓ	π	≤
04	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	†	-	ℓ	Σ	ƒ
05	♣	§	%	5	E	U	e	u	à	ò	Ñ	†	†	†	σ	J
06	♠	—	&	6	F	V	f	v	å	û	ª	†	†	π	μ	÷
07	●	↑	'	7	G	W	g	w	ç	ù	º	π	†	†	τ	≈
08	◼	↑	(8	H	X	h	x	ê	ÿ	¿	†	ℓ	†	Φ	°
09	○	↓)	9	I	Y	i	y	ë	Ö	⌋	†	π	⌋	⊖	•
10	◼	→	*	:	J	Z	j	z	è	Ü	⌋		⊥	†	Ω	•
11	♂	—	+	;	K	[k	{	ï	ç	½	π	π	◼	δ	√
12	♀	⌋	,	<	L	\	l	l	î	£	¼	⊥	†	◼	∞	ⁿ
13	♪	↔	-	=	M]	m	}	ì	¥	¡	⊥	=	◼	φ	²
14	♪	▲	.	>	N	^	n	~	Ä	Pts	«	†	†	◼	ε	◼
15	☼	▼	/	?	O	_	o	Δ	Å	f	»	†	⊥	◼	∩	NBSP

Figure 11-1. CODEPAGE437 Code Table (8CS49.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◼
00/09	Medium circle empty	○
00/10	Medium circle negative	◐
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕ —
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-2. CODEPAGE437 Column Listing
Columns 0 and 1

CODEPAGE437 Coded Character Set (8CS49.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-3. CODEPAGE437 Column Listing
Columns 2 and 3**

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-4. CODEPAGE437 Column Listing
Columns 4 and 5**

CODEPAGE437 Coded Character Set (8CS49.00)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	Δ

Figure 11-5. CODEPAGE437 Column Listing Columns 6 and 7

CODEPAGE437 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
08/00	Latin capital letter C with cedilla	Ç
08/01	Latin small letter u with diaeresis	ü
08/02	Latin small letter e with acute accent	é
08/03	Latin small letter a with circumflex accent	â
08/04	Latin small letter a with diaeresis	ä
08/05	Latin small letter a with grave accent	à
08/06	Latin small letter a with ring above	å
08/07	Latin small letter c with cedilla	ç
08/08	Latin small letter e with circumflex accent	ê
08/09	Latin small letter e with diaeresis	ë
08/10	Latin small letter e with grave accent	è
08/11	Latin small letter i with diaeresis	ï
08/12	Latin small letter i with circumflex accent	î
08/13	Latin small letter i with grave accent	ì
08/14	Latin capital letter A with diaeresis	Ä
08/15	Latin capital letter A with ring above	Å

COL/ROW	ISO Name	SYM
09/00	Latin capital letter E with acute accent	É
09/01	Latin small letter ae	æ
09/02	Latin capital letter AE	Æ
09/03	Latin small letter o with circumflex accent	ô
09/04	Latin small letter o with diaeresis	ö
09/05	Latin small letter o with grave accent	ò
09/06	Latin small letter u with circumflex accent	û
09/07	Latin small letter u with grave accent	ù
09/08	Latin small letter y with diaeresis	ÿ
09/09	Latin capital letter O with diaeresis	Ö
09/10	Latin capital letter U with diaeresis	Ü
09/11	Cent sign	¢
09/12	Pound sign	£
09/13	Yen sign	¥
09/14	Peseta sign	Pts
09/15	Latin small letter f with hook	f

**Figure 11-6. CODEPAGE437 Column Listing
Columns 8 and 9**

CODEPAGE437 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
10/00	Latin small letter a with acute accent	á
10/01	Latin small letter i with acute accent	í
10/02	Latin small letter o with acute accent	ó
10/03	Latin small letter u with acute accent	ú
10/04	Latin small letter n with tilde	ñ
10/05	Latin capital letter N with tilde	Ñ
10/06	Feminine ordinal indicator	ª
10/07	Masculine ordinal indicator	º
10/08	Inverted question mark	¿
10/09	Reversed not sign	¬
10/10	Not sign	¬
10/11	Vulgar fraction one half	½
10/12	Vulgar fraction one quarter	¼
10/13	Inverted exclamation mark	¡
10/14	Left-pointing double angle quotation mark	«
10/15	Right-pointing double angle quotation mark	»

COL/ROW	ISO Name	SYM
11/00	Box drawings light shade (25%)	⦿
11/01	Box drawings medium shade (50%)	◐
11/02	Box drawings dark shade (75%)	◑
11/03	Box drawings light vertical	
11/04	Box drawings light vertical and left	├
11/05	Box drawings light and double left	├
11/06	Box drawings double vertical and light left	├
11/07	Box drawings double down and left single	├
11/08	Box drawings down single and left double	├
11/09	Box drawings double vertical and left	├
11/10	Box drawings double vertical	
11/11	Box drawings double down and left	├
11/12	Box drawings double up and left	├
11/13	Box drawings up double and left single	├
11/14	Box drawings up single and left double	├
11/15	Box drawings light down and left	├

**Figure 11-7. CODEPAGE437 Column Listing
Columns 10 and 11**

COL/ROW	ISO Name	SYM
12/00	Box drawings light up and right	L
12/01	Box drawings light horizontal and up	⊥
12/02	Box drawings light horizontal and down	⊤
12/03	Box drawings light vertical and right	⊢
12/04	Box drawings light horizontal	—
12/05	Box drawings light horizontal and vertical	⊕
12/06	Box drawings light vertical and double right	⊣
12/07	Box drawings double vertical and light right	⊤
12/08	Box drawings double up and right	⊥
12/09	Box drawings double down and right	⊣
12/10	Box drawings double horizontal and up	⊥
12/11	Box drawings double horizontal and down	⊤
12/12	Box drawings double vertical and right	⊣
12/13	Box drawings double horizontal	—
12/14	Box drawings double horizontal and vertical	⊕
12/15	Box drawings double horizontal and light up	⊥

COL/ROW	ISO Name	SYM
13/00	Box drawings light horizontal and double up	⊥
13/01	Box drawings double horizontal and light down	⊤
13/02	Box drawings light horizontal and double down	⊤
13/03	Box drawings up double and right single	⊥
13/04	Box drawings up single and right double	⊥
13/05	Box drawings down single and right double	⊣
13/06	Box drawings double down and right single	⊣
13/07	Box drawings double vertical and light horizontal	⊣
13/08	Box drawings double horizontal and light vertical	⊣
13/09	Box drawings light up and left	┌
13/10	Box drawings light down and right	└
13/11	Box drawings solid full block	■
13/12	Box drawings solid lower half block	▀
13/13	Box drawings solid left half block	▄
13/14	Box drawings solid right half block	▆
13/15	Box drawings solid upper half block	▇

**Figure 11-8. CODEPAGE437 Column Listing
Columns 12 and 13**

CODEPAGE437 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
14/00	Greek small letter alpha	α
14/01	Latin small letter sharp s (German)	β
14/02	Greek capital letter Gamma	Γ
14/03	Greek small letter pi	π
14/04	Greek capital letter Sigma	Σ
14/05	Greek small letter sigma	σ
14/06	Micron sign	μ
14/07	Greek small letter tau	τ
14/08	Greek capital letter Phi	Φ
14/09	Greek capital letter Theta	Θ
14/10	Greek capital letter Omega	Ω
14/11	Greek small letter delta	δ
14/12	Infinity	∞
14/13	Greek small letter phi	ϕ
14/14	Greek small letter epsilon	ϵ
14/15	Intersection	\cap

COL/ROW	ISO Name	SYM
15/00	Identical to	\equiv
15/01	Plus-minus sign	\pm
15/02	Greater-than or equal-to sign	\geq
15/03	Less-than or equal-to sign	\leq
15/04	Top half integral	\int
15/05	Bottom half integral	\int
15/06	Division sign	\div
15/07	Almost equal to	\approx
15/08	Degree sign	$^\circ$
15/09	Bullet	\bullet
15/10	Middle dot	\cdot
15/11	Square root sign	$\sqrt{\quad}$
15/12	Superscript latin small letter n	n
15/13	Superscript two	2
15/14	Black square	■
15/15	No-break space	NBSF

**Figure 11-9. CODEPAGE437 Column Listing
Columns 14 and 15**

CODEPAGE850 Coded Character Set (8CS49.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	Ç	É	á	⋮	Ł	ø	Ó	SHY
01	☺	◀	!	1	A	Q	a	q	ü	æ	í	⋮	ł	Ð	ß	±
02	☹	↕	"	2	B	R	b	r	é	Æ	ó	⋮	Ṭ	Ê	Ô	=
03	♥	!!	#	3	C	S	c	s	â	ô	ú		ł	Ë	Ò	¾
04	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	ł	-	È	ö	¶
05	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	ł	ı	Õ	§
06	♠	—	&	6	F	V	f	v	â	û	ª	Â	ã	Í	μ	÷
07	●	↑	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	¸
08	◼	↑	(8	H	X	h	x	ê	ÿ	¿	©	ℓ	Ï	Ɔ	º
09	○	↓)	9	I	Y	i	y	ë	Ö	®		ℝ	Ј	Ú	¨
10	◉	→	*	:	J	Z	j	z	è	Ü	¬		ℚ	Г	Û	•
11	♂	←	+	;	K	l	k	{	ï	ø	½		ℙ	◼	Û	¹
12	♀	└	,	<	L	\	l		î	Œ	¼			◼	Ý	³
13	♪	↔	-	=	M]	m	}	ì	Ø	¡	¢	=	ı	Ý	²
14	♪	▲	.	>	N	^	n	~	Ä	x	«	¥		Ì	-	■
15	☼	▼	/	?	O	_	o	△	Å	f	»	ł	☉	◼	'	NBSP

Figure 11-10. CODEPAGE850 Code Table (8CS49.00)

CODEPAGE850 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up-and down-pointing arrow	↕
01/03	Double exclamation marks	‼
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up-and down-pointing arrow with line below	↕
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left-and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

**Figure 11-11. CODEPAGE850 Column Listing
Columns 0 and 1**

CODEPAGE850 Coded Character Set (8CS49.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-12. CODEPAGE850 Column Listing
Columns 2 and 3**

CODEPAGE850 Coded Character Set (8CS49.00)

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-13. CODEPAGE850 Column Listing
Columns 4 and 5**

CODEPAGE850 Coded Character Set (8CS49.00)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	Δ

**Figure 11-14. CODEPAGE850 Column Listing
Columns 6 and 7**

CODEPAGE850 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
08/00	Latin capital letter C with cedilla	Ç
08/01	Latin small letter u with diaeresis	ü
08/02	Latin small letter e with acute accent	é
08/03	Latin small letter a with circumflex accent	â
08/04	Latin small letter a with diaeresis	ä
08/05	Latin small letter a with grave accent	à
08/06	Latin small letter a with ring above	å
08/07	Latin small letter c with cedilla	ç
08/08	Latin small letter e with circumflex accent	ê
08/09	Latin small letter e with diaeresis	ë
08/10	Latin small letter e with grave accent	è
08/11	Latin small letter i with diaeresis	ï
08/12	Latin small letter i with circumflex accent	î
08/13	Latin small letter i with grave accent	ì
08/14	Latin capital letter A with diaeresis	Ä
08/15	Latin capital letter A with ring above	Å

COL/ROW	ISO Name	SYM
09/00	Latin capital letter E with acute accent	É
09/01	Latin small letter a with e	æ
09/02	Latin capital letter A with E	Æ
09/03	Latin small letter o with circumflex accent	ô
09/04	Latin small letter o with diaeresis	ö
09/05	Latin small letter o with grave accent	ò
09/06	Latin small letter u with circumflex accent	û
09/07	Latin small letter u with grave accent	ù
09/08	Latin small letter y with diaeresis	ÿ
09/09	Latin capital letter O with diaeresis	Ö
09/10	Latin capital letter U with diaeresis	Ü
09/11	Latin small letter o with oblique stroke	ø
09/12	Pound sign	£
09/13	Latin capital letter O with oblique stroke	Ø
09/14	Multiplication sign	x
09/15	Function sign	f

**Figure 11-15. CODEPAGE850 Column Listing
Columns 8 and 9**

CODEPAGE850 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
10/00	Latin small letter a with acute accent	á
10/01	Latin small letter i with acute accent	í
10/02	Latin small letter o with acute accent	ó
10/03	Latin small letter u with acute accent	ú
10/04	Latin small letter n with tilde	ñ
10/05	Latin capital letter N with tilde	Ñ
10/06	Feminine ordinal indicator	ª
10/07	Masculine ordinal indicator	º
10/08	Inverted question mark	¿
10/09	Registered trademark sign	®
10/10	Not sign	¬
10/11	Vulgar fraction one half	½
10/12	Vulgar fraction one quarter	¼
10/13	Inverted exclamation mark	¡
10/14	Left-pointing double angle question mark	«
10/15	Right-pointing double angle question mark	»

COL/ROW	ISO Name	SYM
11/00	Box drawing light shade (25%)	░
11/01	Box drawing medium shade (50%)	▒
11/02	Box drawing dark shade (75%)	▓
11/03	Box drawing light vertical	
11/04	Box drawing light vertical and left	├
11/05	Latin capital letter A with acute accent	Á
11/06	Latin capital letter A with circumflex accent	Â
11/07	Latin capital letter A with grave accent	À
11/08	Copyright sign	©
11/09	Box drawing double vertical and left	┆
11/10	Box drawing double vertical	
11/11	Box drawing double down and left	┆
11/12	Box drawing double up and left	┆
11/13	Cent sign	¢
11/14	Yen sign	¥
11/15	Box drawing light down and left	└

**Figure 11-16. CODEPAGE850 Column Listing
Columns 10 and 11**

CODEPAGE850 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
12/00	Box drawing light up and right	L
12/01	Box drawing light horizontal and up	┌
12/02	Box drawing light horizontal and down	┐
12/03	Box drawing light vertical and right	└
12/04	Box drawing light horizontal	─
12/05	Box drawing light vertical and horizontal	├
12/06	Latin small letter a with tilde	ã
12/07	Latin capital letter A with tilde	Ã
12/08	Box drawing double light up and right	┌
12/09	Box drawing double down and right	┐
12/10	Box drawing double horizontal and up	┌
12/11	Box drawing double horizontal and down	┐
12/12	Box drawing double vertical and right	└
12/13	Box drawing double horizontal	═
12/14	Box drawing double vertical and horizontal	┌
12/15	Currency sign	⌘

COL/ROW	ISO Name	SYM
13/00	Latin small letter eth (Icelandic)	ð
13/01	Latin capital letter eth (Icelandic)	Ð
13/02	Latin capital letter E with circumflex accent	Ê
13/03	Latin capital letter E with diaeresis	Ë
13/04	Latin capital letter E with grave accent	È
13/05	Latin small letter i without dot	ı
13/06	Latin capital letter I with acute accent	Í
13/07	Latin capital letter I with circumflex accent	Î
13/08	Latin capital letter I with diaeresis	Ï
13/09	Box drawing light up and left	└
13/10	Box drawing light down and right	┘
13/11	Box drawing solid full block	■
13/12	Box drawing solid lower half block	▀
13/13	Broken bar	⌘
13/14	Latin capital letter I with grave accent	Ì
13/15	Box drawing solid upper half block	▄

**Figure 11-17. CODEPAGE850 Column Listing
Columns 12 and 13**

CODEPAGE850 Coded Character Set (8CS49.00)

COL/ROW	ISO Name	SYM
14/00	Latin capital letter O with acute accent	Ó
14/01	Latin small letter sharp s (German)	ß
14/02	Latin capital letter O with circumflex accent	Ô
14/03	Latin capital letter O with grave accent	Ò
14/04	Latin small letter o with tilde	õ
14/05	Latin capital letter O with tilde	Õ
14/06	Micron sign	μ
14/07	Latin small letter thorn (Icelandic)	þ
14/08	Latin capital letter thorn (Icelandic)	Þ
14/09	Latin capital letter U with acute accent	Ú
14/10	Latin capital letter U with circumflex accent	Û
14/11	Latin capital letter U with grave accent	Ù
14/12	Latin small letter y with acute accent	ý
14/13	Latin capital letter Y with acute accent	Ý
14/14	Macron	—
14/15	Acute accent	´

COL/ROW	ISO Name	SYM
15/00	Soft hyphen	SHY
15/01	Plus-minus sign	±
15/02	Double low line	=
15/03	Vulgar fraction three quarters	¾
15/04	Pilcrow sign	¶
15/05	Paragraph sign	§
15/06	Division sign	÷
15/07	Cedilla	¸
15/08	Degree sign	°
15/09	Diaeresis	¨
15/10	Middle dot	•
15/11	Superscript 1	¹
15/12	Superscript 3	³
15/13	Superscript 2	²
15/14	Black square	■
15/15	No-break space	NBSP

**Figure 11-18. CODEPAGE850 Column Listing
Columns 14 and 15**

CODEPAGE851 Coded Character Set (8CS43.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	˘	p	Ç	í	ï	☐	Ł	T	ζ	“
01	☺	◀	!	1	A	Q	a	q	ü	Ñ	í	☐	⊥	Y	η	±
02	☺	↑	"	2	B	R	b	r	é	Ó	ó	☐	⊥	Φ	θ	υ
03	♥	!!	#	3	C	S	c	s	â	ô	ó		†	X	ι	φ
04	♦	¶	\$	4	D	T	d	t	ä	ö	A	†	-	Ψ	κ	χ
05	♣	§	%	5	E	U	e	u	à	Ý	B	K	†	Ω	λ	§
06	♠	—	&	6	F	V	f	v	Á	û	Γ	Λ	Π	α	μ	ψ
07	●	↓	'	7	G	W	g	w	ç	ù	Δ	M	P	β	ν	¨
08	◼	↑	(8	H	X	h	x	ê	Ω	E	N	ℓ	γ	ξ	ú
09	○	↓)	9	I	Y	i	y	ë	Ö	Z		ℓ	↓	ο	ì
10	◼	→	*	:	J	Z	j	z	è	Ü	H		ℓ	↓	π	ω
11	♂	←	+	;	K	[k	{	ï	á	½	¶	¶	◼	ρ	ü
12	♀	⊥	,	<	L	\			î	š	Θ	¶	¶	—	σ	ó
13	♪	↔	-	=	M]	m	}	É	é	I	Ξ	=	δ	ς	ώ
14	♪	▲	.	>	N	^	n	~	Ä	ή	«	O	¶	ε	τ	ñ
15	☼	▼	/	?	O	_	o	△	Í	í	»	γ	Σ	—	´	NBSP

Figure 11-19. CODEPAGE851 Code Table (8CS43.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕ —
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

**Figure 11-20. CODEPAGE851 Column Listing
Columns 0 and 1**

CODEPAGE851 Coded Character Set (8CS43.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-21. CODEPAGE851 Column Listing
Columns 2 and 3**

CODEPAGE851 Coded Character Set (8CS43.00)

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-22. CODEPAGE851 Column Listing
Columns 4 and 5**

CODEPAGE851 Coded Character Set (8CS43.00)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-23. CODEPAGE851 Column Listing
Columns 6 and 7**

CODEPAGE851 Coded Character Set (8CS43.00)

COL/ROW	ISO Name	SYM
08/00	Latin capital letter C with cedilla	Ç
08/01	Latin small letter u with diaeresis	ü
08/02	Latin small letter e with acute accent	é
08/03	Latin small letter a with circumflex accent	â
08/04	Latin small letter a with diaeresis	ä
08/05	Latin small letter a with grave accent	à
08/06	Greek capital letter alpha with Greek accent	Á
08/07	Latin small letter c with cedilla	ç
08/08	Latin small letter e with circumflex accent	ê
08/09	Latin small letter e with diaeresis	ë
08/10	Latin small letter e with grave accent	è
08/11	Latin small letter i with diaeresis	ï
08/12	Latin capital letter I with circumflex accent	Î
08/13	Greek capital letter epsilon with Greek accent	É
08/14	Greek capital letter alpha with diaeresis	Ä
08/15	Greek capital letter eta with Greek accent	Ĥ

COL/ROW	ISO Name	SYM
09/00	Greek capital letter iota with Greek accent	Í
09/01	Latin capital letter N with tilde	Ñ
09/02	Greek capital letter omicron with Greek accent	Ó
09/03	Latin small letter o with circumflex accent	ô
09/04	Latin small letter o with diaeresis	ö
09/05	Greek capital letter upsilon with Greek accent	Ý
09/06	Latin small letter u with circumflex accent	û
09/07	Latin small letter u with grave accent	ù
09/08	Greek capital letter omega with Greek accent	Ω
09/09	Latin capital letter O with diaeresis	Ö
09/10	Latin capital letter U with diaeresis	Ü
09/11	Greek small letter alpha with Greek accent	ά
09/12	Pound sign	£
09/13	Greek small letter epsilon with Greek accent	έ
09/14	Greek small letter eta with Greek accent	ή
09/15	Greek small letter iota with Greek accent	ί

**Figure 11-24. CODEPAGE851 Column Listing
Columns 8 and 9**

CODEPAGE851 Coded Character Set (8CS43.00)

COL/ROW	ISO Name	SYM
10/00	Greek small letter iota with diaeresis	ï
10/01	Greek small letter iota with diaeresis and Greek accent	í
10/02	Greek small letter omicron with Greek accent	ó
10/03	Greek small letter upsilon with Greek accent	ύ
10/04	Greek capital letter alpha	Α
10/05	Greek capital letter beta	Β
10/06	Greek capital letter gamma	Γ
10/07	Greek capital letter delta	Δ
10/08	Greek capital letter epsilon	Ε
10/09	Greek capital letter zeta	Ζ
10/10	Greek capital letter eta	Η
10/11	Vulgar fraction one half	½
10/12	Greek capital letter theta	Θ
10/13	Greek capital letter iota	Ι
10/14	Left-pointing double angle quotation mark	«
10/15	Right-pointing double angle quotation mark	»

COL/ROW	ISO Name	SYM
11/00	Box drawing light shade (25%)	░░░░
11/01	Box drawing medium shade (50%)	▒▒▒▒
11/02	Box drawing dark shade (75%)	▓▓▓▓
11/03	Box drawing light vertical	
11/04	Box drawing light vertical and left	├
11/05	Greek capital letter kappa	Κ
11/06	Greek capital letter lamda	Λ
11/07	Greek capital letter mu	Μ
11/08	Greek capital letter nu	Ν
11/09	Box drawing double vertical and left	≡
11/10	Box drawing double vertical	
11/11	Box drawing double down and left	┘
11/12	Box drawing double up and left	└
11/13	Greek capital letter xi	Ξ
11/14	Greek capital letter omicron	Ο
11/15	Box drawing light down and left	┌

**Figure 11-25. CODEPAGE851 Column Listing
Columns 10 and 11**

COL/ROW	ISO Name	SYM
12/00	Box drawing light up and right	L
12/01	Box drawing light horizontal and up	┌
12/02	Box drawing light horizontal and down	┐
12/03	Box drawing light vertical and right	└
12/04	Box drawing light horizontal	─
12/05	Box drawing light vertical and horizontal	┘
12/06	Greek capital letter pi	Π
12/07	Greek capital letter rho	Ρ
12/08	Box drawing double up and right	⏏
12/09	Box drawing double down and right	⏍
12/10	Box drawing double horizontal and up	⏎
12/11	Box drawing double horizontal and down	⏌
12/12	Box drawing double vertical and right	⏊
12/13	Box drawing double horizontal	═
12/14	Box drawing double vertical and horizontal	⏚
12/15	Greek capital letter sigma	Σ

COL/ROW	ISO Name	SYM
13/00	Greek capital letter tau	Τ
13/01	Greek capital letter upsilon	Υ
13/02	Greek capital letter phi	Φ
13/03	Greek capital letter chi	Χ
13/04	Greek capital letter psi	Ψ
13/05	Greek capital letter omega	Ω
13/06	Greek small letter alpha	α
13/07	Greek small letter beta	β
13/08	Greek small letter gamma	γ
13/09	Box drawing light up and left	└
13/10	Box drawing light down and right	┘
13/11	Box drawing solid full block	■
13/12	Box drawing solid lower half block	▀
13/13	Greek small letter delta	δ
13/14	Greek small letter epsilon	ε
13/15	Box drawing solid upper half block	▄

**Figure 11-26. CODEPAGE851 Column Listing
Columns 12 and 13**

CODEPAGE851 Coded Character Set (8CS43.00)

COL/ROW	ISO Name	SYM
14/00	Greek small letter zeta	ζ
14/01	Greek small letter eta	η
14/02	Greek small letter theta	θ
14/03	Greek small letter iota	ι
14/04	Greek small letter kappa	κ
14/05	Greek small letter lamda	λ
14/06	Greek small letter mu	μ
14/07	Greek small letter nu	ν
14/08	Greek small letter xi	ξ
14/09	Greek small letter omicron	ο
14/10	Greek small letter pi	π
14/11	Greek small letter rho	ρ
14/12	Greek small letter sigma	σ
14/13	Greek small letter final sigma	ς
14/14	Greek small letter tau	τ
14/15	Greek accent	˘

COL/ROW	ISO Name	SYM
15/00	Diaeresis and Greek accent	˘
15/01	Plus-minus sign	±
15/02	Greek small letter upsilon	υ
15/03	Greek small letter phi	φ
15/04	Greek small letter chi	χ
15/05	Paragraph sign	§
15/06	Greek small letter psi	ψ
15/07	Diaeresis	¨
15/08	Latin small letter u with acute accent	ú
15/09	Latin small letter i with grave accent	ì
15/10	Greek small letter omega	ω
15/11	Greek small letter upsilon with diaeresis	ϋ
15/12	Greek small letter upsilon with diaeresis and Greek accent	ϋ˘
15/13	Greek small letter omega with Greek accent	ώ
15/14	Latin small letter n with tilde	ñ
15/15	No-break space	NBSP

**Figure 11-27. CODEPAGE851 Column Listing
Columns 14 and 15**

CODEPAGE852 Coded Character Set (8CS27.03)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	Ç	É	á	☐	Ł	đ	Ó	SHY
01	☺	◀	!	1	A	Q	a	q	ü	Ł	í	☐	Ł	Đ	ß	˘
02	☺	↑	”	2	B	R	b	r	é	Í	ó	☐	Ť	Ď	Ô	,
03	♥	!!	#	3	C	S	c	s	â	ô	ú		†	Ě	Ň	˘
04	♦	¶	\$	4	D	T	d	t	ä	ö	À	‡	-	ď	ň	˘
05	♣	§	%	5	E	U	e	u	û	Ľ	ą	Á	†	Ň	ň	§
06	♠	—	&	6	F	V	f	v	č	ĭ	ž	Â	Ă	í	Š	÷
07	●	↓	’	7	G	W	g	w	ç	Ś	ż	Ě	ă	î	š	,
08	◼	↑	(8	H	X	h	x	ł	ś	Ę	Ş	Ł	ě	Ŕ	°
09	○	↓)	9	I	Y	i	y	ë	Ö	e		ŕ	┘	Ú	˘
10	◉	→	*	:	J	Z	j	z	Ő	Ü	┘		┘	ŕ	ř	•
11	♂	←	+	;	K	[k	{	ó	ť	z			◼	Ů	ů
12	♀	┘	,	<	L	\	l		î	ř	č			◼	ý	ř
13	♪	↔	-	=	M]	m	}	Ž	Ł	ş	Ž	=	Ť	Ý	ř
14	♪	▲	.	>	N	^	n	~	Ä	x	«	ž		Ů	‡	■
15	☼	▼	/	?	O	_	o	△	ć	č	»	ł	◊	◼	˘	NBSP

Figure 11-28. CODEPAGE852 Code Table (8CS27.03)

CODEPAGE852 Coded Character Set (8CS27.03)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◼
00/09	Medium circle empty	○
00/10	Medium circle negative	◐
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕ —
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

**Figure 11-29. CODEPAGE852 Column Listing
Columns 0 and 1**

CODEPAGE852 Coded Character Set (8CS27.03)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-30. CODEPAGE852 Column Listing
Columns 2 and 3**

CODEPAGE852 Coded Character Set (8CS27.03)

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-31. CODEPAGE852 Column Listing
Columns 4 and 5**

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	Δ

**Figure 11-32. CODEPAGE852 Column Listing
Columns 6 and 7**

CODEPAGE852 Coded Character Set (8CS27.03)

COL/ROW	ISO Name	SYM
08/00	Latin capital letter C with cedilla	Ç
08/01	Latin small letter u with diaeresis	ü
08/02	Latin small letter e with acute accent	é
08/03	Latin small letter a with circumflex accent	â
08/04	Latin small letter a with diaeresis	ä
08/05	Latin small letter u with ring above	û
08/06	Latin small letter c with acute accent	ć
08/07	Latin small letter c with cedilla	ç
08/08	Latin small letter l with oblique stroke	ł
08/09	Latin small letter e with diaeresis	ë
08/10	Latin capital letter O with double acute accent	Ő
08/11	Latin small letter o with double acute accent	ő
08/12	Latin small letter i with circumflex accent	î
08/13	Latin capital letter Z with acute accent	Ž
08/14	Latin capital letter A with diaeresis	Ä
08/15	Latin capital letter C with acute accent	Ć

COL/ROW	ISO Name	SYM
09/00	Latin capital letter E with acute accent	É
09/01	Latin capital letter L with acute accent	Ł
09/02	Latin small letter l with acute accent	ł
09/03	Latin small letter o with circumflex accent	ô
09/04	Latin small letter o with diaeresis	ö
09/05	Latin capital letter L with caron accent	Ľ
09/06	Latin small letter l with caron accent	ĺ
09/07	Latin Capital letter S with acute accent	Ś
09/08	Latin small letter s with acute accent	ś
09/09	Latin capital letter O with diaeresis	Ö
09/10	Latin capital letter U with diaeresis	Ü
09/11	Latin capital letter T with caron accent	Ĥ
09/12	Latin small letter t with caron accent	ť
09/13	Latin capital letter L with oblique stroke	Ł
09/14	Multiplication sign	x
09/15	Latin small letter c with caron accent	č

**Figure 11-33. CODEPAGE852 Column Listing
Columns 8 and 9**

COL/ROW	ISO Name	SYM
10/00	Latin small letter a with acute accent	á
10/01	Latin small letter i with acute accent	í
10/02	Latin small letter o with acute accent	ó
10/03	Latin small letter u with acute accent	ú
10/04	Latin capital letter A with ogonek	Ą
10/05	Latin small letter a with ogonek	ą
10/06	Latin capital letter Z with caron accent	Ž
10/07	Latin small letter z with caron accent	ž
10/08	Latin capital letter E with ogonek	Ę
10/09	Latin small letter e with ogonek	ę
10/10	Not sign	¬
10/11	Latin small letter z with acute accent	ź
10/12	Latin capital letter C with caron accent	Č
10/13	Latin small letter s with cedilla	š
10/14	Left-pointing double angle quotation mark	«
10/15	Right-pointing double angle quotation mark	»

COL/ROW	ISO Name	SYM
11/00	Box drawing light shade (25%)	░░░░
11/01	Box drawing medium shade (50%)	▒▒▒▒
11/02	Box drawing dark shade (75%)	▓▓▓▓
11/03	Box drawing light vertical	
11/04	Box drawing light vertical and left	├
11/05	Latin capital letter A with acute accent	Á
11/06	Latin capital letter A with circumflex accent	Â
11/07	Latin capital letter E with caron accent	Ě
11/08	Latin capital letter S with cedilla	Š
11/09	Box drawing double vertical and left	┆
11/10	Box drawing double vertical	
11/11	Box drawing double down and left	┆
11/12	Box drawing double up and left	┆
11/13	Latin capital letter Z with dot above	Ž
11/14	Latin small letter z with dot above	ž
11/15	Box drawing light down and left	┆

Figure 11-34. CODEPAGE852 Column Listing
Columns 10 and 11

CODEPAGE852 Coded Character Set (8CS27.03)

COL/ROW	ISO Name	SYM
12/00	Box drawing light up and right	L
12/01	Box drawing light horizontal and up	┌
12/02	Box drawing light horizontal and down	┐
12/03	Box drawing light vertical and right	└
12/04	Box drawing light horizontal	─
12/05	Box drawing light vertical and horizontal	┘
12/06	Latin capital letter A with breve accent	Ă
12/07	Latin small letter a with breve accent	ă
12/08	Box drawing double up and right	⋈
12/09	Box drawing double down and right	⋇
12/10	Box drawing double horizontal and up	⋆
12/11	Box drawing double horizontal and down	⋅
12/12	Box drawing double vertical and right	⋄
12/13	Box drawing double horizontal	═
12/14	Box drawing double vertical and horizontal	⋈
12/15	Currency sign	₯

COL/ROW	ISO Name	SYM
13/00	Latin small letter d with stroke	đ
13/01	Latin capital letter D with stroke	Đ
13/02	Latin capital letter D with caron accent	Ď
13/03	Latin capital letter E with diaeresis	Ë
13/04	Latin small letter d with caron accent	ď
13/05	Latin capital letter N with caron accent	Ň
13/06	Latin capital letter I with acute accent	Í
13/07	Latin capital letter I with circumflex accent	Î
13/08	Latin small letter e with caron accent	ě
13/09	Box drawing light up and left	└
13/10	Box drawing light down and right	┘
13/11	Box drawing solid full block	■
13/12	Box drawing solid lower half block	▀
13/13	Latin capital letter T with cedilla	Ț
13/14	Latin capital letter U with ring above	Ů
13/15	Box drawing solid upper half block	▄

**Figure 11-35. CODEPAGE852 Column Listing
Columns 12 and 13**

CODEPAGE852 Coded Character Set (8CS27.03)

COL/ROW	ISO Name	SYM
14/00	Latin capital letter O with acute accent	Ó
14/01	Latin small letter sharp s (German)	ß
14/02	Latin capital letter O with circumflex accent	Ô
14/03	Latin capital letter N with acute accent	Ñ
14/04	Latin small letter n with acute accent	ñ
14/05	Latin small letter n with caron accent	ň
14/06	Latin capital letter S with caron accent	Š
14/07	Latin small letter s with caron accent	š
14/08	Latin capital letter R with acute accent	Ř
14/09	Latin capital letter U with acute accent	Ú
14/10	Latin small letter r with acute accent	ř
14/11	Latin capital letter U with double acute accent	Ů
14/12	Latin small letter y with acute accent	ý
14/13	Latin capital letter Y with acute accent	Ý
14/14	Latin small letter t with cedilla	ţ
14/15	Acute accent	´

COL/ROW	ISO Name	SYM
15/00	Soft hyphen	SHY
15/01	Double acute accent	¨
15/02	Ogonek accent	˛
15/03	Caron accent	ˇ
15/04	Breve accent	˘
15/05	Paragraph sign	§
15/06	Division sign	÷
15/07	Cedilla	¸
15/08	Degree sign	°
15/09	Diaeresis	¨
15/10	Dot above	•
15/11	Latin small letter u with double acute accent	ů
15/12	Latin capital letter R with caron accent	Ř
15/13	Latin small letter r with caron accent	ř
15/14	Black square	■
15/15	No-break space	NBSP

**Figure 11-36. CODEPAGE852 Column Listing
Columns 14 and 15**

CODEPAGE857 Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	Ç	É	á	☐	Ł	º	Ó	SHY
01	☺	◀	!	1	A	Q	a	q	ü	æ	í	☐	Ł	ª	ß	±
02	☺	↕	"	2	B	R	b	r	é	Æ	ó	☐	Ť	Ê	Ô	
03	♥	!!	#	3	C	S	c	s	â	ô	ú		†	Ë	Ò	¼
04	♦	¶	\$	4	D	T	d	t	ä	ö	ñ	‡	-	È	õ	¶
05	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	†		Õ	§
06	♠	—	&	6	F	V	f	v	â	û	Ğ	Â	ã	Í	µ	÷
07	●	↕	'	7	G	W	g	w	ç	ù	ğ	À	Ã	Î		,
08	◼	↑	(8	H	X	h	x	ê	ï	¿	©	ℓ	Ï	x	°
09	○	↓)	9	I	Y	i	y	ë	Ö	®		ŕ	Ј	Ú	¨
10	◉	→	*	:	J	Z	j	z	è	Ü	¬		⊥	ŕ	Û	·
11	♂	←	+	;	K	[k	{	ï	ø	½		⊥	◼	Û	¹
12	♀	└	,	<	L	\	l		î	š	¼			◼	ì	³
13	♪	↔	-	=	M]	m	}	ı	Ø	ı	¢	=	ı	ÿ	²
14	♪	▲	.	>	N	^	n	~	Ä	Ş	<	¥		Ï	-	◼
15	☼	▼	/	?	O	_	o		Å	ş	>	ı	☐	◼	'	NBSP

Figure 11-37. CODEPAGE857 Code Table (8CS00.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕ —
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-38. CODEPAGE857 Column Listing Columns 0 and 1

CODEPAGE857 Coded Character Set (8CS00.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-39. CODEPAGE857 Column Listing
Column 2 and 3**

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-40. CODEPAGE857 Column Listing
Columns 4 and 5**

CODEPAGE857 Coded Character Set (8CS00.00)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	Δ

**Figure 11-41. CODEPAGE857 Column Listing
Columns 6 and 7**

COL/ROW	ISO Name	SYM
08/00	Latin capital letter C with cedilla	Ç
08/01	Latin small letter u with diaeresis	ü
08/02	Latin small letter e with acute accent	é
08/03	Latin small letter a with circumflex accent	â
08/04	Latin small letter a with diaeresis	ä
08/05	Latin small letter a with grave accent	à
08/06	Latin small letter a with ring above	å
08/07	Latin small letter c with cedilla	ç
08/08	Latin small letter e with circumflex accent	ê
08/09	Latin small letter e with diaeresis	ë
08/10	Latin small letter e with grave accent	è
08/11	Latin small letter i with diaeresis	ï
08/12	Latin small letter i with circumflex accent	î
08/13	Latin small letter dotless i	ı
08/14	Latin capital letter A with diaeresis	Ä
08/15	Latin capital letter A with ring above	Å

COL/ROW	ISO Name	SYM
09/00	Latin capital letter E with acute accent	É
09/01	Latin small letter ae	æ
09/02	Latin capital letter AE	Æ
09/03	Latin small letter o with circumflex accent	ô
09/04	Latin small letter o with diaeresis	ö
09/05	Latin small letter o with grave accent	ò
09/06	Latin small letter u with circumflex accent	û
09/07	Latin small letter u with grave accent	ù
09/08	Latin capital letter I with dot above	İ
09/09	Latin capital letter O with diaeresis	Ö
09/10	Latin capital letter U with diaeresis	Ü
09/11	Latin small letter o with stroke	ø
09/12	Pound sign	£
09/13	Latin capital letter O with stroke	Ø
09/14	Latin capital letter S with cedilla	Ş
09/15	Latin small letter s with cedilla	ş

**Figure 11-42. CODEPAGE857 Column Listing
Columns 8 and 9**

CODEPAGE857 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
10/00	Latin small letter a with acute accent	á
10/01	Latin small letter i with acute accent	í
10/02	Latin small letter o with acute accent	ó
10/03	Latin small letter u with acute accent	ú
10/04	Latin small letter n with tilde	ñ
10/05	Latin capital letter N with tilde	Ñ
10/06	Latin capital letter G with breve	Ğ
10/07	Latin small letter g with breve	ğ
10/08	Inverted question mark	¿
10/09	Registered trademark sign	®
10/10	Not sign	¬
10/11	Vulgar fraction one half	½
10/12	Vulgar fraction one quarter	¼
10/13	Inverted exclamation mark	¡
10/14	Left-pointing double angle quotation mark	«
10/15	Right-pointing double angle quotation mark	»

COL/ROW	ISO Name	SYM
11/00	Box drawing light shade (25%)	░
11/01	Box drawing medium shade (50%)	▒
11/02	Box drawing dark shade (75%)	▓
11/03	Box drawing light vertical	
11/04	Box drawing light vertical and left	├
11/05	Latin capital letter A with acute accent	Á
11/06	Latin capital letter A with circumflex accent	Â
11/07	Latin capital letter A with grave accent	À
11/08	Copyright sign	©
11/09	Box drawing double vertical and left	┆
11/10	Box drawing double vertical	
11/11	Box drawing double down and left	┆
11/12	Box drawing double up and left	┆
11/13	Cent sign	¢
11/14	Yen sign	¥
11/15	Box drawing light down and left	└

**Figure 11-43. CODEPAGE857 Column Listing
Columns 10 and 11**

COL/ROW	ISO Name	SYM
12/00	Box drawing light up and right	L
12/01	Box drawing light up and horizontal	┌
12/02	Box drawing light down and horizontal	└
12/03	Box drawing light vertical and right	┐
12/04	Box drawing light horizontal	─
12/05	Box drawing light vertical and horizontal	├
12/06	Latin small letter a with tilde	ã
12/07	Latin capital letter A with tilde	Ã
12/08	Box drawing double up and right	⋈
12/09	Box drawing double down and right	⋇
12/10	Box drawing double up and horizontal	⋊
12/11	Box drawing double down and horizontal	⋋
12/12	Box drawing double vertical and right	⋌
12/13	Box drawing double horizontal	═
12/14	Box drawing double vertical and horizontal	⋍
12/15	Currency sign	⌘

COL/ROW	ISO Name	SYM
13/00	Masculine ordinal indicator	º
13/01	Feminine ordinal indicator	ª
13/02	Latin capital letter E with circumflex accent	Ê
13/03	Latin capital letter E with diaeresis	Ë
13/04	Latin capital letter E with grave accent	È
13/05		
13/06	Latin capital letter I with acute accent	Í
13/07	Latin capital letter I with circumflex accent	Î
13/08	Latin capital letter I with diaeresis	Ï
13/09	Box drawing light up and left	┐
13/10	Box drawing light down and right	└
13/11	Box drawing solid full block	■
13/12	Box drawing solid lower half block	▀
13/13	Broken bar	⋈
13/14	Latin capital letter I with grave accent	Ì
13/15	Box drawing solid upper half block	▄

Figure 11-44. CODEPAGE857 Column Listing
Columns 12 and 13

CODEPAGE857 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
14/00	Latin capital letter O with acute accent	Ó
14/01	Latin small letter sharp s (German)	ß
14/02	Latin capital letter O with circumflex accent	Ô
14/03	Latin capital letter O with grave accent	Ò
14/04	Latin small letter o with tilde	õ
14/05	Latin capital letter O with tilde	Õ
14/06	Micron sign	μ
14/07		
14/08	Multiplication sign	x
14/09	Latin capital letter U with acute accent	Ú
14/10	Latin capital letter U with circumflex accent	Û
14/11	Latin capital letter U with grave accent	Ù
14/12	Latin small letter i with grave accent	ì
14/13	Latin small letter y with diaeresis	ÿ
14/14	Macron	-
14/15	Acute accent	'

COL/ROW	ISO Name	SYM
15/00	Soft hyphen	SHY
15/01	Plus-minus sign	±
15/02		
15/03	Vulgar fraction three quarters	¾
15/04	Pilcrow sign	¶
15/05	Paragraph sign	§
15/06	Division sign	÷
15/07	Cedilla	¸
15/08	Degree sign	°
15/09	Diaeresis	¨
15/10	Middle dot	•
15/11	Superscript one	¹
15/12	Superscript three	³
15/13	Superscript two	²
15/14	Black square	■
15/15	No-break space	NBSP

**Figure 11-45. CODEPAGE857 Column Listing
Columns 14 and 15**

CODEPAGE864 Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	°	β	NBSP	•	ø	ث	—	=
01	☺	◀	!	1	A	Q	a	q	•	∞	SHY	١	٤	ر	ف	س
02	☺	↕	“	2	B	R	b	r	•	Φ	[٢	أ	ز	ق	ن
03	♥	!!	#	3	C	S	c	s	√	±	£	٣	أ	س	ك	ه
04	♦	¶	\$	4	D	T	d	t	■	½	α	٤	ق	ش	ل	ه
05	♣	§	%	5	E	U	e	u	—	¼	أ	٥	ع	ط	م	و
06	♠	■	&	6	F	V	f	v		≈		٦	د	ض	ذ	ي
07	•	↕	·	7	G	W	g	w	+	«		٧	ا	ظ	ه	غ
08	■	↑	(8	H	X	h	x	⊥	»	ل	٨	ب	ظ	و	ق
09	○	↓)	9	I	Y	i	y	⊥	لا	ب	٩	ة	ع	و	لا
10	◎	→	*	:	J	Z	j	z	⊥	لا	ن	ف	ت	ع	ي	لا
11	♂	←	+	:	K	[k	{	⊥		ث	ء	ث	ا	ط	ل
12	♀	L	,	<	L	\			⊥		•	س	ج	ر	ع	ل
13	♪	↔	-	=	M]	m	}	⊥	لا	ج	ش	ط	+	ن	ي
14	♪	▲	.	>	N	^	n	~	⊥	لا	ح	ص	ط	x	غ	■
15	⚙	▼	/	?	O	_	o	△	⊥		خ	?	د	غ	م	

Figure 11-46. CODEPAGE864 Code Table (8CS00.00)

CODEPAGE864 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-47. CODEPAGE864 Column Listing Columns 0 and 1

Col/Row	ISO Name	SYM
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	SYM
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equals sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-48. CODEPAGE864 Column Listing
Columns 2 and 3**

CODEPAGE864 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left square bracket	{
07/12	Reverse solidus	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-49. CODEPAGE864 Column Listing
Columns 4 and 5**

Col/Row	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left square bracket	{
07/12	Reverse solidus	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-50. CODEPAGE864 Column Listing
Columns 6 and 7**

CODEPAGE864 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
08/00	Degree sign	°
08/01	Middle dot	•
08/02	Bullet operator	●
08/03	Square root	√
08/04	Medium shade	▒
08/05	Box drawing light horizontal	—
08/06	Box drawing light vertical	
08/07	Box drawing light vertical and horizontal	+
08/08	Box drawing light vertical and left	┌
08/09	Box drawing light down and horizontal	└
08/10	Box drawing light up and horizontal	┌
08/11	Box drawing light vertical and right	┐
08/12	Box drawing light down and left	┘
08/13	Box drawing light down and right	┘
08/14	Box drawing light up and right	┐
08/15	Box drawing light up and left	┐

Col/Row	ISO Name	SYM
09/00	Greek small letter beta	β
09/01	Infinity	∞
09/02	Greek small letter phi	φ
09/03	Plus minus sign	±
09/04	Vulgar fraction one half	½
09/05	Vulgar fraction one quarter	¼
09/06	Almost equal to	≈
09/07	Left-pointing double angle quotation mark	«
09/08	Right-pointing double angle quotation mark	»
09/09	Arabic ligature lam with alef with hamza above isolated form	ﻻ
09/10	Arabic ligature lam with alef with hamza above final form	ﻻ
09/11		
09/12		
09/13	Arabic ligature with alef isolated form	ﻻ
09/14	Arabic ligature with alef final form	ﻻ
09/15		

**Figure 11-51. CODEPAGE864 Column Listing
Columns 8 and 9**

Col/Row	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Soft hyphen	SHY
10/02	Arabic letter alef with madda above final form	آ
10/03	Pound sign	£
10/04	Currency sign	¤
10/05	Arabic letter alef with hamza above final form	أ
10/06		
10/07		
10/08	Arabic letter alef final form	ا
10/09	Arabic letter beh isolated form	ب
10/10	Arabic letter teh isolated form	ت
10/11	Arabic letter theh isolated form	ث
10/12	Arabic comma	،
10/13	Arabic letter jeem isolated form	ج
10/14	Arabic letter hah isolated form	ح
10/15	Arabic letter khah isolated form	خ

Col/Row	ISO Name	SYM
11/00	Arabic-indic digit zero	٠
11/01	Arabic-indic digit one	١
11/02	Arabic-indic digit two	٢
11/03	Arabic-indic digit three	٣
11/04	Arabic-indic digit four	٤
11/05	Arabic-indic digit five	٥
11/06	Arabic-indic digit six	٦
11/07	Arabic-indic digit seven	٧
11/08	Arabic-indic digit eight	٨
11/09	Arabic-indic digit nine	٩
11/10	Arabic letter feh isolated form	ف
11/11	Arabic semicolon	؛
11/12	Arabic letter seen isolated form	س
11/13	Arabic letter sheen isolated form	ش
11/14	Arabic letter sad isolated form	ص
11/15	Arabic question mark	؟

**Figure 11-52. CODEPAGE864 Column Listing
Columns 10 and 11**

CODEPAGE864 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
12/00	Cent sign	¢
12/01	Arabic letter hamza isolated form	ء
12/02	Arabic letter alef with madda above isolated form	آ
12/03	Arabic letter alef with hamza above isolated form	أ
12/04	Arabic letter waw with hamza above	ؤ
12/05	Arabic letter ain final form	ع
12/06	Arabic letter yeh with hamza above initial form	د
12/07	Arabic letter alef isolated form	ا
12/08	Arabic letter beh initial form	ب
12/09	Arabic letter teh marbuta isolated form	ة
12/10	Arabic letter teh initial form	ت
12/11	Arabic letter theh initial form	ث
12/12	Arabic letter jeem initial form	ج
12/13	Arabic letter hah initial form	ح
12/14	Arabic letter khah initial form	خ
12/15	Arabic letter dal isolated form	د

Col/Row	ISO Name	SYM
13/00	Arabic letter thal isolated form	ث
13/01	Arabic letter reh isolated form	ر
13/02	Arabic letter zain isolated form	ز
13/03	Arabic letter seen isolated form	س
13/04	Arabic letter sheen initial form	ش
13/05	Arabic letter sad initial form	ط
13/06	Arabic letter dad initial form	ظ
13/07	Arabic letter tah isolated form	ط
13/08	Arabic letter zah isolated form	ظ
13/09	Arabic letter ain initial form	ع
13/10	Arabic letter ghain initial form	غ
13/11	Broken bar	ا
13/12	Not sign	؎
13/13	Division sign	÷
13/14	Multiplication sign	×
13/15	Arabic letter ain isolated form	ع

**Figure 11-53. CODEPAGE864 Column Listing
Columns 12 and 13**

Col/Row	ISO Name	SYM
14/00	Arabic tatweel	—
14/01	Arabic letter feh initial form	ف
14/02	Arabic letter qaf initial form	ق
14/03	Arabic letter kaf initial form	ك
14/04	Arabic letter lam initial form	ل
14/05	Arabic letter meem initial form	م
14/06	Arabic letter noon initial form	ن
14/07	Arabic letter heh initial form	ه
14/08	Arabic letter waw isolated form	و
14/09	Arabic letter alef masurka isolated form	ع
14/10	Arabic letter yeh initial form	ي
14/11	Arabic letter dad isolated form	ط
14/12	Arabic letter ain medial form	ع
14/13	Arabic letter ghain final form	غ
14/14	Arabic letter ghain isolated form	غ
14/15	Arabic letter meem isolated form	م

Col/Row	ISO Name	SYM
15/00	Arabic shadda medial form	◌ّ
15/01	Arabic shadda	ّ
15/02	Arabic letter noon isolated form	ن
15/03	Arabic letter heh isolated form	ه
15/04	Arabic letter heh medial form	هـ
15/05	Arabic letter alef masurka	ع
15/06	Arabic letter yeh final form	يـ
15/07	Arabic letter ghain medial form	غـ
15/08	Arabic letter qaf isolated form	ق
15/09	Arabic ligature lam with alef with madda above isolated form	لا
15/10	Arabic ligature lam with alef with madda above final form	لاّ
15/11	Arabic letter lam isolated form	ل
15/12	Arabic letter kaf isolated form	ك
15/13	Arabic letter yeh isolated form	ي
15/14	Arabic letter black square	■
15/15		

**Figure 11-54. CODEPAGE864 Column Listing
Columns 14 and 15**

CODEPAGE866 Coded Character Set (8CS56.03)

COL ROW	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	∅	@	P	`	p	А	Р	а	☐	Л	Ш	Р	Ё
01	😊	◀	!	1	А	Q	a	q	Б	С	б	☐	Л	Ш	С	ё
02	😄	↕	"	2	В	R	b	r	В	Т	В	☐	Т	Ш	Т	ё
03	♥	!!	#	3	С	S	c	s	Г	У	г		Т	Ш	У	е
04	♦	¶	\$	4	D	T	d	t	Д	Ф	д	┌	—	Е	Ф	İ
05	♣	§	%	5	E	U	e	u	Е	Х	e	≡	+	Ф	Х	ı
06	♠	■	&	6	F	V	f	v	Ж	Ц	ж		≡	П	Ц	ÿ
07	●	↕	'	7	G	W	g	w	Э	Ч	э	П		Ш	Ч	ÿ
08	◉	↑	(8	H	X	h	x	И	Ш	и	≡	Л	≡	Ш	°
09	◯	↓)	9	I	Y	i	y	Й	Щ	й	≡	П	┌	Щ	●
10	◐	→	*	:	J	Z	j	z	К	Ъ	к		≡	Г	Ъ	•
11	♂	←	+	;	K	[k	{	Л	Ы	л	П	≡	■	Ы	√
12	♀	┌	,	<	L	\	l		М	Ь	м	┌		■	Ь	N²
13	🎵	↔	-	=	M]	m	}	Н	Э	н	Ш	≡	■	Э	⌘
14	🎶	▲	.	>	N	^	n	~	О	Ю	о	≡		■	Ю	■
15	☀	▼	/	?	0	_	o	△	П	Я	п	┌	≡	■	Я	NBSP

Figure 11-55. CODEPAGE866 Code Table (8CS66.03)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-56. CODEPAGE866 Column Listing Columns 0 and 1

CODEPAGE866 Coded Character Set (8CS56.03)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-57. CODEPAGE866 Column Listing
Columns 2 and 3**

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-58. CODEPAGE866 Column Listing
Columns 4 and 5**

CODEPAGE866 Coded Character Set (8CS56.03)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-59. CODEPAGE866 Column Listing
Columns 6 and 7**

CODEPAGE866 Coded Character Set (8CS56.03)

COL/ROW	ISO Name	SYM
08/00	Cyrillic capital letter A	А
08/01	Cyrillic capital letter BE	Б
08/02	Cyrillic capital letter VE	В
08/03	Cyrillic capital letter GHE	Г
08/04	Cyrillic capital letter DE	Д
08/05	Cyrillic capital letter IE	Е
08/06	Cyrillic capital letter ZHE	Ж
08/07	Cyrillic capital letter ZE	З
08/08	Cyrillic capital letter I	И
08/09	Cyrillic capital letter short I	Й
08/10	Cyrillic capital letter KA	К
08/11	Cyrillic capital letter EL	Л
08/12	Cyrillic capital letter EM	М
08/13	Cyrillic capital letter EN	Н
08/14	Cyrillic capital letter O	О
08/15	Cyrillic capital letter PE	П

COL/ROW	ISO Name	SYM
09/00	Cyrillic capital letter ER	Р
09/01	Cyrillic capital letter ES	С
09/02	Cyrillic capital letter TE	Т
09/03	Cyrillic capital letter U	У
09/04	Cyrillic capital letter EF	Ф
09/05	Cyrillic capital letter HA	Х
09/06	Cyrillic capital letter TSE	Ц
09/07	Cyrillic capital letter CHE	Ч
09/08	Cyrillic capital letter SHA	Ш
09/09	Cyrillic capital letter SHCHA	Щ
09/10	Cyrillic capital letter hard sign	Ъ
09/11	Cyrillic capital letter YERU	Ы
09/12	Cyrillic capital letter soft sign	Ь
09/13	Cyrillic capital letter E	Э
09/14	Cyrillic capital letter YU	Ю
09/15	Cyrillic capital letter YA	Я

**Figure 11-60. CODEPAGE866 Column Listing
Columns 8 and 9**

CODEPAGE866 Coded Character Set (8CS56.03)

COL/ROW	ISO Name	SYM
10/00	Cyrillic small letter a	а
10/01	Cyrillic small letter be	б
10/02	Cyrillic small letter ve	в
10/03	Cyrillic small letter ghe	г
10/04	Cyrillic small letter de	д
10/05	Cyrillic small letter ie	е
10/06	Cyrillic small letter zhe	ж
10/07	Cyrillic small letter ze	з
10/08	Cyrillic small letter i	и
10/09	Cyrillic small letter short i	й
10/10	Cyrillic small letter ka	к
10/11	Cyrillic small letter el	л
10/12	Cyrillic small letter em	м
10/13	Cyrillic small letter en	н
10/14	Cyrillic small letter o	о
10/15	Cyrillic small letter pe	п

COL/ROW	ISO Name	SYM
11/00	Box drawings light shade (25%)	⠄
11/01	Box drawings medium shade (50%)	⠅
11/02	Box drawings dark shade (75%)	⠆
11/03	Box drawings light vertical	⠇
11/04	Box drawings light vertical and left	⠈
11/05	Box drawings light and double left	⠉
11/06	Box drawings double vertical and light left	⠊
11/07	Box drawings double down and left single	⠋
11/08	Box drawings down single and left double	⠌
11/09	Box drawings double vertical and left	⠍
11/10	Box drawings double vertical	⠎
11/11	Box drawings double down and left	⠏
11/12	Box drawings double up and left	⠑
11/13	Box drawings up double and left single	⠒
11/14	Box drawings up single and left double	⠓
11/15	Box drawings light down and left	⠔

**Figure 11-61. CODEPAGE866 Column Listing
Columns 10 and 11**

COL/ROW	ISO Name	SYM
12/00	Box drawings light up and right	L
12/01	Box drawings light horizontal and up	⊥
12/02	Box drawings light horizontal and down	⊤
12/03	Box drawings light vertical and right	⊢
12/04	Box drawings light horizontal	—
12/05	Box drawings light horizontal and vertical	⊕
12/06	Box drawings light vertical and double right	⊢
12/07	Box drawings double vertical and light right	⊢
12/08	Box drawings double up and right	⊢
12/09	Box drawings double down and right	⊢
12/10	Box drawings double horizontal and up	⊢
12/11	Box drawings double horizontal and down	⊢
12/12	Box drawings double vertical and right	⊢
12/13	Box drawings double horizontal	=
12/14	Box drawings double horizontal and vertical	⊢
12/15	Box drawings double horizontal and light up	⊢

COL/ROW	ISO Name	SYM
13/00	Box drawings light horizontal and double up	⊢
13/01	Box drawings double horizontal and light down	⊢
13/02	Box drawings light horizontal and double down	⊢
13/03	Box drawings up double and right single	⊢
13/04	Box drawings up single and right double	⊢
13/05	Box drawings down single and right double	⊢
13/06	Box drawings double down and right single	⊢
13/07	Box drawings double vertical and light horizontal	⊢
13/08	Box drawings double horizontal and light vertical	⊢
13/09	Box drawings light up and left	⊢
13/10	Box drawings light down and right	⊢
13/11	Box drawings solid full block	■
13/12	Box drawings solid lower half block	▬
13/13	Box drawings solid left half block	▬
13/14	Box drawings solid right half block	▬
13/15	Box drawings solid upper half block	▬

Figure 11-62. CODEPAGE866 Column Listing
Columns 12 and 13

CODEPAGE866 Coded Character Set (8CS56.03)

COL/ROW	ISO Name	SYM
14/00	Cyrillic small letter er	р
14/01	Cyrillic small letter es	с
14/02	Cyrillic small letter te	т
14/03	Cyrillic small letter u	у
14/04	Cyrillic small letter ef	ф
14/05	Cyrillic small letter ha	х
14/06	Cyrillic small letter tse	ц
14/07	Cyrillic small letter che	ч
14/08	Cyrillic small letter sha	ш
14/09	Cyrillic small letter shcha	щ
14/10	Cyrillic small letter hard sign	Ъ
14/11	Cyrillic small letter yeru	ъ
14/12	Cyrillic small letter soft sign	ь
14/13	Cyrillic small letter e	э
14/14	Cyrillic small letter yu	ю
14/15	Cyrillic small letter ya	я

COL/ROW	ISO Name	SYM
15/00	Cyrillic capital letter IO	Ё
15/01	Cyrillic small letter io	ё
15/02	Cyrillic capital letter Ukrainian IE	Є
15/03	Cyrillic small letter Ukrainian ie	є
15/04	Cyrillic capital letter YI	І
15/05	Cyrillic small letter yi	і
15/06	Cyrillic capital letter Byelorussian short U	Ў
15/07	Cyrillic small letter Byelorussian short u	ў
15/08	Degree sign	°
15/09	Bullet	•
15/10	Middle dot	•
15/11	Square root sign	√
15/12	Number acronym	№
15/13	Currency sign	¤
15/14	Box centered square	■
15/15	No-break space	NBSP

**Figure 11-63. CODEPAGE866 Column Listing
Columns 14 and 15**

CODEPAGE1250 Coded Character Set (8CS58.03)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	€		NBSP	°	Ř	Đ	í	ď
01	☺	◀	!	1	A	Q	a	q		`	˘	±	Á	Ň	á	ň
02	☺	↕	"	2	B	R	b	r	,	'	ˇ	˘	Â	Ň	â	ň
03	♥	!!	#	3	C	S	c	s		"	ℒ	ℒ	Ǻ	Ó	ǻ	ó
04	♦	¶	\$	4	D	T	d	t	„	“	⊘	'	Ä	Ô	ä	ô
05	♣	§	%	5	E	U	e	u	...	•	Å	μ	Í	Ŏ	í	ŏ
06	♠	—	&	6	F	V	f	v	†	—	:	¶	Ć	Ö	ć	ö
07	●	↑	'	7	G	W	g	w	‡	—	§	•	Ç	x	ç	÷
08	◼	↑	{	8	H	X	h	x			"	˘	Č	Ř	č	ř
09	○	↓)	9	I	Y	i	y	‰	™	©	q	É	Ů	é	ů
10	◼	→	*	:	J	Z	j	z	š	š	§	§	Ě	Ú	ě	ú
11	♂	—	+	;	K	[k	{	<	>	«	»	Ě	Ú	ě	ú
12	♀	⊥	,	<	L	\	l		ś	ś	¬	ł	Ě	Ü	ě	ü
13	♪	↔	-	=	M]	m	}	ř	ř	_	“	Í	Ý	í	ý
14	♪	▲	.	>	N	^	n	~	ž	ž	®	ř	Î	Ť	î	ť
15	☼	▼	/	?	O	_	o	Δ	ž	ž	ž	ž	Ď	β	ď	•

Figure 11-64. CODEPAGE1250 Code Table (8CS58.03)

CODEPAGE1250 Coded Character Set (8CS58.03)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	‼
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-65. CODEPAGE1250 Column Listing Columns 0 and 1.

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-66. CODEPAGE1250 Column Listing
Columns 2 and 3**

CODEPAGE1250 Coded Character Set (8CS58.03)

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	-

**Figure 11-67. CODEPAGE1250 Column Listing
Columns 4 and 5**

COL/ROW	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

COL/ROW	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-68. CODEPAGE1250 Column Listing
Columns 6 and 7**

CODEPAGE1250 Coded Character Set (8CS58.03)

COL/ROW	ISO Name	SYM
08/00	Euro sign	€
08/01		
08/02	Single low-9 quotation mark	,
08/03		
08/04	Double low-9 quotation mark	"
08/05	Horizontal ellipsis	...
08/06	Dagger	†
08/07	Double dagger	‡
08/08		
08/09	Per mille sign	‰
08/10	Latin capital letter S with caron	Š
08/11	Single left-pointing angle quotation mark	‹
08/12	Latin capital letter S with acute accent	Ś
08/13	Latin capital letter T with caron	Ť
08/14	Latin capital letter Z with caron	Ž
08/15	Latin capital letter Z with acute accent	Ż

COL/ROW	ISO Name	SYM
09/00		
09/01	Left single quotation mark	‘
09/02	Right single quotation mark	’
09/03	Left double quotation mark	“
09/04	Right double quotation mark	”
09/05	Bullet	•
09/06	En dash	—
09/07	Em dash	—
09/08		
09/09	Trademark sign	™
09/10	Latin small letter s with caron	š
09/11	Single right-pointing angle quotation mark	›
09/12	Latin small letter s with acute accent	ś
09/13	Latin small letter t with caron	ť
09/14	Latin small letter z with caron	ž
09/15	Latin small letter z with acute accent	ż

**Figure 11-69. CODEPAGE1250 Column Listing
Columns 8 and 9**

CODEPAGE1250 Coded Character Set (8CS58.03)

COL/ROW	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Combining caron	ˇ
10/02	Combining breve	˘
10/03	Capital letter L with stroke	Ł
10/04	Currency sign	¤
10/05	Latin capital letter with ogonek	Ą
10/06	Broken bar	¡
10/07	Section sign	§
10/08	Combining diaeresis	¨
10/09	Copyright sign	©
10/10	Latin capital letter S with cedilla	Ş
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	–
10/14	Registered sign	®
10/15	Latin capital letter Z with dot above	Ž

COL/ROW	ISO Name	SYM
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Combining ogonek	˛
11/03	Latin small letter l with stroke	ł
11/04	Combining acute	´
11/05	Micron sign	µ
11/06	Pilcrow sign	¶
11/07	Middle dot	•
11/08	Combining cedilla	¸
11/09	Latin small letter a with ogonek	ą
11/10	Latin small letter s with cedilla	ş
11/11	Right-pointing double angle quotation mark	»
11/12	Latin capital L with caron	Ĺ
11/13	Combining double acute	ˆ
11/14	Latin small letter l with caron	ĺ
11/15	Latin small letter z with dot above	ž

**Figure 11-70. CODEPAGE1250 Column Listing
Columns 10 and 11**

CODEPAGE1250 Coded Character Set (8CS58.03)

COL/ROW	ISO Name	SYM
12/00	Latin capital letter R with acute accent	Ŕ
12/01	Latin capital letter A with acute accent	Á
12/02	Latin capital letter A with circumflex	Â
12/03	Latin capital letter A with breve accent	Ă
12/04	Latin capital letter A with diaeresis	Ä
12/05	Latin capital letter L with acute accent	Ĺ
12/06	Latin capital letter C with acute accent	Č
12/07	Latin capital letter C with cedilla	Ç
12/08	Latin capital letter C with caron	Ĉ
12/09	Latin capital letter E with acute accent	É
12/10	Latin capital letter E with ogonek	Ę
12/11	Latin capital letter E with diaeresis	Ë
12/12	Latin capital letter E with caron	Ě
12/13	Latin capital letter I with acute accent	Í
12/14	Latin capital letter I with circumflex	Î
12/15	Latin capital letter D with caron	Ď

COL/ROW	ISO Name	
13/00	Latin capital letter D with stroke	Đ
13/01	Latin capital letter N with acute accent	Ń
13/02	Latin capital letter N with caron	Ñ
13/03	Latin capital letter O with acute accent	Ó
13/04	Latin capital letter O with circumflex	Ô
13/05	Latin capital letter O with double acute accent	Ő
13/06	Latin capital letter O with diaeresis	Ö
13/07	Multiplication sign	×
13/08	Latin capital letter R with caron	Ř
13/09	Latin capital letter U with ring above	Ů
13/10	Latin capital letter U with acute accent	Ú
13/11	Latin capital letter U with double acute accent	Ű
13/12	Latin capital letter U with diaeresis	Ü
13/13	Latin capital letter Y with acute accent	Ý
13/14	Latin capital letter T with cedilla	Ț
13/15	Latin small letter sharp s (German)	ß

**Figure 11-71. CODEPAGE1250 Column Listing
Columns 12 and 13**

COL/ROW	ISO Name	SYM
14/00	Latin small letter r with acute accent	ř
14/01	Latin small letter a with acute accent	á
14/02	Latin small letter a with circumflex accent	â
14/03	Latin small letter a with breve accent	ă
14/04	Latin small letter a with diaeresis	ä
14/05	Latin small letter l with acute accent	ĺ
14/06	Latin small letter c with acute accent	ć
14/07	Latin small letter c with cedilla	ç
14/08	Latin small letter c with caron	č
14/09	Latin small letter e with acute accent	é
14/10	Latin small letter e with ogonek	ę
14/11	Latin small letter e with diaeresis	ë
14/12	Latin small letter e with caron	ě
14/13	Latin small letter i with acute accent	í
14/14	Latin small letter i with circumflex accent	î
14/15	Latin small letter d with caron	ď

COL/ROW	ISO Name	SYM
15/00	Latin small letter d with stroke	ð
15/01	Latin small letter n with acute accent	ń
15/02	Latin small letter n with caron	ň
15/03	Latin small letter o with acute accent	ó
15/04	Latin small letter o with circumflex accent	ô
15/05	Latin small letter o with double acute accent	ő
15/06	Latin small letter o with diaeresis	ö
15/07	Division sign	÷
15/08	Latin small letter r with caron	ř
15/09	Latin small letter u with ring above	û
15/10	Latin small letter u with acute accent	ú
15/11	Latin small letter u with double acute accent	ű
15/12	Latin small letter u with diaeresis	ü
15/13	Latin small letter y with acute accent	ý
15/14	Latin small letter t with cedilla	‡
15/15	Combining dot above	•

**Figure 11-72. CODEPAGE1250 Column Listing
Columns 14 and 15**

CODEPAGE1251 Coded Character Set (8CS57.03)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	Ђ	ђ	NBSP	°	А	Р	а	р
01	☺	◀	!	1	А	Q	а	q	Ѓ	‘	Ў	±	Б	С	б	с
02	☺	↕	"	2	В	Р	в	р	,	'	Ў	І	В	Т	в	т
03	♥	!!	#	3	С	Ѕ	с	ѕ	Ѓ	"	Ј	і	Г	У	г	у
04	♦	¶	\$	4	Д	Т	д	т	„	“	Ѡ	г	Д	Ф	д	ф
05	♣	§	%	5	Е	У	е	у	...	•	Г	μ	Е	Х	е	х
06	♠	▬	&	6	Ф	В	ф	в	†	—	‡	¶	Ж	Ц	ж	ц
07	●	↕	'	7	Г	W	г	w	‡	—	§	·	З	Ч	з	ч
08	◼	↑	(8	Н	Х	н	х	€		Ë	ë	И	Ш	и	ш
09	○	↓)	9	І	У	і	у	‰	™	©	№	И	Щ	й	щ
10	◼	→	*	:	Ј	З	ј	з	Љ	љ	Є	є	К	Ъ	к	ъ
11	♂	←	+	;	К	[к	{	<	>	«	»	Л	Ы	л	ы
12	♀	└	,	<	Л	\	І		Њ	њ	¬	ј	М	Ь	м	ь
13	♎	↔	-	=	М]	м	}	Ќ	ќ	SHY	Ѕ	Н	Э	н	э
14	↖	▲	.	>	Н	^	н	~	Ћ	ћ	®	ѕ	О	Ю	о	ю
15	☼	▼	/	?	О	_	о	△	Ќ	ќ	Ї	ї	П	Я	п	я

Figure 11-73. CODEPAGE1251 Code Table (8CS57.03)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	●
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☼

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕ ┆
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-74. CODEPAGE1251 Column Listing Columns 0 and 1

CODEPAGE1251 Coded Character Set (8CS57.03)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-75. CODEPAGE1251 Column Listing
Columns 2 and 3**

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-76. CODEPAGE1251 Column Listing
Columns 4 and 5**

CODEPAGE1251 Coded Character Set (8CS57.03)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-77. CODEPAGE1251 Column Listing
Columns 6 and 7**

COL/ROW	ISO Name	SYM
08/00	Cyrillic capital letter Serbo-Croatian DJE	Ђ
08/01	Cyrillic capital letter Macedonian GJE	Ѓ
08/02	Single low-9 quotation mark	‘
08/03	Cyrillic small letter Macedonian gje	ђ
08/04	Double low-9 quotation mark	”
08/05	Horizontal ellipsis	…
08/06	Dagger	†
08/07	Double dagger	‡
08/08		
08/09	Per mille sign	‰
08/10	Cyrillic capital letter LJE	Љ
08/11	Single left-pointing angle quotation mark	<
08/12	Cyrillic capital letter NJE	Њ
08/13	Cyrillic capital letter Macedonian KJE	Ќ
08/14	Cyrillic capital letter Serbo-Croatian TSHE	Ћ
08/15	Cyrillic capital letter DZHE	Џ

COL/ROW	ISO Name	SYM
09/00	Cyrillic small letter Serbo-Croatian dje	ђ
09/01	Left single quotation mark	‘
09/02	Right single quotation mark	’
09/03	Left double quotation mark	“
09/04	Right double quotation mark	”
09/05	Bullet	•
09/06	En dash	–
09/07	Em dash	—
09/08		
09/09	Trademark sign	™
09/10	Cyrillic small letter lje	љ
09/11	Single right-pointing angle quotation mark	>
09/12	Cyrillic small letter nje	њ
09/13	Cyrillic small letter Macedonian kje	ќ
09/14	Cyrillic small letter Serbo-Croatian tshe	ћ
09/15	Cyrillic small letter dzhe	џ

**Figure 11-78. CODEPAGE1251 Column Listing
Columns 8 and 9**

CODEPAGE1251 Coded Character Set (8CS57.03)

COL/ROW	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Cyrillic capital letter Byelorussian short U	Ў
10/02	Cyrillic small letter Byelorussian short u	ў
10/03	Cyrillic capital letter JE	Ј
10/04	Currency sign	¤
10/05	Cyrillic capital letter GHE with upturn	Ґ
10/06	Broken bar	‡
10/07	Paragraph sign	§
10/08	Cyrillic capital letter IO	Ё
10/09	Copyright sign	©
10/10	Cyrillic capital letter Ukrainian IE	Є
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	SHY
10/14	Registered trademark sign	®
10/15	Cyrillic capital letter Ukrainian YI	І

COL/ROW	ISO Name	SYM
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Cyrillic capital letter Byelorussian-Ukrainian I	І
11/03	Cyrillic small letter Byelorussian-Ukrainian i	і
11/04	Cyrillic small letter ghe with upturn	ґ
11/05	Micron sign	μ
11/06	Pilcrow sign	¶
11/07	Middle dot	·
11/08	Cyrillic small letter io	ё
11/09	Number acronym	N ^o
11/10	Cyrillic small letter Ukrainian ie	є
11/11	Right-pointing double angle quotation mark	»
11/12	Cyrillic small letter je	ј
11/13	Cyrillic capital letter Macedonian DZE	Ѕ
11/14	Cyrillic small letter Macedonian dze	ѕ
11/15	Cyrillic small letter Ukrainian yi	ї

**Figure 11-79. CODEPAGE1251 Column Listing
Columns 10 and 11**

CODEPAGE1251 Coded Character Set (8CS57.03)

COL/ROW	ISO Name	SYM
12/00	Cyrillic capital letter A	А
12/01	Cyrillic capital letter BE	Б
12/02	Cyrillic capital letter VE	В
12/03	Cyrillic capital letter GHE	Г
12/04	Cyrillic capital letter DE	Д
12/05	Cyrillic capital letter IE	Е
12/06	Cyrillic capital letter ZHE	Ж
12/07	Cyrillic capital letter ZE	З
12/08	Cyrillic capital letter I	И
12/09	Cyrillic capital letter short I	Й
12/10	Cyrillic capital letter KA	К
12/11	Cyrillic capital letter EL	Л
12/12	Cyrillic capital letter EM	М
12/13	Cyrillic capital letter EN	Н
12/14	Cyrillic capital letter O	О
12/15	Cyrillic capital letter PE	П

COL/ROW	ISO Name	SYM
13/00	Cyrillic capital letter ER	Р
13/01	Cyrillic capital letter ES	С
13/02	Cyrillic capital letter TE	Т
13/03	Cyrillic capital letter U	У
13/04	Cyrillic capital letter EF	Ф
13/05	Cyrillic capital letter HA	Х
13/06	Cyrillic capital letter TSE	Ц
13/07	Cyrillic capital letter CHE	Ч
13/08	Cyrillic capital letter SHA	Ш
13/09	Cyrillic capital letter SHCHA	Щ
13/10	Cyrillic capital letter Hard sign	Ъ
13/11	Cyrillic capital letter YERU	Ы
13/12	Cyrillic capital letter Soft sign	Ь
13/13	Cyrillic capital letter E	Э
13/14	Cyrillic capital letter YU	Ю
13/15	Cyrillic capital letter YA	Я

**Figure 11-80. CODEPAGE1251 Column Listing
Columns 12 and 13**

CODEPAGE1251 Coded Character Set (8CS57.03)

COL/ROW	ISO Name	SYM
14/00	Cyrillic small letter a	а
14/01	Cyrillic small letter be	б
14/02	Cyrillic small letter ve	в
14/03	Cyrillic small letter ghe	г
14/04	Cyrillic small letter de	д
14/05	Cyrillic small letter ie	е
14/06	Cyrillic small letter zhe	ж
14/07	Cyrillic small letter ze	з
14/08	Cyrillic small letter i	и
14/09	Cyrillic small letter short i	й
14/10	Cyrillic small letter ka	к
14/11	Cyrillic small letter el	л
14/12	Cyrillic small letter em	м
14/13	Cyrillic small letter en	н
14/14	Cyrillic small letter o	о
14/15	Cyrillic small letter pe	п

COL/ROW	ISO Name	SYM
15/00	Cyrillic small letter er	р
15/01	Cyrillic small letter es	с
15/02	Cyrillic small letter te	т
15/03	Cyrillic small letter u	у
15/04	Cyrillic small letter ef	ф
15/05	Cyrillic small letter ha	х
15/06	Cyrillic small letter tse	ц
15/07	Cyrillic small letter che	ч
15/08	Cyrillic small letter sha	ш
15/09	Cyrillic small letter shcha	щ
15/10	Cyrillic small letter hard sign	ъ
15/11	Cyrillic small letter yeru	ы
15/12	Cyrillic small letter soft sign	ь
15/13	Cyrillic small letter e	э
15/14	Cyrillic small letter yu	ю
15/15	Cyrillic small letter ya	я

**Figure 11-81. CODEPAGE1251 Column Listing
Columns 14 and 15**

CODEPAGE1252 Coded Character Set (8CS60.03)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	€		NBSP	°	À	Ð	à	ð
01	☺	◀	!	1	A	Q	a	q		'	i	±	Á	Ñ	á	ñ
02	☺	↕	"	2	B	R	b	r	,	'	ç	²	Â	Ò	â	ò
03	♥	!!	#	3	C	S	c	s	f	"	£	³	Ã	Ó	ã	ó
04	♦	¶	\$	4	D	T	d	t	„	”	¤	´	Ä	Ô	ä	ô
05	♣	§	%	5	E	U	e	u	...	•	¥	μ	Å	Ö	å	ö
06	♠	—	&	6	F	V	f	v	†	—	‡	¶	Æ	Ö	æ	ö
07	●	↕	'	7	G	W	g	w	‡	—	§	·	Ç	×	ç	÷
08	◼	↑	(8	H	X	h	x	^	˘	¨	˙	È	Ø	è	ø
09	○	↓)	9	I	Y	i	y	‰	™	©	¹	É	Ù	é	ù
10	◼	→	*	:	J	Z	j	z	Š	š	ª	º	Ê	Ú	ê	ú
11	♂	←	+	;	K	[k	{	<	>	«	»	Ë	Û	ë	û
12	♀	└	,	<	L	\	l		Œ	œ	¬	¼	Ì	Ü	ì	ü
13	♪	↔	-	=	M]	m	}			SHY	½	Í	Ý	í	ý
14	♪	▲	.	>	N	^	n	~			®	¾	Î	Þ	î	þ
15	☼	▼	/	?	O	_	o	△		ÿ	-	ı	Ï	ß	ï	ÿ

Figure 11-82. CODEPAGE1252 Code Table (8CS60.03)

CODEPAGE1252 Coded Character Set (8CS60.03)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	∟
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

**Figure 11-83. CODEPAGE1252 Column Listing
Columns 0 and 1**

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-84. CODEPAGE1252 Column Listing
Columns 2 and 3**

CODEPAGE1252 Coded Character Set (8CS60.03)

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	-

**Figure 11-85. CODEPAGE1252 Column Listing
Columns 4 and 5**

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	Δ

**Figure 11-86. CODEPAGE1252 Column Listing
Columns 6 and 7**

CODEPAGE1252 Coded Character Set (8CS60.03)

COL/ROW	ISO Name	SYM
08/00	Euro sign	€
08/01		
08/02	Single low-9 quotation mark	'
08/03	Function sign	f
08/04	Double low-9 quotation mark	"
08/05	Horizontal ellipsis	...
08/06	Dagger	†
08/07	Double dagger	‡
08/08	Circumflex accent	^
08/09	Per mille sign	‰
08/10	Latin capital letter S with caron	Š
08/11	Single left-pointing angle quotation mark	‹
08/12	Latin capital letter OE	Œ
08/13		
08/14		
08/15		

COL/ROW	ISO Name	SYM
09/00		
09/01	Left single quotation mark	‘
09/02	Right single quotation mark	’
09/03	Left double quotation mark	“
09/04	Right double quotation mark	”
09/05	Bullet	•
09/06	En dash	—
09/07	Em dash	—
09/08	Small tilde	~
09/09	Trademark sign	™
09/10	Latin small letter s with caron	š
09/11	Single right-pointing angle quotation mark	›
09/12	Latin small letter oe	œ
09/13		
09/14		
09/15	Latin capital letter Y with diaeresis	ÿ

**Figure 11-87. CODEPAGE1252 Column Listing
Columns 8 and 9**

CODEPAGE1252 Coded Character Set (8CS60.03)

COL/ROW	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Inverted exclamation mark	¡
10/02	Cent sign	¢
10/03	Pound sign	£
10/04	Currency sign	¤
10/05	Yen sign	¥
10/06	Broken bar	¦
10/07	Paragraph sign	¶
10/08	Diaeresis	¨
10/09	Copyright sign	©
10/10	Feminine ordinal indicator	ª
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	SHY
10/14	Registered sign	®
10/15	Macron	—

COL/ROW	ISO Name	
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Superscript two	²
11/03	Superscript three	³
11/04	Acute accent	´
11/05	Micron sign	µ
11/06	Pilcrow sign	¶
11/07	Middle dot	·
11/08	Cedilla	¸
11/09	Superscript one	¹
11/10	Masculine ordinal indicator	º
11/11	Right-pointing double angle quotation mark	»
11/12	Vulgar fraction one quarter	¼
11/13	Vulgar fraction one half	½
11/14	Vulgar fraction three quarters	¾
11/15	Inverted question mark	¿

**Figure 11-88. CODEPAGE1252 Column Listing
Columns 10 and 11**

CODEPAGE1252 Coded Character Set (8CS60.03)

COL/ROW	ISO Name	SYM
12/00	Latin capital letter A with grave accent	À
12/01	Latin capital letter A with acute accent	Á
12/02	Latin capital letter A with circumflex accent	Â
12/03	Latin capital letter A with tilde	Ã
12/04	Latin capital letter A with diaeresis	Ä
12/05	Latin capital letter A with ring above	Å
12/06	Latin capital letter AE	Æ
12/07	Latin capital letter C with cedilla	Ç
12/08	Latin capital letter E with grave accent	È
12/09	Latin capital letter E with acute accent	É
12/10	Latin capital letter E with circumflex accent	Ê
12/11	Latin capital letter E with diaeresis	Ë
12/12	Latin capital letter I with grave accent	Ì
12/13	Latin capital letter I with acute accent	Í
12/14	Latin capital letter I with circumflex accent	Î
12/15	Latin capital letter I with diaeresis	Ï

COL/ROW	ISO Name	SYM
13/00	Latin capital letter eth (Icelandic)	Ð
13/01	Latin capital letter N with tilde	Ñ
13/02	Latin capital letter O with grave accent	Ò
13/03	Latin capital letter O with acute accent	Ó
13/04	Latin capital letter O with circumflex accent	Ô
13/05	Latin capital letter O with tilde	Õ
13/06	Latin capital letter O with diaeresis	Ö
13/07	Multiplication sign	×
13/08	Latin capital letter O with oblique stroke	Ø
13/09	Latin capital letter U with grave accent	Ù
13/10	Latin capital letter U with acute accent	Ú
13/11	Latin capital letter U with circumflex accent	Û
13/12	Latin capital letter U with diaeresis	Ü
13/13	Latin capital letter Y with acute accent	Ý
13/14	Latin capital letter thorn (Icelandic)	Þ
13/15	Latin small letter sharp s (German)	ß

**Figure 11-89. CODEPAGE1252 Column Listing
Columns 12 and 13**

COL/ROW	ISO Name	SYM
14/00	Latin small letter a with grave accent	à
14/01	Latin small letter a with acute accent	á
14/02	Latin small letter a with circumflex accent	â
14/03	Latin small letter a with tilde	ã
14/04	Latin small letter a with diaeresis	ä
14/05	Latin small letter a with ring above	å
14/06	Latin small letter ae	æ
14/07	Latin small letter c with cedilla	ç
14/08	Latin small letter e with grave accent	è
14/09	Latin small letter e with acute accent	é
14/10	Latin small letter e with circumflex accent	ê
14/11	Latin small letter e with diaeresis	ë
14/12	Latin small letter i with grave accent	ì
14/13	Latin small letter i with acute accent	í
14/14	Latin small letter i with circumflex accent	î
14/15	Latin small letter i with diaeresis	ï

COL/ROW	ISO Name	SYM
15/00	Latin small letter eth (Icelandic)	ð
15/01	Latin small letter n with tilde	ñ
15/02	Latin small letter o with grave accent	ò
15/03	Latin small letter o with acute accent	ó
15/04	Latin small letter o with circumflex accent	ô
15/05	Latin small letter o with tilde	õ
15/06	Latin small letter o with diaeresis	ö
15/07	Division sign	÷
15/08	Latin small letter o with oblique stroke	ø
15/09	Latin small letter u with grave accent	ù
15/10	Latin small letter u with acute accent	ú
15/11	Latin small letter u with circumflex accent	û
15/12	Latin small letter u with diaeresis	ü
15/13	Latin small letter y with acute accent	ý
15/14	Latin small letter thorn (Icelandic)	þ
15/15	Latin small letter y with diaeresis	ÿ

**Figure 11-90. CODEPAGE1252 Column Listing
Columns 14 and 15**

CODEPAGE1253 Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	€		NBSP	°	í	Π	ú	π
01	☺	◀	!	1	A	Q	a	q		`	´	±	A	P	α	ρ
02	☺	↕	"	2	B	R	b	r	,	'	Á	²	B		β	ς
03	♥	!!	#	3	C	S	c	s	f	"	£	³	Γ	Σ	γ	σ
04	♦	¶	\$	4	D	T	d	t	„	“	¤	´	Δ	T	δ	τ
05	♣	§	%	5	E	U	e	u	...	•	¥	μ	E	Υ	ε	υ
06	♠	—	&	6	F	V	f	v	†	-	¡	¶	Z	Φ	ζ	φ
07	●	↕	'	7	G	W	g	w	‡	—	§	•	H	X	η	χ
08	◼	↑	(8	H	X	h	x			¨	´E	Θ	Ψ	θ	ψ
09	○	↓)	9	I	Y	i	y	‰	™	©	´H	I	Ω	ι	ω
10	◼	→	*	:	J	Z	j	z			ª	´I	K	´I	κ	ı
11	♂	—	+	;	K	[k	{	<	>	«	»	Λ	Ÿ	λ	ü
12	♀	└	,	<	L	\	l				¬	ˆO	M	´α	μ	´ó
13	♪	→	-	=	M]	m	}			SHY	½	N	´ε	v	´ú
14	♪	▲	.	>	N	^	n	~			®	´Υ	Ξ	´η	ξ	´ó
15	☼	▼	/	?	O	_	o				¯	ˆΩ	O	´ı	o	

Figure 11-91. CODEPAGE1253 Code Table (8CS00.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	!!
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕ —
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

**Figure 11-92. CODEPAGE1253 Column Listing
Columns 0 and 1**

CODEPAGE1253 Coded Character Set (8CS00.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-93. CODEPAGE1253 Column Listing
Columns 2 and 3**

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-94. CODEPAGE1253 Column Listing
Columns 4 and 5**

CODEPAGE1253 Coded Character Set (8CS00.00)

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-95. CODEPAGE1253 Column Listing
Columns 6 and 7**

COL/ROW	ISO Name	SYM
08/00	Euro sign	€
08/01		
08/02	Single low-9 quotation mark	,
08/03	Function sign	f
08/04	Double low-9 quotation mark	”
08/05	Horizontal ellipsis	…
08/06	Dagger	†
08/07	Double dagger	‡
08/08		
08/09	Per mille sign	‰
08/10		
08/11	Single left-pointing angle quotation mark	‹
08/12		
08/13		
08/14		
08/15		

COL/ROW	ISO Name	SYM
09/00		
09/01	Left single quotation mark	‘
09/02	Right single quotation mark	’
09/03	Left double quotation mark	“
09/04	Right double quotation mark	”
09/05	Bullet	•
09/06	En dash	—
09/07	Em dash	—
09/08		
09/09	Trademark sign	™
09/10		
09/11	Single right-pointing angle quotation mark	›
09/12		
09/13		
09/14		
09/15		

**Figure 11-96. CODEPAGE1253 Column Listing
Columns 8 and 9**

CODEPAGE1253 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Diaeresis and Greek accent	˙
10/02	Greek capital letter alpha with Greek accent	Ᾱ
10/03	Pound sign	£
10/04	Currency sign	¤
10/05	Yen sign	¥
10/06	Broken bar	⌈
10/07	Paragraph sign	§
10/08	Diaeresis	¨
10/09	Copyright sign	©
10/10	Feminine ordinal indicator	ª
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	SHY
10/14	Registered sign	®
10/15	Horizontal bar	—

COL/ROW	ISO Name	
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Superscript two	²
11/03	Superscript three	³
11/04	Greek accent	˘
11/05	Micron sign	μ
11/06	Pilcrow sign	¶
11/07	Middle dot	·
11/08	Greek capital letter epsilon with Greek accent	Ἔ
11/09	Greek capital letter eta with Greek accent	Ἠ
11/10	Greek capital letter iota with Greek accent	Ἰ
11/11	Right-pointing double angle quotation mark	»
11/12	Greek capital letter omicron with Greek accent	Ὀ
11/13	Vulgar fraction one half	½
11/14	Greek capital letter upsilon with Greek accent	ῤ
11/15	Greek capital letter omega with Greek accent	Ὠ

**Figure 11-97. CODEPAGE1253 Column Listing
Columns 10 and 11**

COL/ROW	ISO Name	SYM
12/00	Greek small letter iota with diaeresis and Greek accent	í
12/01	Greek capital letter alpha	Α
12/02	Greek capital letter beta	Β
12/03	Greek capital letter gamma	Γ
12/04	Greek capital letter delta	Δ
12/05	Greek capital letter epsilon	Ε
12/06	Greek capital letter zeta	Ζ
12/07	Greek capital letter eta	Η
12/08	Greek capital letter theta	Θ
12/09	Greek capital letter iota	Ι
12/10	Greek capital letter kappa	Κ
12/11	Greek capital letter lamda	Λ
12/12	Greek capital letter mu	Μ
12/13	Greek capital letter nu	Ν
12/14	Greek capital letter xi	Ξ
12/15	Greek capital letter omicron	Ο

COL/ROW	ISO Name	
13/00	Greek capital letter pi	Π
13/01	Greek capital letter rho	Ρ
13/02		
13/03	Greek capital letter sigma	Σ
13/04	Greek capital letter tau	Τ
13/05	Greek capital letter upsilon	Υ
13/06	Greek capital letter phi	Φ
13/07	Greek capital letter chi	Χ
13/08	Greek capital letter psi	Ψ
13/09	Greek capital letter omega	Ω
13/10	Greek capital letter iota with diaeresis	ϊ
13/11	Greek capital letter upsilon with diaeresis	ϋ
13/12	Greek small letter alpha with Greek accent	ά
13/13	Greek small letter epsilon with Greek accent	έ
13/14	Greek small letter eta with Greek accent	ή
13/15	Greek small letter iota with Greek accent	ί

**Figure 11-98. CODEPAGE1253 Column Listing
Columns 12 and 13**

CODEPAGE1253 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
14/00	Greek small letter upsilon with diaeresis and Greek accent	ύ
14/01	Greek small letter alpha	α
14/02	Greek small letter beta	β
14/03	Greek small letter gamma	γ
14/04	Greek small letter delta	δ
14/05	Greek small letter epsilon	ε
14/06	Greek small letter zeta	ζ
14/07	Greek small letter eta	η
14/08	Greek small letter theta	θ
14/09	Greek small letter iota	ι
14/10	Greek small letter kappa	κ
14/11	Greek small letter lamda	λ
14/12	Greek small letter mu	μ
14/13	Greek small letter nu	ν
14/14	Greek small letter xi	ξ
14/15	Greek small letter omicron	ο

COL/ROW	ISO Name	SYM
15/00	Greek small letter pi	π
15/01	Greek small letter rho	ρ
15/02	Greek small letter final sigma	ς
15/03	Greek small letter sigma	σ
15/04	Greek small letter tau	τ
15/05	Greek small letter upsilon	υ
15/06	Greek small letter phi	φ
15/07	Greek small letter chi	χ
15/08	Greek small letter psi	ψ
15/09	Greek small letter omega	ω
15/10	Greek small letter iota with diaeresis	ϊ
15/11	Greek small letter upsilon with diaeresis	ϋ
15/12	Greek small letter omicron with Greek accent	ό
15/13	Greek small letter upsilon with Greek accent	ύ
15/14	Greek small letter omega with Greek accent	ώ
15/15		

**Figure 11-99. CODEPAGE1253 Column Listing
Columns 14 and 15**

CODEPAGE1254 Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	€		NBSP	°	À	Ě	à	ǧ
01	☺	◀	!	1	A	Q	a	q		`	i	±	Á	Ň	á	ň
02	☺	↕	"	2	B	R	b	r	,	'	ç	²	Â	Ò	â	ò
03	♥	!!	#	3	C	S	c	s	f	"	£	³	Ã	Ó	ã	ó
04	♦	¶	\$	4	D	T	d	t	„	“	¤	´	Ä	Ô	ä	ô
05	♣	§	%	5	E	U	e	u	...	•	¥	μ	Å	Õ	å	õ
06	♠	—	&	6	F	V	f	v	†	-	¡	¶	Æ	Ö	æ	ö
07	●	↑	'	7	G	W	g	w	‡	—	§	•	Ç	x	ç	÷
08	◼	↑	(8	H	X	h	x	^	~	¨	˘	È	Ø	è	ø
09	○	↓)	9	I	Y	i	y	‰	™	©	¹	É	Ù	é	ù
10	◼	→	*	:	J	Z	j	z	Š	š	ª	º	Ê	Ú	ê	ú
11	♂	—	+	;	K	[k	{	<	>	«	»	Ë	Û	ë	û
12	♀	└	,	<	L	\	l		Œ	œ	–	¼	Ì	Ü	ì	ü
13	♆	→	-	=	M]	m	}			SHY	½	Í	İ	í	ı
14	↖	▲	.	>	N	^	n	~			®	¾	Î	Ş	î	ş
15	☼	▼	/	?	O	_	o				ÿ	–	Ï	ß	ï	ÿ

Figure 11-100. CODEPAGE1254 Code Table (8CS00.00)

CODEPAGE1254 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

COL/ROW	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	‼
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	▬
01/07	Up- and down-pointing arrow with line below	↕
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	└
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-101. CODEPAGE1254 Column Listing Columns 0 and 1

CODEPAGE1254 Coded Character Set (8CS00.00)

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-102. CODEPAGE1254 Column Listing
Columns 2 and 3**

CODEPAGE1254 Coded Character Set (8CS00.00)

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-103. CODEPAGE1254 Column Listing
Columns 4 and 5**

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	Δ

**Figure 11-104. CODEPAGE1254 Column Listing
Columns 6 and 7**

CODEPAGE1254 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
08/00	Euro sign	€
08/01		
08/02	Single low-9 quotation mark	‚
08/03	Function sign	ƒ
08/04	Double low-9 quotation mark	“
08/05	Horizontal ellipsis	…
08/06	Dagger	†
08/07	Double dagger	‡
08/08	Circumflex accent	^
08/09	Per mille sign	‰
08/10	Latin capital letter S with caron	Š
08/11	Single left-pointing angle quotation mark	‹
08/12	Latin capital letter OE	Œ
08/13		
08/14		
08/15		

COL/ROW	ISO Name	SYM
09/00		
09/01	Left single quotation mark	‘
09/02	Right single quotation mark	’
09/03	Left double quotation mark	“
09/04	Right double quotation mark	”
09/05	Bullet	•
09/06	En dash	—
09/07	Em dash	—
09/08	Small tilde	~
09/09	Trademark sign	™
09/10	Latin small letter s with caron	š
09/11	Single right-pointing angle quotation mark	›
09/12	Latin small letter oe	œ
09/13		
09/14		
09/15	Latin capital letter Y with diaeresis	ÿ

Figure 11-105. CODEPAGE1254 Column Listing Columns 8 and 9

CODEPAGE1254 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Inverted exclamation mark	¡
10/02	Cent sign	¢
10/03	Pound sign	£
10/04	Currency sign	¤
10/05	Yen sign	¥
10/06	Broken bar	¦
10/07	Paragraph sign	¶
10/08	Diaeresis	¨
10/09	Copyright sign	©
10/10	Feminine ordinal indicator	ª
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	SHY
10/14	Registered sign	®
10/15	Macron	—

COL/ROW	ISO Name	
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Superscript two	²
11/03	Superscript three	³
11/04	Acute accent	´
11/05	Micron sign	µ
11/06	Pilcrow sign	¶
11/07	Middle dot	·
11/08	Cedilla	¸
11/09	Superscript one	¹
11/10	Masculine ordinal indicator	º
11/11	Right-pointing double angle quotation mark	»
11/12	Vulgar fraction one quarter	¼
11/13	Vulgar fraction one half	½
11/14	Vulgar fraction three quarters	¾
11/15	Inverted question mark	¿

**Figure 11-106. CODEPAGE1254 Column Listing
Columns 10 and 11**

CODEPAGE1254 Coded Character Set (8CS00.00)

COL/ROW	ISO Name	SYM
12/00	Latin capital letter A with grave accent	À
12/01	Latin capital letter A with acute accent	Á
12/02	Latin capital letter A with circumflex accent	Â
12/03	Latin capital letter A with tilde	Ã
12/04	Latin capital letter A with diaeresis	Ä
12/05	Latin capital letter A with ring above	Å
12/06	Latin capital letter AE	Æ
12/07	Latin capital letter C with cedilla	Ç
12/08	Latin capital letter E with grave accent	È
12/09	Latin capital letter E with acute accent	É
12/10	Latin capital letter E with circumflex accent	Ê
12/11	Latin capital letter E with diaeresis	Ë
12/12	Latin capital letter I with grave accent	Ì
12/13	Latin capital letter I with acute accent	Í
12/14	Latin capital letter I with circumflex accent	Î
12/15	Latin capital letter I with diaeresis	Ï

COL/ROW	ISO Name	SYM
13/00	Latin capital letter G with breve accent	Ğ
13/01	Latin capital letter N with tilde	Ñ
13/02	Latin capital letter O with grave accent	Ò
13/03	Latin capital letter O with acute accent	Ó
13/04	Latin capital letter O with circumflex accent	Ô
13/05	Latin capital letter O with tilde	Õ
13/06	Latin capital letter O with diaeresis	Ö
13/07	Multiplication sign	×
13/08	Latin capital letter O with stroke	Ø
13/09	Latin capital letter U with grave accent	Ù
13/10	Latin capital letter U with acute accent	Ú
13/11	Latin capital letter U with circumflex accent	Û
13/12	Latin capital letter U with diaeresis	Ü
13/13	Latin capital letter I with dot above	İ
13/14	Latin capital letter S with cedilla	Ş
13/15	Latin small letter sharp s (German)	ß

**Figure 11-107. CODEPAGE1254 Column Listing
Columns 12 and 13**

COL/ROW	ISO Name	SYM
14/00	Latin small letter a with grave accent	à
14/01	Latin small letter a with acute accent	á
14/02	Latin small letter a with circumflex accent	â
14/03	Latin small letter a with tilde	ã
14/04	Latin small letter a with diaeresis	ä
14/05	Latin small letter a with ring above	å
14/06	Latin small letter ae	æ
14/07	Latin small letter c with cedilla	ç
14/08	Latin small letter e with grave accent	è
14/09	Latin small letter e with acute accent	é
14/10	Latin small letter e with circumflex accent	ê
14/11	Latin small letter e with diaeresis	ë
14/12	Latin small letter i with grave accent	ì
14/13	Latin small letter i with acute accent	í
14/14	Latin small letter i with circumflex accent	î
14/15	Latin small letter i with diaeresis	ï

COL/ROW	ISO Name	SYM
15/00	Latin small letter g with breve accent	ğ
15/01	Latin small letter n with tilde	ñ
15/02	Latin small letter o with grave accent	ò
15/03	Latin small letter o with acute accent	ó
15/04	Latin small letter o with circumflex accent	ô
15/05	Latin small letter o with tilde	õ
15/06	Latin small letter o with diaeresis	ö
15/07	Division sign	÷
15/08	Latin small letter o with stroke	ø
15/09	Latin small letter u with grave accent	ù
15/10	Latin small letter u with acute accent	ú
15/11	Latin small letter u with circumflex accent	û
15/12	Latin small letter u with diaeresis	ü
15/13	Latin small letter dotless i	ı
15/14	Latin small letter s with cedilla	ş
15/15	Latin small letter y with diaeresis	ÿ

**Figure 11-108. CODEPAGE1254 Column Listing
Columns 14 and 15**

CODEPAGE1256 Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	▶	SP	0	@	P	`	p	€	گ	NBSP	°	م	ن	à	´
01	☺	◀	!	1	A	Q	a	q	پ	‘	،	±	ء	ر	ل	ں
02	☺	↕	“	2	B	R	b	r	,	’	∅	²	آ	ز	â	¨
03	♥	!!	#	3	C	S	c	s	f	“	£	³	أ	س	م	´
04	♦	¶	\$	4	D	T	d	t	’	”	¤	´	ؤ	ش	ن	ô
05	♣	§	%	5	E	U	e	u	...	•	¥	µ	إ	ص	ه	³
06	♠	■	&	6	F	V	f	v	†	_		¶	ئ	ض	و	÷
07	•	↕	·	7	G	W	g	w	‡	_	§	·	ا	×	؟	˘
08	■	↑	(8	H	X	h	x	^	ک	”	،	ب	ط	è	ù
09	○	↓)	9	I	Y	i	y	%	™	©	¹	ة	ظ	é	ُ
10	⊙	→	*	:	J	Z	j	z	ٹ	ژ	ھ	؛	ت	ع	ê	
11	♂	←	+	;	K	[k	{	<	>	«	»	ث	غ	ë	û
12	♀	L	,	<	L	\			Œ	œ	¬	¼	ج	-	ی	ü
13	♪	↔	-	=	M]	m	}	چ	ZWNJ	SHY	½	ح	ف	ي	LRM
14	♪	▲	.	>	N	^	n	~	ژ	ZWJ	®	¾	خ	ق	î	RLM
15	⚙	▼	/	?	O	_	o	△	ڑ	ں	-	؟	د	ك	ï	ے

Figure 11-109. CODEPAGE1256 Code Table (8CS00.00) CODEPAGE1256 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
00/00	Null	NUL
00/01	Smiling face white	☺
00/02	Smiling face black	☹
00/03	Heart symbol	♥
00/04	Diamond symbol	♦
00/05	Club symbol	♣
00/06	Spade symbol	♠
00/07	Circle bullet solid	•
00/08	Circle bullet negative	◐
00/09	Medium circle empty	○
00/10	Medium circle negative	◑
00/11	Male sign	♂
00/12	Female sign	♀
00/13	Music note	♪
00/14	Double music notes	♫
00/15	Radiant sun	☀

Col/Row	ISO Name	SYM
01/00	Rightward pointer solid	▶
01/01	Leftward pointer solid	◀
01/02	Up- and down-pointing arrow	↕
01/03	Double exclamation marks	‼
01/04	Pilcrow sign	¶
01/05	Paragraph sign	§
01/06	Rectangle solid	■
01/07	Up- and down-pointing arrow with line below	↕ —
01/08	Upward arrow	↑
01/09	Downward arrow	↓
01/10	Rightward arrow	→
01/11	Leftward arrow	←
01/12	Right angle sign	⊥
01/13	Left- and right-pointing arrow	↔
01/14	Upward-pointing triangle solid	▲
01/15	Downward-pointing triangle solid	▼

Figure 11-110. CODEPAGE1256 Column Listing
Columns 0 and 1

CODEPAGE1256 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	SYM
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equals sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 11-111. CODEPAGE1256 Column Listing
Columns 2 and 3**

Col/Row	ISO Name	SYM
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	SYM
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 11-112. CODEPAGE1256 Column Listing
Columns 4 and 5**

CODEPAGE1256 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left square bracket	{
07/12	Reverse solidus	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 11-113. CODEPAGE1256 Column Listing
Columns 6 and 7**

Col/Row	ISO Name	SYM
08/00	Euro sign	€
08/01	Arabic letter peh	پ
08/02	Single low-9 quotation mark	،
08/03	Function sign	f
08/04	Double-low-9 quotation mark	”
08/05	Horizontal ellipsis	…
08/06	Dagger	†
08/07	Double dagger	‡
08/08	Modifier letter circumflex accent	ˆ
08/09	Per mille sign	‰
08/10	Arabic letter tteh	ٹ
08/11	Single left-pointing angle quotation mark	‹
08/12	Latin capital ligature OE	Œ
08/13	Arabic capital letter tcheh	چ
08/14	Arabic letter jeh	ج
08/15	Arabic letter ddal	ڈ

Col/Row	ISO Name	SYM
09/00	Arabic letter gaf	گ
09/01	Left single quotation mark	‘
09/02	Right single quotation mark	’
09/03	Left double quotation mark	“
09/04	Right double quotation mark	”
09/05	Bullet	•
09/06	En dash	–
09/07	Em dash	—
09/08	Arabic letter keheh	ک
09/09	Trademark sign	™
09/10	Arabic letter rreh	ر
09/11	Single right-pointing angle quotation mark	›
09/12	Latin small ligature oe	œ
09/13	Zero width non-joiner	ZWNJ
09/14	Zero width joiner	ZWJ
09/15	Arabic letter noon ghunna	ں

Figure 11-114. CODEPAGE1256 Column Listing
Columns 8 and 9

CODEPAGE1256 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Arabic comma	٫
10/02	Cent sign	¢
10/03	Pound sign	£
10/04	Currency sign	¤
10/05	Yen sign	¥
10/06	Broken bar	⌋
10/07	Section sign	§
10/08	Dieresis	¨
10/09	Copyright sign	©
10/10	Arabic letter heh doachashmee	ﻩ
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	SHY
10/14	Registered sign	®
10/15	Macron	¯

Col/Row	ISO Name	SYM
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Superscript 2	²
11/03	Superscript 3	³
11/04	Acute accent	´
11/05	Micro sign	μ
11/06	Pilcrow sign	¶
11/07	Middle dot	·
11/08	Cedilla	¸
11/09	Superscript 1	¹
11/10	Arabic semicolon	؛
11/11	Right-pointing double angle quotation mark	»
11/12	Vulgar fraction one quarter	¼
11/13	Vulgar fraction one half	½
11/14	Vulgar fraction three quarters	¾
11/15	Arabic question mark	؟

**Figure 11-115. CODEPAGE1256 Column Listing
Columns 10 and 11**

Col/Row	ISO Name	SYM
12/00	Arabic letter heh goal	هـ
12/01	Arabic letter hamza	ء
12/02	Arabic letter alef with madda above	آ
12/03	Arabic letter alef with hamza above	أ
12/04	Arabic letter waw with hamza above	ؤ
12/05	Arabic letter alef with hamza below	إ
12/06	Arabic letter yeh with hamza above	يـ
12/07	Arabic letter alef	ا
12/08	Arabic letter beh	بـ
12/09	Arabic letter teh marbuta	ة
12/10	Arabic letter teh	تـ
12/11	Arabic letter theh	ثـ
12/12	Arabic letter jeem	جـ
12/13	Arabic letter hah	حـ
12/14	Arabic letter khah	خـ
12/15	Arabic letter dal	د

Col/Row	ISO Name	SYM
13/00	Arabic letter thal	ذ
13/01	Arabic letter reh	ر
13/02	Arabic letter zain	ز
13/03	Arabic letter seen	س
13/04	Arabic letter sheen	ش
13/05	Arabic letter sad	ص
13/06	Arabic letter dad	ض
13/07	Multiplication sign	×
13/08	Arabic letter tah	ط
13/09	Arabic letter zah	ظ
13/10	Arabic letter ain	ع
13/11	Arabic letter ghain	غ
13/12	'Arabic letter tatweel	-
13/13	Arabic letter feh	ف
13/14	Arabic letter qaf	ق
13/15	Arabic letter kaf	ك

Figure 11-116. CODEPAGE1256 Column Listing
Columns 11 and 12

CODEPAGE1256 Coded Character Set (8CS00.00)

Col/Row	ISO Name	SYM
14/00	Latin small letter a with grave accent	à
14/01	Arabic letter lam	ل
14/02	Latin small letter a with circumflex accent	â
14/03	Arabic letter meem	م
14/04	Arabic letter noon	ن
14/05	Arabic letter heh	ه
14/06	Arabic letter waw	و
14/07	Latin small letter c with cedilla	ç
14/08	Latin small letter e with grave accent	è
14/09	Latin small letter e with acute accent	é
14/10	Latin small letter e circumflex accent	ê
14/11	Latin small letter e with dieresis	ë
14/12	Arabic letter alef maksura	ى
14/13	Arabic letter yeh	ي
14/14	Latin small letter i with circumflex accent	î
14/15	Latin small letter i with dieresis	ï

Col/Row	ISO Name	SYM
15/00	Arabic fathatan	ﻻ
15/01	Arabic dammatan	ﻻ
15/02	Arabic kasratan	ﻻ
15/03	Arabic fatha	ﻻ
15/04	Latin small letter o with circumflex accent	ô
15/05	Arabic damma	ﻻ
15/06	Arabic kasra	ﻻ
15/07	Division sign	÷
15/08	Arabic shadda	ّ
15/09	Latin small letter u with grave accent	ù
15/10	Arabic sukun	◌ْ
15/11	Latin small letter u with circumflex accent	û
15/12	Latin small letter u with dieresis	ü
15/13	Left-to-right mark	LRM
15/14	Right-to left mark	RLM
15/15	Arabic letter yeh baree	ﻻ

**Figure 11-117. CODEPAGE1256 Column Listing
Columns 13 and 14**

Section 12

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

This section includes coded character sets arranged alphabetically from FrenchArabic EBCDIC to NorwayBTOS. You can find additional 8-bit coded character sets in Sections 10 and 11. Double-byte and mixed, multibyte coded character sets are in Section 13. Ccsversions are in Section 14 and standard convention definitions are in Section 15.

Refer to Section 9, “Coded Character Sets and Coded Character Set Mapping Tables,” for a complete list of the coded character sets and mapping tables supported in the current release, and for how to obtain decimal and hexadecimal numbers for characters in any 8-bit table.

How to Read Coded Character Set Tables

In the coded character set tables on the following pages, read the information in the columns from left to right and the information in the rows from top to bottom. A blank box—that is, a box without any symbols or characters—means no assigned value is in that code position. A shaded box means that the position is reserved.

In the columnar listings that accompany each coded character set table, read the information as follows:

- Column 1 notes the code position by its column and row. For example, the notation 04/15 means the fourth column, fifteenth row.
- Column 2 gives the ISO name for the character. For example, *Latin a* is the lowercase letter *a*; *question mark* is the name for the symbol *?*.
- Column 3 shows the character symbol. The heading for column 3, “SYM,” stands for symbol.

FrenchArabicEBCDIC (8CS46.01) and FrenchArabicISO (8CS46.00) Coded Character Sets

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	14/02	Latin small letter a with circumflex accent	â
04/03	14/04	Latin small letter a with diaeresis	ä
04/04	14/00	Latin small letter a with grave accent	à
04/05	14/01	Latin small letter a with acute accent	á
04/06	14/03	Latin small letter a with tilde	ã
04/07	14/05	Latin small letter a with ring above	å
04/08	14/07	Latin small letter c with cedilla	ç
04/09	15/01	Latin small letter n with tilde	ñ
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	14/09	Latin small letter e with acute accent	é
05/02	14/10	Latin small letter e with circumflex accent	ê
05/03	14/11	Latin small letter e with diaeresis	ë
05/04	14/08	Latin small letter e with grave accent	è
05/05	14/13	Latin small letter i with acute accent	í
05/06	14/14	Latin small letter i with circumflex accent	î
05/07	14/15	Latin small letter i with diaeresis	ï
05/08	14/12	Latin small letter i with grave accent	ì
05/09	13/15	Latin small letter sharp s (German)	ß
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

Figure 12-1. FrenchArabicEBCDIC Code Table (8CS46.01)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	°		ذ	ـ	ر
01	SOH	DC1	!	1	A	Q	a	q		PU1	POSP	♀	°	ر	ف	س
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	PESP	è	٢	ز	ق	•
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	RESP	é	أ	س	ك	س
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	⌘	ê	ؤ	ش	ل	ف
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	NOSP	μ	!	ص	م	س
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA	NESP	ë	ئ	ض	ن	س
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA	ROSP	î	ا	ط	ه	س
08	BS	CAN	(8	H	X	h	x	HTS	SOS		ï	ب	ظ	و	س
09	HT	EM)	9	I	Y	i	y	HTJ		§	ô	ة	ع	Z	NUSP
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	±	ù	ت	غ	Z	LISP
11	VT	ESC	+	;	K	[k	{	PLD	CSI		؛	ث	»	=	
12	FF	IS4	,	<	L	\	l		PLU	ST	,	û	ج	÷	و	
13	CR	IS3	-	=	M]	m	}	RI	OSC	SHY	ü	ح	..	=	
14	LSO	IS2	.	>	N	^	n	~	SS2	PM	à	آ	خ		ا	
15	LSI	IS1	/	?	O	_	o	DEL	SS3	APC	â	؟	د		,	

Figure 12-2. FrenchArabicISO Code Table (8CS46.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Locking-shift 0	LS0
00/15	00/15	Locking-shift 1	LS1

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

**Figure 12-3. FrenchArabicEBCDIC and FrenchArabicISO Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledgement	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substituted character	SUB

**Figure 12-4. FrenchArabicEBCDIC and FrenchArabicISO Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	10/15	Latin small letter a with circumflex accent	â
04/03			
04/04	10/14	Latin small letter a with grave accent	à
04/05			
04/06			
04/07			
04/08	11/01	Latin small letter c with cedilla	ç
04/09			
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	11/02	Latin small letter e with acute accent	é
05/02	11/04	Latin small letter e with circumflex accent	ê
05/03	11/06	Latin small letter e with diaeresis	ë
05/04	11/03	Latin small letter e with grave accent	è
05/05			
05/06	11/07	Latin small letter i with circumflex accent	î
05/07	11/08	Latin small letter i with diaeresis	ï
05/08			
05/09			
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

Figure 12-5. FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 4 and 5

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	15/09	Numeric space	NUSP
06/03	15/10	Linking space	LISP
06/04	10/01	Positive orientation space	POSP
06/05	10/02	Positive escapement space	PESP
06/06	10/03	Reset escapement space	RESP
06/07	10/05	Negative orientation space	NOSP
06/08	10/06	Negative escapement space	NESP
06/09	10/07	Reset orientation space	ROSP
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	10/12	Arabic comma	،
07/01	11/11	Arabic semicolon	؛
07/02	11/15	Arabic question mark	؟
07/03	15/03	Arabic letter shadda fathatan	ﻪ
07/04	15/04	Arabic letter shadda dammatan	ﻪ
07/05	15/05	Arabic letter shadda kasratan	ﻪ
07/06	15/06	Arabic letter shadda fatha	ﻪ
07/07	15/07	Arabic letter shadda damma	ﻪ
07/08	15/08	Arabic letter shadda kasra	ﻪ
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

Figure 12-6. FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 6 and 7

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00			
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10		Left-pointing double angle quotation mark	«
08/11	13/11	Right-pointing double angle quotation mark	»
08/12			
08/13	12/02	Box drawing light vertical and right	⌈
08/14	12/03	Box drawing light vertical and left	⌋
08/15	10/10	Plus-minus sign	±

EBCDIC	ISO	ISO Name	SYM
09/00	11/00	Degree sign	°
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	12/04	Arabic letter waw with hamza above	ﻭ̣
09/11	12/05	Arabic letter alef with hamza below	ﺀ
09/12	12/06	Arabic letter yeh with hamza above	ﻱ̣
09/13	12/07	Arabic letter alef	ﺀ
09/14	12/08	Arabic letter beh	ﺏ
09/15	10/04	Currency sign	₮

Figure 12-7. FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 8 and 9

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	11/05	Micro sign	μ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	12/09	Arabic letter teh marbuta	ة
10/11	12/10	Arabic letter teh	ت
10/12	12/11	Arabic letter theh	ث
10/13	12/12	Arabic letter jeem	ج
10/14	12/13	Arabic letter hah	ح
10/15	12/14	Arabic letter khah	خ

EBCDIC	ISO	ISO Name	SYM
11/00	12/15	Arabic letter dal	د
11/01			
11/02	13/00	Arabic letter thal	ذ
11/03	13/01	Arabic letter ra	ر
11/04	13/02	Arabic letter zain	ز
11/05	10/09	Paragraph sign	¶
11/06	11/14	Pilcrow sign	¶
11/07	13/03	Arabic letter seen	س
11/08	13/04	Arabic letter sheen	ش
11/09	13/05	Arabic letter sad	ص
11/10	13/06	Arabic letter dad	ض
11/11	13/07	Arabic letter tah	ط
11/12	13/08	Arabic letter zah	ظ
11/13	13/09	Arabic letter ain	ع
11/14	13/10	Arabic letter ghain	غ
	13/12	Division sign	÷
	13/13	Diaeresis	¨
11/15	14/00	Arabic letter tatweel	ا

Figure 12-8. FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 10 and 11

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hyphen	SHY
12/11	11/09	Latin small letter o with circumflex accent	ô
12/12	14/01	Arabic letter feh	ف
12/13	14/02	Arabic letter gaf	ق
12/14	14/03	Arabic letter kaf	ك
12/15	14/04	Arabic letter lam	ل

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	14/05	Arabic letter meem	م
13/11	11/12	Latin small letter u with circumflex accent	û
13/12	11/13	Latin small letter u with diaeresis	ü
13/13	11/10	Latin small letter u with grave accent	ù
13/14	14/06	Arabic letter noon	ن
13/15	14/07	Arabic letter heh	ه

Figure 12-9. FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 12 and 13

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01			
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	14/08	Arabic letter waw	و
14/11	14/09	Arabic letter alef maksura	ى
14/12	14/10	Arabic letter yeh	ي
14/13	14/11	Arabic letter fathatan	=
14/14	14/12	Arabic letter dammatan	◌ُ
14/15	14/13	Arabic letter kasratan	=

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	14/14	Arabic letter fatha	َ
15/11	14/15	Arabic letter damma	◌ُ
15/12	15/00	Arabic letter kasra	َ
15/13	15/01	Arabic letter shadda	◌ّ
15/14	15/02	Arabic letter sukun	◌ْ
15/15	07/15	Delete	DEL

Figure 12-10. FrenchArabicEBCDIC and FrenchArabicISO Column Listing Columns 14 and 15

HungarianBTOS Coded Character Sets (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00		¼	SP	0	@	P	`	p		4	Á	ü	†	+	⌞	⌏
01	↑	†	!	1	A	Q	a	q	l	5	É	ú	‡			
02	▲	→	"	2	B	R	b	r	l0	6	Í	Ä	‡		†	⌞
03	♣	⌎	#	3	C	S	c	s	l1	7	Ó	ä	‡		†	
04	■	▪	\$	4	D	T	d	t	l2	8	Ö	§	†		⌞	
05	□	÷	%	5	E	U	e	u	l3	9	Ő	™	‡		†	†
06	½		&	6	F	V	f	v	l4	0	Ú	l			†	†
07	☺	§	'	7	G	W	g	w	l5	1	Ü	l1	‡		⌞	†
08	†	≠	(8	H	X	h	x	l6	2	Ű	l2	‡	=	⌞	†
09	→	=)	9	I	Y	i	y	l7	3	á	l3	†		⌞	†
10	↙	¶	*	:	J	Z	j	z	l8	4	é	l4	†	—	⌞	†
11	↓	•	+	;	K	[k	{	l9	5	í	l5	‡	†	⌞	
12	↘	⌏	,	<	L	\	l	!	0	6	ó	⌘	†	†	⌞	
13	▲	≤	-	=	M]	m	}	1	7	ö	l7	†	†	⌞	▨
14	←	±	.	>	N	^	n	~	2	8	ő	x	—		⌞	▨
15	‡	≥	/	?	O	_	o	■	3	9	ú	÷	‡	†	⌞	▨

Figure 12-11. HungarianBTOS Code Table (8CS00.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00		
00/01	Upward arrow	↑
00/02	Lower left corner triangle solid	▴
00/03	Cent sign	¢
00/04	Square solid	■
00/05	Square empty	□
00/06	Vulgar fraction one half	½
00/07	Bell sign	🔔
00/08	Left tab sign	⌠
00/09	Right tab sign	⌡
00/10	Left return sign	↶
00/11	Downward arrow	↓
00/12	Right return sign	↷
00/13	Lower right corner triangle solid	▾
00/14	Leftward arrow	←
00/15	Double dagger	‡

COL/ROW	ISO Name	
01/00	Vulgar fraction one quarter	¼
01/01	Dagger	†
01/02	Rightward arrow	→
01/03	Space symbol	␣
01/04	Box drawing solid small square	◼
01/05	Division sign	÷
01/06	Vertical line	
01/07	Paragraph sign	§
01/08	Not equal-to sign	≠
01/09	Asymptotically equal sign	≈
01/10	Pilcrow sign	¶
01/11	Middle dot	•
01/12	Not sign	¬
01/13	Less-than or equal-to sign	≤
01/14	Plus-minus sign	±
01/15	Greater-than or equal-to sign	≥

**Figure 12-12. HungarianBTOS Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
02/00	Space	
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

COL/ROW	ISO Name	SYM
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 12-13. HungarianBTOS Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

COL/ROW	ISO Name	SYM
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	-

**Figure 12-14. HungarianBTOS Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

COL/ROW	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Broken bar	¡
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Box drawing solid half block	◻

Figure 12-15. HungarianBTOS Column Listing Columns 6 and 7

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00		
08/01	Vertical line	
08/02	Vertical line with subscript 0	₀
08/03	Vertical line with subscript 1	₁
08/04	Vertical line with subscript 2	₂
08/05	Vertical line with subscript 3	₃
08/06	Vertical line with subscript 4	₄
08/07	Vertical line with subscript 5	₅
08/08	Vertical line with subscript 6	₆
08/09	Vertical line with subscript 7	₇
08/10	Vertical line with subscript 8	₈
08/11	Vertical line with subscript 9	₉
08/12	Superscript 0	⁰
08/13	Superscript 1	¹
08/14	Superscript 2	²
08/15	Superscript 3	³

COL/ROW	ISO Name	
09/00	Superscript 4	⁴
09/01	Superscript 5	⁵
09/02	Superscript 6	⁶
09/03	Superscript 7	⁷
09/04	Superscript 8	⁸
09/05	Superscript 9	⁹
09/06	Subscript 0	₀
09/07	Subscript 1	₁
09/08	Subscript 2	₂
09/09	Subscript 3	₃
09/10	Subscript 4	₄
09/11	Subscript 5	₅
09/12	Subscript 6	₆
09/13	Subscript 7	₇
09/14	Subscript 8	₈
09/15	Subscript 9	₉

**Figure 12-16. HungarianBTOS Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	Latin capital letter A with acute accent	Á
10/01	Latin capital letter E with acute accent	É
10/02	Latin capital letter I with acute accent	Í
10/03	Latin capital letter O with acute accent	Ó
10/04	Latin capital letter O with diaeresis	Ö
10/05	Latin capital letter O with double acute accent	Ő
10/06	Latin capital letter U with acute accent	Ú
10/07	Latin capital letter U with diaeresis	Ü
10/08	Latin capital letter U with double acute accent	Û
10/09	Latin small letter a with acute accent	á
10/10	Latin small letter e with acute accent	é
10/11	Latin small letter i with acute accent	í
10/12	Latin small letter o with acute accent	ó
10/13	Latin small letter o with diaeresis	ö
10/14	Latin small letter o with double acute accent	ő
10/15	Latin small letter u with acute accent	ú

COL/ROW	ISO Name	SYM
11/00	Latin small letter u with diaeresis	ü
11/01	Latin small letter u with double acute accent	ű
11/02	Latin capital letter A with diaeresis	Ä
11/03	Latin small letter a with diaeresis	ä
11/04	Paragraph sign	§
11/05	Trade mark sign	™
11/06	Short vertical line	l
11/07	Short vertical line with subscript 1	l ₁
11/08	Short vertical line with subscript 2	l ₂
11/09	Short vertical line with subscript 3	l ₃
11/10	Short vertical line with subscript 4	l ₄
11/11	Short vertical line with subscript 5	l ₅
11/12	Currency sign	¤
11/13	Short vertical line with subscript 7	l ₇
11/14	Multiplication sign	×
11/15	Division sign	÷

**Figure 12-17. HungarianBTOS Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Box drawing heavy vertical and light right	┃
12/01	Box drawing heavy vertical and light left	┃
12/02	Box drawing heavy vertical and light horizontal	┃┃
12/03	Box drawing heavy vertical and horizontal	┃┃
12/04	Box drawing heavy vertical and right	┃┃
12/05	Box drawing heavy vertical and left	┃┃
12/06	Box drawing heavy vertical	┃
12/07	Box drawing heavy vertical and left and light right	┃┃
12/08	Box drawing heavy vertical and right and light left	┃┃
12/09	Box drawing heavy horizontal and light down	┃
12/10	Box drawing light horizontal and heavy up	┃
12/11	Box drawing heavy horizontal and light vertical	┃┃
12/12	Box drawing heavy horizontal and down	┃
12/13	Box drawing heavy horizontal and up	┃
12/14	Box drawing heavy horizontal	┃
12/15	Box drawing heavy horizontal and up and light down	┃┃

COL/ROW	ISO Name	
13/00	Box drawing heavy horizontal and down and light up	┃┃
13/01	Box drawing double vertical and light right	┃┃
13/02	Box drawing double vertical and light left	┃┃
13/03	Box drawing double vertical	┃┃
13/04	Box drawing double vertical and light horizontal	┃┃
13/05	Box drawing double vertical and horizontal	┃┃
13/06	Box drawing double horizontal and light down	┃┃
13/07	Box drawing double horizontal and light up	┃┃
13/08	Box drawing double horizontal	┃┃
13/09	Box drawing double horizontal and light vertical	┃┃
13/10	Box drawing light horizontal	┃
13/11	Box drawing light horizontal and vertical	┃┃
13/12	Box drawing light horizontal and down	┃
13/13	Box drawing light horizontal and up	┃
13/14	Box drawing light horizontal and double up	┃┃
13/15	Box drawing light horizontal and double down	┃┃

**Figure 12-18. HungarianBTOS Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Box drawing double up and right	┐
14/01	Box drawing light vertical	
14/02	Box drawing light vertical and right	┌
14/03	Box drawing light vertical and left	└
14/04	Box drawing double down and right	┘
14/05	Box drawing light vertical and double right	┆
14/06	Box drawing light vertical and double left	┆
14/07	Box drawing double up and left	┐
14/08	Box drawing double down and left	┘
14/09	Box drawing heavy up and right	┐
14/10	Box drawing heavy up and left	┐
14/11	Box drawing heavy down and right	┘
14/12	Box drawing heavy down and left	┘
14/13	Box drawing light up and right	┐
14/14	Box drawing light up and left	┐
14/15	Box drawing light down and right	┘

COL/ROW	ISO Name	SYM
15/00	Box drawing light down and left	└
15/01	Box drawing double vertical and right	┆
15/02	Box drawing double horizontal and down	┆
15/03	Box drawing double vertical and left	┆
15/04	Box drawing double horizontal and up	┆
15/05	Box drawing heavy vertical and double right	┆
15/06	Box drawing heavy horizontal and double down	┆
15/07	Box drawing heavy vertical and double left	┆
15/08	Box drawing heavy horizontal and double up	┆
15/09	Box drawing double vertical and heavy right	┆
15/10	Box drawing double horizontal and heavy down	┆
15/11	Box drawing double vertical and heavy left	┆
15/12	Box drawing double horizontal and heavy up	┆
15/13	Bar chart	▒
15/14	Bar chart	▒
15/15	Bar chart	▒

Figure 12-19. HungarianBTOS Column Listing Columns 14 and 15

IBM297 Coded Character Set (8CS62.01)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ø	Ø	l	`	ç	é	è	ç	0
01	SOH	DC1		PU1	NBSP	{	/	É	a	j	¨	#	A	J	÷	1
02	STX	DC2	BPH	SYN	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03	ETX	DC3	NBH	STS	ä	ë	Ä	Ë	c	l	t	•	C	L	T	3
04	ST	OSC	IND	CCH	@	}	À	È	d	m	u	©	D	M	U	4
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v]	E	N	V	5
06	SSA	BS	ETB	SPA	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07	APC	ESA	ESC	EOT	å	ï	Å	Ï	g	p	x	¼	G	P	X	7
08	EPA	CAN	HTS	SOS	\	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
09	RI	EM	HTJ		ñ	ß	Ñ	µ	i	r	z	¾	I	R	Z	9
10	SS2	PU2	VTS	SCI	°	§	ù	:	«	ª	ı	–	SHY	¹	²	³
11	VT	SS3	PLD	CSI	.	\$,	£	»	º	ı	l	ô	û	Ô	Û
12	FF	IS4	PLU	DC4	<	*	%	à	ð	æ	Ð	-	ö	ü	Ö	Ü
13	CR	IS3	ENQ	NAK	()	_	'	ý	.	Ý	~	ò	ı	Ò	Ù
14	SO	IS2	ACK	PM	+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
15	SI	IS1	BEL	SUB	!	^	?	"	±	⊘	®	x	õ	ÿ	Õ	DEL

Figure 12-20. IBM297 Code Table (8CS62.01)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Start of heading	SOH
00/02	Start of text	STX
00/03	End of text	ETX
00/04	String terminator	ST
00/05	Horizontal tabulation	HT
00/06	Start of selected area	SSA
00/07	Application program command	APC
00/08	End of guarded protected area	EPA
00/09	Reverse index	RI
00/10	Single shift 2	SS2
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

COL/ROW	ISO Name	SYM
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04	Operation system command	OSC
01/05	Next line	NEL
01/06	Backspace	BS
01/07	End of selected area	ESA
01/08	Cancel	CAN
01/09	End medium	EM
01/10	Private use 2	PU2
01/11	Single shift 3	SS3
01/12	Information separator 4	IS4
01/13	Information separator 3	IS3
01/14	Information separator 2	IS2
01/15	Information separator 1	IS1

**Figure 12-21. IBM297 Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
02/00		
02/01		
02/02	Break permitted here	BPH
02/03	No break here	NBH
02/04	Index	IND
02/05	Line feed	LF
02/06	End-of-transmission block	ETB
02/07	Escape	ESC
02/08	Horizontal tabulation set	HTS
02/09	Horizontal tabulation with justification	HTJ
02/10	Vertical tabulation set	VTB
02/11	Partial line down	PLD
02/12	Partial line up	PLU
02/13	Enquiry	ENQ
02/14	Acknowledgement	ACK
02/15	Bell	BEL

COL/ROW	ISO Name	SYM
03/00	Device control string	DCS
03/01	Private use 1	PU1
03/02	Synchronous idle	SYN
03/03	Set transmission state	STS
03/04	Cancel character	CCH
03/05	Message waiting	MW
03/06	Start of guarded protected area	SPA
03/07	End of transmission	EOT
03/08	Start of string	SOS
03/09		
03/10	Single character introducer	SCI
03/11	Control sequence introducer	CSI
03/12	Device control 4	DC4
03/13	Negative acknowledge	NAK
03/14	Privacy message	PM
03/15	Substituted character	SUB

**Figure 12-22. IBM297 Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
04/00	Space	SP
04/01	No-break space	NBSP
04/02	Latin small letter a with circumflex accent	â
04/03	Latin small letter a with diaeresis	ä
04/04	Commercial at	@
04/05	Latin small letter a with acute accent	á
04/06	Latin small letter a with tilde	ã
04/07	Latin small letter a with ring above	å
04/08	Reverse solidus	\
04/09	Latin small letter n with tilde	ñ
04/10	Degree sign	°
04/11	Full stop	.
04/12	Less-than sign	<
04/13	Left parenthesis	(
04/14	Plus sign	+
04/15	Exclamation mark	!

COL/ROW	ISO Name	SYM
05/00	Ampersand	&
05/01	Left curly bracket	{
05/02	Latin small letter e with circumflex accent	ê
05/03	Latin small letter e with diaeresis	ë
05/04	Right curly bracket	}
05/05	Latin small letter i with acute accent	í
05/06	Latin small letter i with circumflex accent	î
05/07	Latin small letter i with diaeresis	ï
05/08	Latin small letter i with grave accent	ì
05/09	Latin small letter sharp s (German)	ß
05/10	Paragraph sign	§
05/11	Dollar sign	\$
05/12	Asterisk	*
05/13	Right parenthesis)
05/14	Semicolon	;
05/15	Circumflex accent	^

**Figure 12-23. IBM297 Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
06/00	Hyphen-minus	-
06/01	Solidus	/
06/02	Latin capital letter A with circumflex accent	Â
06/03	Latin capital letter A with diaeresis	Ä
06/04	Latin capital letter A with grave accent	À
06/05	Latin capital letter A with acute accent	Á
06/06	Latin capital letter A with tilde	Ã
06/07	Latin capital letter A with ring above	Å
06/08	Latin capital letter C with cedilla	Ç
06/09	Latin capital letter N with tilde	Ñ
06/10	Latin small letter u with grave accent	ù
06/11	Comma	,
06/12	Percent sign	%
06/13	Low line	_
06/14	Greater-than sign	>
06/15	Question mark	?

COL/ROW	ISO Name	SYM
07/00	Latin small letter o with oblique stroke	ø
07/01	Latin capital letter E with acute accent	É
07/02	Latin capital letter E with circumflex accent	Ê
07/03	Latin capital letter E with diaeresis	Ë
07/04	Latin capital letter E with grave accent	È
07/05	Latin capital letter I with acute accent	Í
07/06	Latin capital letter I with circumflex accent	Î
07/07	Latin capital letter I with diaeresis	Ï
07/08	Latin capital letter I with grave accent	Ì
07/09	Micro sign	μ
07/10	Colon	:
07/11	Pound sign	£
07/12	Latin small letter a with grave accent	à
07/13	Apostrophe	'
07/14	Equal sign	=
07/15	Quotation mark	"

**Figure 12-24. IBM297 Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00	Latin capital letter O with oblique stroke	Ø
08/01	Latin small letter a	a
08/02	Latin small letter b	b
08/03	Latin small letter c	c
08/04	Latin small letter d	d
08/05	Latin small letter e	e
08/06	Latin small letter f	f
08/07	Latin small letter g	g
08/08	Latin small letter h	h
08/09	Latin small letter i	i
08/10	Left-pointing double angle quotation mark	«
08/11	Right-pointing double angle quotation mark	»
08/12	Latin small letter eth (Icelandic)	ð
08/13	Latin small letter y with acute accent	ý
08/14	Latin small letter thorn (Icelandic)	þ
08/15	Plus-minus sign	±

COL/ROW	ISO Name	SYM
09/00	Left square bracket	[
09/01	Latin small letter j	j
09/02	Latin small letter k	k
09/03	Latin small letter l	l
09/04	Latin small letter m	m
09/05	Latin small letter n	n
09/06	Latin small letter o	o
09/07	Latin small letter p	p
09/08	Latin small letter q	q
09/09	Latin small letter r	r
09/10	Feminine ordinal indicator	ª
09/11	Masculine ordinal indicator	º
09/12	Latin small letter a with e	æ
09/13	Cedilla	¸
09/14	Latin capital letter A with E	Æ
09/15	Currency sign	₰

**Figure 12-25. IBM297 Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	Grave accent	`
10/01	Diaeresis	¨
10/02	Latin small letter s	s
10/03	Latin small letter t	t
10/04	Latin small letter u	u
10/05	Latin small letter v	v
10/06	Latin small letter w	w
10/07	Latin small letter x	x
10/08	Latin small letter y	y
10/09	Latin small letter z	z
10/10	Inverted exclamation mark	¡
10/11	Inverted question mark	¿
10/12	Latin capital letter eth (Icelandic)	Ð
10/13	Latin capital letter Y with acute accent	Ý
10/14	Latin capital letter thorn (Icelandic)	Þ
10/15	Registered trademark sign	®

COL/ROW	ISO Name	SYM
11/00	Cent sign	¢
11/01	Number sign	#
11/02	Yen sign	¥
11/03	Middle dot	•
11/04	Copyright sign	©
11/05	Right square bracket]
11/06	Pilcrow sign	¶
11/07	Vulgar fraction one quarter	¼
11/08	Vulgar fraction one half	½
11/09	Vulgar fraction three quarters	¾
11/10	Not sign	¬
11/11	Vertical line	
11/12	Macron	-
11/13	Tilde	~
11/14	Acute accent	´
11/15	Multiplication sign	×

**Figure 12-26. IBM297 Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Latin small letter e with acute accent	é
12/01	Latin capital letter A	A
12/02	Latin capital letter B	B
12/03	Latin capital letter C	C
12/04	Latin capital letter D	D
12/05	Latin capital letter E	E
12/06	Latin capital letter F	F
12/07	Latin capital letter G	G
12/08	Latin capital letter H	H
12/09	Latin capital letter I	I
12/10	Soft hyphen	SHY
12/11	Latin small letter o with circumflex accent	ô
12/12	Latin small letter o with diaeresis	ö
12/13	Latin small letter o with grave accent	ò
12/14	Latin small letter o with acute accent	ó
12/15	Latin small letter o with tilde	õ

COL/ROW	ISO Name	SYM
13/00	Latin small letter e with grave accent	è
13/01	Latin capital letter J	J
13/02	Latin capital letter K	K
13/03	Latin capital letter L	L
13/04	Latin capital letter M	M
13/05	Latin capital letter N	N
13/06	Latin capital letter O	O
13/07	Latin capital letter P	P
13/08	Latin capital letter Q	Q
13/09	Latin capital letter R	R
13/10	Superscript 1	¹
13/11	Latin small letter u with circumflex accent	û
13/12	Latin small letter u with diaeresis	ü
13/13	Broken bar	¡
13/14	Latin small letter u with acute accent	ú
13/15	Latin small letter y with diaeresis	ÿ

**Figure 12-27. IBM297 Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Latin small letter c with cedilla	ç
14/01	Division sign	÷
14/02	Latin capital letter S	S
14/03	Latin capital letter T	T
14/04	Latin capital letter U	U
14/05	Latin capital letter V	V
14/06	Latin capital letter W	W
14/07	Latin capital letter X	X
14/08	Latin capital letter Y	Y
14/09	Latin capital letter Z	Z
14/10	Superscript 2	²
14/11	Latin capital letter O with circumflex accent	Ô
14/12	Latin capital letter O with diaeresis	Ö
14/13	Latin capital letter O with grave accent	Ò
14/14	Latin capital letter O with acute accent	Ó
14/15	Latin capital letter O with tilde	Õ

COL/ROW	ISO Name	SYM
15/00	Digit 0	0
15/01	Digit 1	1
15/02	Digit 2	2
15/03	Digit 3	3
15/04	Digit 4	4
15/05	Digit 5	5
15/06	Digit 6	6
15/07	Digit 7	7
15/08	Digit 8	8
15/09	Digit 9	9
15/10	Superscript 3	³
15/11	Latin capital letter U with circumflex accent	Û
15/12	Latin capital letter U with diaeresis	Ü
15/13	Latin capital letter U with grave accent	Ù
15/14	Latin capital letter U with acute accent	Ú
15/15	Delete	DEL

**Figure 12-28. IBM297 Column Listing
Columns 14 and 15**

IBMSwedenEBCDIC Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	PAD	DCS	SP	&	a	j	-		ü		ä	å	É	0
01	SOH	DC1	HOP	PU1			b	k	/		s		A	J		1
02	STX	DC2	BPH	SYN			c	l			t		B	K	S	2
03	ETX	DC3	NBH	STS			d	m			u		C	L	T	3
04	ST	OSC	IND	CCH			e	n			v		D	M	U	4
05	HT	NEL	LF	MW			f	o			w		E	N	V	5
06	SSA	BS	ETB	SPA			g	p			x		F	O	W	6
07	DEL	ESA	ESC	EOT			h	q			y		G	P	X	7
08	EPA	CAN	HTS	SOS			i	r			z		H	Q	Y	8
09	RI	EM	HTJ	SGCI						é			I	R	Z	9
10	SS2	PU2	VTS	SCI	§	¤			ö	:						
11	VT	SS3	PLD	CSI	.	Å			,	Â						
12	FF	IS4	PLU	DC4	<	*			%	Ö						
13	CR	IS3	ENQ	NAK	()			-	‘						
14	SO	IS2	ACK	PM	+	;			>	+						
15	SI	IS1	BEL	SUB	!	^			?	“						APC

Figure 12-29. IBMSwedenEBCDIC Code Table

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
00/00	Null	NUL
00/01	Start of heading	SOH
00/02	Start of text	STX
00/03	End of text	ETX
00/04	String terminator	ST
00/05	Horizontal tabulation	HT
00/06	Start of selected area	SSA
00/07	Delete	DEL
00/08	End of guarded protected area	EPA
00/09	Reverse index	TI
00/10	Single shift 2	SS2
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

Col/Row	ISO Name	SYM
01/00	Data link escape	SYM
01/01	Device control 1	DLE
01/02	Device control 2	DC1
01/03	Device control 3	DC2
01/04	Operation system command	DC3
01/05	Next line	OSC
01/06	Backspace	NEL
01/07	End of selected area	BS
01/08	Cancel	ESA
01/09	End medium	CAN
01/10	Private use 2	EM
01/11	Single shift 3	PU2
01/12	Information separator 4	SS3
01/13	Information separator 3	IS4
01/14	Information separator 2	IS3
01/15	Information separator 1	IS2
		IS1

**Figure 12-30. IBMSwedenEBCDIC Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
02/00	Padding character	PAD
02/01	High octet preset	HOP
02/02	Break permitted here	BPH
02/03	No break here	NBH
02/04	Index	IND
02/05	Line feed	LF
02/06	End-of-transmission block	ETB
02/07	Escape	ESC
02/08	Horizontal tabulation set	HTS
02/09	Horizontal tabulation with justification	HTJ
02/10	Vertical tabulation set	VTB
02/11	Partial line down	PLD
02/12	Partial line up	PLU
02/13	Enquiry	ENQ
02/14	Acknowledgment	ACK
02/15	Bell	BEL

Col/Row	ISO Name	SYM
03/00	Device control string	DCS
03/01	Private use 1	PU1
03/02	Synchronous idle	SYN
03/03	Set transmission state	STS
03/04	Cancel character	CCH
03/05	Message waiting	MW
03/06	Start of guarded protection area	SPA
03/07	End of transmission	EOT
03/08	Start of string	SOS
03/09	Single graphic character introducer	SGCI
03/10	Single character introducer	SCI
03/11	Control sequence introducer	CSI
03/12	Device control 4	DC4
03/13	Negative acknowledge	NAK
03/14	Privacy message	PM
03/15	Substituted character	SUB

**Figure 12-31. IBMSwedenEBCDIC Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
04/00	Space	SP
04/01		
04/02		
04/03		
04/04		
04/05		
04/06		
04/07		
04/08		
04/09		
04/10	Section sign	§
04/11	Full stop	.
04/12	Less-than sign	<
04/13	Left parenthesis	(
04/14	Plus sign	+
04/15	Exclamation mark	!

Col/Row	ISO Name	SYM
05/00	Ampersand	&
05/01		
05/02		
05/03		
05/04		
05/05		
05/06		
05/07		
05/08		
05/09		
05/10	Currency sign	¤
05/11	Latin capital letter A with ring above	Å
05/12	Asterisk	*
05/13	Right parenthesis)
05/14	Semicolon	;
05/15	Circumflex accent	^

**Figure 12-32. IBMSwedenEBCDIC Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
06/00	Hyphen-minus	-
06/01	Solidus	/
06/02		
06/03		
06/04		
06/05		
06/06		
06/07		
06/08		
06/09		
06/10	Latin small letter o with dieresis	ö
06/11	Comma	,
06/12	Percent sign	%
06/13	Low line	_
06/14	Greater-than-sign	>
06/15	Question mark	?

Col/Row	ISO Name	SYM
07/00		
07/01		
07/02		
07/03		
07/04		
07/05		
07/06		
07/07		
07/08		
07/09	Latin small letter e with acute accent	é
07/10	Colon	:
07/11	Latin capital letter A with dieresis	Ä
07/12	Latin capital letter O with dieresis	Ö
07/13	Apostrophe	'
07/14	Equal sign	=
07/15	Quotation mark	"

**Figure 12-33. IBMSwedenEBCDIC Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
08/00	Latin small letter a	a
08/01	Latin small letter b	b
08/02	Latin small letter c	c
08/03	Latin small letter d	d
08/04	Latin small letter e	e
08/05	Latin small letter f	f
08/06	Latin small letter g	g
08/07	Latin small letter h	h
08/08	Latin small letter i	i
08/09		
08/10		
08/11		
08/12		
08/13		
08/14		
08/15		

Col/Row	ISO Name	SYM
09/00		
09/01	Latin small letter j	j
09/02	Latin small letter k	k
09/03	Latin small letter l	l
09/04	Latin small letter m	m
09/05	Latin small letter n	n
09/06	Latin small letter o	o
09/07	Latin small letter p	p
09/08	Latin small letter q	q
09/09	Latin small letter r	r
09/10		
09/11		
09/12		
09/13		
09/14		
09/15		

**Figure 12-34. IBMSwedenEBCDIC Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
10/00	Latin small letter u with dieresis	ü
10/01	Latin small letter s	s
10/02	Latin small letter t	t
10/03	Latin small letter u	u
10/04	Latin small letter v	v
10/05	Latin small letter w	w
10/06	Latin small letter x	x
10/07	Latin small letter y	y
10/08	Latin small letter z	z
10/09		
10/10		
10/11		
10/12		
10/13		
10/14		
10/15		

Col/Row	ISO Name	SYM
11/00		
11/01		
11/02		
11/03		
11/04		
11/05		
11/06		
11/07		
11/08		
11/09		
11/10		
11/11		
11/12		
11/13		
11/14		
11/15		

**Figure 12-35. IBMSwedenEBCDIC Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
12/00	Latin small letter a with dieresis	ä
12/01	Latin capital letter A	A
12/02	Latin capital letter B	B
12/03	Latin capital letter C	C
12/04	Latin capital letter D	D
12/05	Latin capital letter E	E
12/06	Latin capital letter F	F
12/07	Latin capital letter G	G
12/08	Latin capital letter H	H
12/09	Latin capital letter I	I
12/10		
12/11		
12/12		
12/13		
12/14		
12/15		

Col/Row	ISO Name	SYM
13/00	Latin small letter a with ring above	å
13/01	Latin capital letter J	J
13/02	Latin capital letter K	K
13/03	Latin capital letter L	L
13/04	Latin capital letter M	M
13/05	Latin capital letter N	N
13/06	Latin capital letter O	O
13/07	Latin capital letter P	P
13/08	Latin capital letter Q	Q
13/09	Latin capital letter R	R
13/10		
13/11		
13/12		
13/13		
13/14		
13/15		

**Figure 12-36. IBMSwedenEBCDIC Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
14/00	Latin capital letter E with acute accent	É
14/01		
14/02	Latin capital letter S	S
14/03	Latin capital letter T	T
14/04	Latin capital letter U	U
14/05	Latin capital letter V	V
14/06	Latin capital letter W	W
14/07	Latin capital letter X	X
14/08	Latin capital letter Y	Y
14/09	Latin capital letter Z	Z
14/10		
14/11		
14/12		
14/13		
14/14		
14/15		

Col/Row	ISO Name	SYM
15/00	Digit 0	0
15/01	Digit 1	1
15/02	Digit 2	2
15/03	Digit 3	3
15/04	Digit 4	4
15/05	Digit 5	5
15/06	Digit 6	6
15/07	Digit 7	7
15/08	Digit 8	8
15/09	Digit 9	9
15/10		
15/11		
15/12		
15/13		
15/14		
15/15	Application program command	APC

**Figure 12-37. IBMSwedenEBCDIC Column Listing
Columns 14 and 15**

JapanEBCDIC Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE			SP	&	-	コ	テ	ハ	ム	リ	{	}	\	0
01	SOH	DC1			。	エ	/	サ	a	j	~	ル	A	J	APC	1
02	STX	DC2		SYN	「	オ	イ	シ	b	k	s		B	K	S	2
03	ETX	DC3			」	ヤ	ウ	ス	c	l	t		C	L	T	3
04					、	ユ	エ	セ	d	m	u		D	M	U	4
05	HT		LF		・	ヨ	オ	ソ	e	n	v		E	N	V	5
06		BS	ETB		ヲ	ツ	カ	タ	f	o	w		F	O	W	6
07	DEL		ESC	EOT	ア	ー	キ	チ	g	p	x		G	P	X	7
08		CAN			イ	ア	ク	ツ	h	q	y		H	Q	Y	8
09		EM			ウ		ケ	`	i	r	z		I	R	Z	9
10					[]		:	ト	ヒ	メ	レ				
11	VT		SDO		.	\$,	#	ナ	フ	モ	ロ				
12	FF	FS	EDO	DC4	<	*	%	@	ニ	ハ	ヤ	ワ				
13	CR	GS	ENQ	NAK	()	_	'	ヌ	ホ	ユ	ン				
14	SO	RS	ACK		+	;	>	=	ネ	マ	ヨ	°				
15	SI	US	BEL	SUB	!	^	?	"	ノ	ミ	ラ	°				MD

Figure 12-38. JapanEBCDIC Code Table (8CS00.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Start of heading	SOH
00/02	Start of text	STX
00/03	End of text	ETX
00/04		
00/05	Horizontal tabulation	HT
00/06		
00/07	Delete	DEL
00/08		
00/09		
00/10		
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

COL/ROW	ISO Name	SYM
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04		
01/05		
01/06	Backspace	BS
01/07		
01/08	Cancel	CAN
01/09	End medium	EM
01/10		
01/11		
01/12	File separator	FS
01/13	Group separator	GS
01/14	Record separator	RS
01/15	Unit separator	US

**Figure 12-39. JapanEBCDIC Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
02/00		
02/01		
02/02		
02/03		
02/04		
02/05	Line feed	LF
02/06	End-of-transmission block	ETB
02/07	Escape	ESC
02/08		
02/09		
02/10		
02/11	Start double octet	SDO
02/12	End double octet	EDO
02/13	Enquiry	ENQ
02/14	Acknowledgement	ACK
02/15	Bell	BEL

COL/ROW	ISO Name	SYM
03/00		
03/01		
03/02	Synchronous idle	SYN
03/03		
03/04		
03/05		
03/06		
03/07	End of transmission	EOT
03/08		
03/09		
03/10		
03/11		
03/12	Device control 4	DC4
03/13	Negative acknowledge	NAK
03/14		
03/15	Substituted character	SUB

**Figure 12-40. JapanEBCDIC Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
04/00	Space	SP
04/01	Halfwidth ideographic period	。
04/02	Halfwidth opening corner bracket	〔
04/03	Halfwidth closing corner bracket	〕
04/04	Halfwidth ideographic comma	、
04/05	Halfwidth Katakana middle dot	・
04/06	Halfwidth Katakana letter wo	ワ
04/07	Halfwidth Katakana letter small a	ア
04/08	Halfwidth Katakana letter small i	イ
04/09	Halfwidth Katakana letter small u	ウ
04/10	Left square bracket	[
04/11	Full stop	.
04/12	Less-than sign	<
04/13	Left parenthesis	(
04/14	Plus sign	+
04/15	Exclamation mark	!

COL/ROW	ISO Name	
05/00	Ampersand	&
05/01	Halfwidth Katakana letter small e	エ
05/02	Halfwidth Katakana letter small o	オ
05/03	Halfwidth Katakana letter small ya	ヤ
05/04	Halfwidth Katakana letter small yu	ユ
05/05	Halfwidth Katakana letter small yo	ヨ
05/06	Halfwidth Katakana letter small tu	ツ
05/07	Halfwidth Katakana-Hiragana prolonged sound mark	ー
05/08	Halfwidth Katakana letter a	ア
05/09		
05/10	Right square bracket]
05/11	Dollar sign	\$
05/12	Asterisk	*
05/13	Right parenthesis)
05/14	Semicolon	;
05/15	Circumflex accent	^

**Figure 12-41. JapanEBCDIC Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
06/00	Hyphen-minus	-
06/01	Solidus	/
06/02	Halfwidth Katakana letter i	イ
06/03	Halfwidth Katakana letter u	ウ
06/04	Halfwidth Katakana letter e	エ
06/05	Halfwidth Katakana letter o	オ
06/06	Halfwidth Katakana letter ka	カ
06/07	Halfwidth Katakana letter ki	キ
06/08	Halfwidth Katakana letter ku	ク
06/09	Halfwidth Katakana letter ke	ケ
06/10	Vertical line	
06/11	Comma	,
06/12	Percent sign	%
06/13	Low line	—
06/14	Greater-than sign	>
06/15	Question mark	?

COL/ROW	ISO Name	SYM
07/00	Halfwidth Katakana letter ko	コ
07/01	Halfwidth Katakana letter sa	サ
07/02	Halfwidth Katakana letter si	シ
07/03	Halfwidth Katakana letter su	ス
07/04	Halfwidth Katakana letter se	セ
07/05	Halfwidth Katakana letter so	ソ
07/06	Halfwidth Katakana letter ta	タ
07/07	Halfwidth Katakana letter ti	チ
07/08	Halfwidth Katakana letter tu	ツ
07/09	Grave Accent	`
07/10	Colon	:
07/11	Number sign	#
07/12	Commercial at	@
07/13	Apostrophe	'
07/14	Equal sign	=
07/15	Quotation mark	"

**Figure 12-42. JapanEBCDIC Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00	Halfwidth Katakana letter te	テ
08/01	Latin small letter a	a
08/02	Latin small letter b	b
08/03	Latin small letter c	c
08/04	Latin small letter d	d
08/05	Latin small letter e	e
08/06	Latin small letter f	f
08/07	Latin small letter g	g
08/08	Latin small letter h	h
08/09	Latin small letter i	i
08/10	Halfwidth Katakana letter to	ト
08/11	Halfwidth Katakana letter na	ナ
08/12	Halfwidth Katakana letter ni	ニ
08/13	Halfwidth Katakana letter nu	ヌ
08/14	Halfwidth Katakana letter ne	ネ
08/15	Halfwidth Katakana letter no	ノ

COL/ROW	ISO Name	SYM
09/00	Halfwidth Katakana letter ha	ハ
09/01	Latin small letter j	j
09/02	Latin small letter k	k
09/03	Latin small letter l	l
09/04	Latin small letter m	m
09/05	Latin small letter n	n
09/06	Latin small letter o	o
09/07	Latin small letter p	p
09/08	Latin small letter q	q
09/09	Latin small letter r	r
09/10	Halfwidth Katakana letter hi	ヒ
09/11	Halfwidth Katakana letter hu	フ
09/12	Halfwidth Katakana letter he	ヘ
09/13	Halfwidth Katakana letter ho	ホ
09/14	Halfwidth Katakana letter ma	マ
09/15	Halfwidth Katakana letter mi	ミ

**Figure 12-43. JapanEBCDIC Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	Halfwidth Katakana letter mu	ム
10/01	tilde	~
10/02	Latin small letter s	s
10/03	Latin small letter t	t
10/04	Latin small letter u	u
10/05	Latin small letter v	v
10/06	Latin small letter w	w
10/07	Latin small letter x	x
10/08	Latin small letter y	y
10/09	Latin small letter z	z
10/10	Halfwidth Katakana letter me	メ
10/11	Halfwidth Katakana letter mo	モ
10/12	Halfwidth Katakana letter ya	ヤ
10/13	Halfwidth Katakana letter yu	ユ
10/14	Halfwidth Katakana letter yo	ヨ
10/15	Halfwidth Katakana letter ra	ラ

COL/ROW	ISO Name	SYM
11/00	Halfwidth Katakana letter ri	リ
11/01	Halfwidth Katakana letter ru	ル
11/02		
11/03		
11/04		
11/05		
11/06		
11/07		
11/08		
11/09		
11/10	Halfwidth Katakana letter re	レ
11/11	Halfwidth Katakana letter ro	ロ
11/12	Halfwidth Katakana letter wa	ワ
11/13	Halfwidth Katakana letter n	ン
1 /14	Halfwidth Katakana voiced sound mark	゛
11/15	Halfwidth Katakana semi-voiced sound mark	゜

**Figure 12-44. JapanEBCDIC Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Left curly bracket	{
12/01	Latin capital letter A	A
12/02	Latin capital letter B	B
12/03	Latin capital letter C	C
12/04	Latin capital letter D	D
12/05	Latin capital letter E	E
12/06	Latin capital letter F	F
12/07	Latin capital letter G	G
12/08	Latin capital letter H	H
12/09	Latin capital letter I	I
12/10		
12/11		
12/12		
12/13		
12/14		
12/15		

COL/ROW	ISO Name	SYM
13/00	Right curly bracket	}
13/01	Latin capital letter J	J
13/02	Latin capital letter K	K
13/03	Latin capital letter L	L
13/04	Latin capital letter M	M
13/05	Latin capital letter N	N
13/06	Latin capital letter O	O
13/07	Latin capital letter P	P
13/08	Latin capital letter Q	Q
13/09	Latin capital letter R	R
13/10		
13/11		
13/12		
13/13		
13/14		
13/15		

**Figure 12-45. JapanEBCDIC Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Reverse solidus	\
14/01	Application program command	APC
14/02	Latin capital letter S	S
14/03	Latin capital letter T	T
14/04	Latin capital letter U	U
14/05	Latin capital letter V	V
14/06	Latin capital letter W	W
14/07	Latin capital letter X	X
14/08	Latin capital letter Y	Y
14/09	Latin capital letter Z	Z
14/10		
14/11		
14/12		
14/13		
14/14		
14/15		

COL/ROW	ISO Name	SYM
15/00	Digit 0	0
15/01	Digit 1	1
15/02	Digit 2	2
15/03	Digit 3	3
15/04	Digit 4	4
15/05	Digit 5	5
15/06	Digit 6	6
15/07	Digit 7	7
15/08	Digit 8	8
15/09	Digit 9	9
15/10		
15/11		
15/12		
15/13		
15/14		
15/15	Marked deleted	MD

**Figure 12-46. JapanEBCDIC Column Listing
Columns 14 and 15**

JapanV24 Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE			SP	&	-	j	†	γ	x		{	}	\$	0
01	SOH	DC1			。	エ	/	k	ア	タ	γ	~	A	J		1
02	STX	DC2		SYN	「	オ	a	l	イ	チ	ハ		B	K	S	2
03	ETX	DC3			」	ヤ	b	m	ウ	ツ	ホ		C	L	T	3
04					、	ユ	c	n	エ	テ	マ		D	M	U	4
05	HT		LF		・	ヨ	d	o	オ	ト	ミ		E	N	V	5
06		BS	ETB		ヲ	ツ	e	p	カ	ナ	ム		F	O	W	6
07	DEL		ESC	EOT	ア	、	f	q	キ	ニ	メ		G	P	X	7
08		CAN			イ	ー	g	r	ク	ヌ	モ		H	Q	Y	8
09		EM			ウ		h	s	ケ	ネ	ヤ		I	R	Z	9
10					[]	i	:	コ	ノ	ユ	レ				
11	VT		SDO		・	\	,	#	u	v	z	ロ				
12	FF	FS	EDO	DC4	<	*	%	@	サ	w	ヨ	ワ				
13	CF	GS	ENQ	NAK	()	-	'	シ	ハ	ラ	ン				
14	SO	RS	ACK		+	;	>	=	ス	ヒ	リ	・				
15	SI	US	BEL	SUB	!	^	?	"	セ	フ	ル	・				

Figure 12-47. JapanV24 Code Table (8CS00.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Start of heading	SOH
00/02	Start of text	STX
00/03	End of text	ETX
00/04		
00/05	Horizontal tabulation	HT
00/06		
00/07	Delete	DEL
00/08		
00/09		
00/10		
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

COL/ROW	ISO Name	SYM
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04		
01/05		
01/06	Backspace	BS
01/07		
01/08	Cancel	CAN
01/09	End medium	EM
01/10		
01/11		
01/12	File separator	FS
01/13	Group separator	GS
01/14	Record separator	RS
01/15	Unit separator	US

**Figure 12-48. JapanV24 Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
02/00		
02/01		
02/02		
02/03		
02/04		
02/05	Line feed	LF
02/06	End-of-transmission block	ETB
02/07	Escape	ESC
02/08		
02/09		
02/10		
02/11	Start double octet	SDO
02/12	End double octet	EDO
02/13	Enquiry	ENQ
02/14	Acknowledgement	ACK
02/15	Bell	BEL

COL/ROW	ISO Name	SYM
03/00		
03/01		
03/02	Synchronous idle	SYN
03/03		
03/04		
03/05		
03/06		
03/07	End of transmission	EOT
03/08		
03/09		
03/10		
03/11		
03/12	Device control 4	DC4
03/13	Negative acknowledge	NAK
03/14		
03/15	Substituted character	SUB

**Figure 12-49. JapanV24 Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
04/00	Space	SP
04/01	Halfwidth ideographic period	・
04/02	Halfwidth opening comer bracket	「
04/03	Halfwidth closing comer bracket	」
04/04	Halfwidth ideographic comma	、
04/05	Halfwidth Katakana middle dot	・
04/06	Halfwidth Katakana letter wo	ワ
04/07	Halfwidth Katakana letter small a	ア
04/08	Halfwidth Katakana letter small i	イ
04/09	Halfwidth Katakana letter small u	ウ
04/10	Left square bracket	[
04/11	Full stop	.
04/12	Less-than sign	<
04/13	Left parenthesis	(
04/14	Plus sign	+
04/15	Exclamation mark	!

COL/ROW	ISO Name	
05/00	Ampersand	&
05/01	Halfwidth Katakana letter small e	エ
05/02	Halfwidth Katakana letter small o	オ
05/03	Halfwidth Katakana letter small ya	ヤ
05/04	Halfwidth Katakana letter small yu	ユ
05/05	Halfwidth Katakana letter small yo	ヨ
05/06	Halfwidth Katakana letter small tu	ツ
05/07	Grave accent	`
05/08	Halfwidth Katakana-Hiragana prolonged sound mark	ー
05/09		
05/10	Right square bracket]
05/11	Reverse solidus	\
05/12	Asterisk	*
05/13	Right parenthesis)
05/14	Semicolon	;
05/15	Circumflex accent	^

**Figure 12-50. JapanV24 Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

CCL/ROW	ISO Name	SYM
06/00	Hyphen-minus	-
06/01	Solidus	/
06/02	Latin small letter a	a
06/03	Latin small letter b	b
06/04	Latin small letter c	c
06/05	Latin small letter d	d
06/06	Latin small letter e	e
06/07	Latin small letter f	f
06/08	Latin small letter g	g
06/09	Latin small letter h	h
06/10	Latin small letter i	i
06/11	Comma	,
06/12	Percent sign	%
06/13	Low line	_
06/14	Greater-than sign	>
06/15	Questionmark	?

CCL/ROW	ISO Name	SYM
07/00	Latin small letter j	j
07/01	Latin small letter k	k
07/02	Latin small letter l	l
07/03	Latin small letter m	m
07/04	Latin small letter n	n
07/05	Latin small letter o	o
07/06	Latin small letter p	p
07/07	Latin small letter q	q
07/08	Latin small letter r	r
07/09	Latin small letter s	s
07/10	Colon	:
07/11	Number sign	#
07/12	Commercial at	@
07/13	Apostrophe	'
07/14	Equal sign	=
07/15	Quotation mark	"

**Figure 12-51. JapanV24 Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00	Latin small letter t	†
08/01	Halfwidth Katakana letter a	ア
08/02	Halfwidth Katakana letter i	イ
08/03	Halfwidth Katakana letter u	ウ
08/04	Halfwidth Katakana letter e	エ
08/05	Halfwidth Katakana letter o	オ
08/06	Halfwidth Katakana letter ka	カ
08/07	Halfwidth Katakana letter ki	キ
08/08	Halfwidth Katakana letter ku	ク
08/09	Halfwidth Katakana letter ke	ケ
08/10	Halfwidth Katakana letter ko	コ
08/11	Latin small letter u	ウ
08/12	Halfwidth Katakana letter sa	サ
08/13	Halfwidth Katakana letter si	シ
08/14	Halfwidth Katakana letter su	ス
08/15	Halfwidth Katakana letter se	セ

COL/ROW	ISO Name	SYM
09/00	Halfwidth Katakana letter so	ソ
09/01	Halfwidth Katakana letter ta	タ
09/02	Halfwidth Katakana letter ti	チ
09/03	Halfwidth Katakana letter tu	ツ
09/04	Halfwidth Katakana letter te	テ
09/05	Halfwidth Katakana letter to	ト
09/06	Halfwidth Katakana letter na	ナ
09/07	Halfwidth Katakana letter ni	ニ
09/08	Halfwidth Katakana letter nu	ヌ
09/09	Halfwidth Katakana letter ne	ネ
09/10	Halfwidth Katakana letter no	ノ
09/11	Latin small letter v	v
09/12	Latin small letter w	w
09/13	Halfwidth Katakana letter ha	ハ
09/14	Halfwidth Katakana letter hi	ヒ
09/15	Halfwidth Katakana letter hu	フ

**Figure 12-52. JapanV24 Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	Latin small letter x	x
10/01	Latin small letter y	y
10/02	Halfwidth Katakana letter he	ハ
10/03	Halfwidth Katakana letter ho	ホ
10/04	Halfwidth Katakana letter ma	マ
10/05	Halfwidth Katakana letter mi	ミ
10/06	Halfwidth Katakana letter mu	ム
10/07	Halfwidth Katakana letter me	メ
10/08	Halfwidth Katakana letter mo	モ
10/09	Halfwidth Katakana letter ya	ヤ
10/10	Halfwidth Katakana letter yu	ユ
10/11	Latin small letter z	z
10/12	Halfwidth Katakana letter yo	ヨ
10/13	Halfwidth Katakana letter ra	ラ
10/14	Halfwidth Katakana letter ri	リ
10/15	Halfwidth Katakana letter ru	ル

COL/ROW	ISO Name	SYM
11/00	Vertical line	
11/01	Tilde	~
11/02		
11/03		
11/04		
11/05		
11/06		
11/07		
11/08		
11/09		
11/10	Halfwidth Katakana letter re	レ
11/11	Halfwidth Katakana letter ro	ロ
11/12	Halfwidth Katakana letter wa	ワ
11/13	Halfwidth Katakana letter n	ン
11/14	Halfwidth Katakana voiced sound mark	゛
11/15	Halfwidth Katakana semi-voiced sound mark	゜

**Figure 12-53. JapanV24 Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Left curly bracket	{
12/01	Latin capital letter A	A
12/02	Latin capital letter B	B
12/03	Latin capital letter C	C
12/04	Latin capital letter D	D
12/05	Latin capital letter E	E
12/06	Latin capital letter F	F
12/07	Latin capital letter G	G
12/08	Latin capital letter H	H
12/09	Latin capital letter I	I
12/10		
12/11		
12/12		
12/13		
12/14		
12/15		

COL/ROW	ISO Name	SYM
13/00	Right curly bracket	}
13/01	Latin capital letter J	J
13/02	Latin capital letter K	K
13/03	Latin capital letter L	L
13/04	Latin capital letter M	M
13/05	Latin capital letter N	N
13/06	Latin capital letter O	O
13/07	Latin capital letter P	P
13/08	Latin capital letter Q	Q
13/09	Latin capital letter R	R
13/10		
13/11		
13/12		
13/13		
13/14		
13/15		

**Figure 12-54. JapanV24 Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Dollar sign	\$
14/01		
14/02	Latin capital letter S	S
14/03	Latin capital letter T	T
14/04	Latin capital letter U	U
14/05	Latin capital letter V	V
14/06	Latin capital letter W	W
14/07	Latin capital letter X	X
14/08	Latin capital letter Y	Y
14/09	Latin capital letter Z	Z
14/10		
14/11		
14/12		
14/13		
14/14		
14/15		

COL/ROW	ISO Name	SYM
15/00	Digit 0	0
15/01	Digit 1	1
15/02	Digit 2	2
15/03	Digit 3	3
15/04	Digit 4	4
15/05	Digit 5	5
15/06	Digit 6	6
15/07	Digit 7	7
15/08	Digit 8	8
15/09	Digit 9	9
15/10		
15/11		
15/12		
15/13		
15/14		
15/15		

**Figure 12-55. JapanV24 Column Listing
Columns 14 and 15**

JISASCII Coded Character Set (8CS00.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p				ー	タ	ミ		
01	SOH	DC1	!	1	A	Q	a	q			。	ア	チ	ム		
02	STX	DC2	"	2	B	R	b	r			「	イ	ツ	メ		
03	ETX	DC3	#	3	C	S	c	s			」	ウ	テ	モ		
04	EOT	DC4	\$	4	D	T	d	t			、	エ	ト	ヤ		
05	ENQ	NAK	%	5	E	U	e	u			・	オ	ナ	ユ		
06	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
07	BEL	ETB	'	7	G	W	g	w			ア	キ	ヌ	ラ		
08	BS	CAN	(8	H	X	h	x			イ	ク	ネ	リ		
09	HT	EM)	9	I	Y	i	y			ウ	ケ	ノ	ル		
10	LF	SUB	*	:	J	Z	j	z			エ	コ	ハ	レ		
11	VT	ESC	+	;	K	[k	{			オ	サ	ヒ	ロ		
12	FF	FS	,	<	L	\	l				ヤ	シ	フ	ワ		
13	CR	GS	-	=	M]	m	}			ユ	ス	ハ	ン		
14	SO	RS	.	>	N	^	n	~			ヨ	セ	ホ	。		
15	SI	US	/	?	O	_	o	DEL			ッ	ソ	マ	。		

Figure 12-56. JISASCII Code Table (8CS00.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Start of heading	SOH
00/02	Start of text	STX
00/03	End of text	ETX
00/04	End of transmission	EOT
00/05	Enquiry	ENQ
00/06	Acknowledge	ACK
00/07	Bell	BEL
00/08	Backspace	BS
00/09	Horizontal tabulation	HT
00/10	Line feed	LF
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

COL/ROW	ISO Name	SYM
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04	Device control 4	DC4
01/05	Negative acknowledge	NAK
01/06	Synchronous idle	SYN
01/07	End-of-transmission block	ETB
01/08	Cancel	CAN
01/09	End medium	EM
01/10	Substituted character	SUB
01/11	Escape	ESC
01/12	File separator	FS
01/13	Group separator	GS
01/14	Record separator	RS
01/15	Unit separator	US

**Figure 12-57. JISASCII Column Listings
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 12-58. JISASCII Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 12-59. JISASCII Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	Sym
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Delete	DEL

**Figure 12-60. JISASCII Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
08/00		
08/01		
08/02		
08/03		
08/04		
08/05		
08/06		
08/07		
08/08		
08/09		
08/10		
08/11		
08/12		
08/13		
08/14		
08/15		

Col/Row	ISO Name	Sym
09/00		
09/01		
09/02		
09/03		
09/04		
09/05		
09/06		
09/07		
09/08		
09/09		
09/10		
09/11		
09/12		
09/13		
09/14		
09/15		

**Figure 12-61. JISASCII Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00		
10/01	Halfwidth ideographic period	。
10/02	Halfwidth opening corner bracket	「
10/03	Halfwidth closing corner bracket	」
10/04	Halfwidth ideographic comma	、
10/05	Halfwidth Katakana middle dot	・
10/06	Halfwidth Katakana letter wo	ヲ
10/07	Halfwidth Katakana letter small a	ァ
10/08	Halfwidth Katakana letter small i	ィ
10/09	Halfwidth Katakana letter small u	ゥ
10/10	Halfwidth Katakana letter small e	ヱ
10/11	Halfwidth Katakana letter small o	ォ
10/12	Halfwidth Katakana letter small ya	ャ
10/13	Halfwidth Katakana letter small yu	ュ
10/14	Halfwidth Katakana letter small yo	ョ
10/15	Halfwidth Katakana letter small tu	ヅ

COL/ROW	ISO Name	
11/00	Halfwidth Katakana-Hiragana prolonged sound mark	ー
11/01	Halfwidth Katakana letter a	ァ
11/02	Halfwidth Katakana letter i	ィ
11/03	Halfwidth Katakana letter u	ゥ
11/04	Halfwidth Katakana letter e	ヱ
11/05	Halfwidth Katakana letter o	ォ
11/06	Halfwidth Katakana letter ka	カ
11/07	Halfwidth Katakana letter ki	キ
11/08	Halfwidth Katakana letter ku	ク
11/09	Halfwidth Katakana letter ke	ケ
11/10	Halfwidth Katakana letter ko	コ
11/11	Halfwidth Katakana letter sa	サ
11/12	Halfwidth Katakana letter si	シ
11/13	Halfwidth Katakana letter su	ス
11/14	Halfwidth Katakana letter se	セ
11/15	Halfwidth Katakana letter so	ソ

**Figure 12-62. JISASCII Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Halfwidth Katakana letter ta	タ
12/01	Halfwidth Katakana letter ti	チ
12/02	Halfwidth Katakana letter tu	ツ
12/03	Halfwidth Katakana letter te	テ
12/04	Halfwidth Katakana letter to	ト
12/05	Halfwidth Katakana letter na	ナ
12/06	Halfwidth Katakana letter ni	ニ
12/07	Halfwidth Katakana letter nu	ヌ
12/08	Halfwidth Katakana letter ne	ネ
12/09	Halfwidth Katakana letter no	ノ
12/10	Halfwidth Katakana letter ha	ハ
12/11	Halfwidth Katakana letter hi	ヒ
12/12	Halfwidth Katakana letter hu	フ
12/13	Halfwidth Katakana letter he	ヘ
12/14	Halfwidth Katakana letter ho	ホ
12/15	Halfwidth Katakana letter ma	マ

COL/ROW	ISO Name	
13/00	Halfwidth Katakana letter mi	ミ
13/01	Halfwidth Katakana letter mu	ム
13/02	Halfwidth Katakana letter me	メ
13/03	Halfwidth Katakana letter mo	モ
13/04	Halfwidth Katakana letter ya	ヤ
13/05	Halfwidth Katakana letter yu	ユ
13/06	Halfwidth Katakana letter yo	ヨ
13/07	Halfwidth Katakana letter ra	ラ
13/08	Halfwidth Katakana letter ri	リ
13/09	Halfwidth Katakana letter ru	ル
13/10	Halfwidth Katakana letter re	レ
13/11	Halfwidth Katakana letter ro	ロ
13/12	Halfwidth Katakana letter wa	ワ
13/13	Halfwidth Katakana letter n	ン
13/14	Halfwidth Katakana voiced sound mark	・
13/15	Halfwidth Katakana semi-voiced sound mark	゜

**Figure 12-63. JISASCII Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
14/00		
14/01		
14/02		
14/03		
14/04		
14/05		
14/06		
14/07		
14/08		
14/09		
14/10		
14/11		
14/12		
14/13		
14/14		
14/15		

Col/Row	ISO Name	Sym
15/00		
15/01		
15/02		
15/03		
15/04		
15/05		
15/06		
15/07		
15/08		
15/09		
15/10		
15/11		
15/12		
15/13		
15/14		
15/15		

**Figure 12-64. JISASCII Column Listing
Columns 14 and 15**

**Latin1EBCDIC (8CS26.01) and Latin1ISO (8CS26.01)
Coded Character Sets**

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ø	Ø	°	μ	φ	{	}	\	0
01	SOH	DC1		PU1	NBSP	é	/	É	a	j	~	£	A	J	÷	1
02	STX	DC2	BPH	SYN	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03	ETX	DC3	NBH	STS	ä	ë	Ä	Ë	c	l	t	•	C	L	T	3
04	ST	OSC	IND	CCH	à	è	À	È	d	m	u	©	D	M	U	4
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v	§	E	N	V	5
06	SSA	BS	ETB	SPA	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07	APC	ESA	ESC	EOT	â	ï	À	Ï	g	p	x	¼	G	P	X	7
08	EPA	CAN	HTS	SOS	ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
09	RI	EM	HTJ		ñ	ß	Ñ	·	i	r	z	¾	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	«	a	i	¬	SHY	'	'	'
11	VT	SS3	PLD	CSI	·	\$,	#	»	º	¿		ø	ú	Ô	Û
12	FF	IS4	PLU	DC4	<	*	%	@	ð	æ	Ð	—	ö	ü	Ö	Ü
13	CR	IS3	ENQ	NAK	()	_	'	ý	.	Ý	..	ò	ù	Ò	Ù
14		IS2	ACK	PM	+	;	>	=	þ	Æ	Þ	,	ó	ú	Ó	Ú
15		IS1	BEL	SUB	!	^	?	"	±	€	®	x	õ	ÿ	Õ	DEL

Figure 12-65. Latin1EBCDIC Code Table (8CS26.01)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	°	À	Đ	à	đ
01	SOH	DC1	!	1	A	Q	a	q		PU1	ı	±	Á	Ñ	á	ñ
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	¢	²	Â	Ò	â	ò
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	£	³	Ã	Ó	ã	ó
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	€	´	Ä	Ô	ä	ô
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	¥	µ	Å	Õ	å	õ
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA		¶	Æ	Ö	æ	ö
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA	§	•	Ç	x	ç	÷
08	BS	CAN	(8	H	X	h	x	HTS	SOS	¨	.	È	Ø	è	ø
09	HT	EM)	9	I	Y	i	y	HTJ		©	¹	É	Ù	é	ù
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	ª	º	Ê	Ú	ê	ú
11	VT	ESC	+	;	K	[k	{	PLD	CSI	«	»	Ë	Û	ë	û
12	FF	IS4	,	<	L	\	l		PLU	ST	¬	¼	Ì	Ü	ì	ü
13	CR	IS3	-	=	M]	m	}	RI	OSC	SHY	½	Í	Ý	í	ý
14	SO	IS2	.	>	N	^	n	~	SS2	PM	®	¾	Î	Þ	î	þ
15	SI	IS1	/	?	O	_	o	DEL	SS3	APC	¯	¿	Ï	ß	ï	ÿ

Figure 12-66. Latin1ISO Code Table (8CS26.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Shift out	SO
00/15	00/15	Shift in	SI

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

**Figure 12-67. Latin1EBCDIC and Latin1ISO Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledgement	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substituted character	SUB

**Figure 12-68. Latin1EBCDIC and Latin1ISO Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	14/02	Latin small letter a with circumflex accent	â
04/03	14/04	Latin small letter a with diaeresis	ä
04/04	14/00	Latin small letter a with grave accent	à
04/05	14/01	Latin small letter a with acute accent	á
04/06	14/03	Latin small letter a with tilde	ã
04/07	14/05	Latin small letter a with ring above	â
04/08	14/07	Latin small letter c with cedilla	ç
04/09	15/01	Latin small letter n with tilde	ñ
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	14/09	Latin small letter e with acute accent	é
05/02	14/10	Latin small letter e with circumflex accent	ê
05/03	14/11	Latin small letter e with diaeresis	ë
05/04	14/08	Latin small letter e with grave accent	è
05/05	14/13	Latin small letter i with acute accent	í
05/06	14/14	Latin small letter i with circumflex accent	î
05/07	14/15	Latin small letter i with diaeresis	ï
05/08	14/12	Latin small letter i with grave accent	ì
05/09	13/15	Latin small letter sharp s (German)	ß
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

**Figure 12-69. Latin1EBCDIC and Latin1ISO Column Listing
Columns 4 and 5**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	12/02	Latin capital letter A with circumflex accent	Â
06/03	12/04	Latin capital letter A with diaeresis	Ä
06/04	12/00	Latin capital letter A with grave accent	À
06/05	12/01	Latin capital letter A with acute accent	Á
06/06	12/03	Latin capital letter A with tilde	Ã
06/07	12/05	Latin capital letter A with ring above	Å
06/08	12/07	Latin capital letter C with cedilla	Ç
06/09	13/01	Latin capital letter N with tilde	Ñ
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	15/08	Latin small letter o with oblique stroke	ø
07/01	12/09	Latin capital letter E with acute accent	É
07/02	12/10	Latin capital letter E with circumflex accent	Ê
07/03	12/11	Latin capital letter E with diaeresis	Ë
07/04	12/08	Latin capital letter E with grave accent	È
07/05	12/13	Latin capital letter I with acute accent	Í
07/06	12/14	Latin capital letter I with circumflex accent	Î
07/07	12/15	Latin capital letter I with diaeresis	Ï
07/08	12/12	Latin capital letter I with grave accent	Ì
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

**Figure 12-70. Latin1EBCDIC and Latin1ISO Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00	13/08	Latin capital letter O with oblique stroke	Ø
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	10/11	Left-pointing double angle quotation mark	«
08/11	11/11	Right-pointing double angle quotation mark	»
08/12	15/00	Latin small letter eth (Icelandic)	ð
08/13	15/13	Latin small letter y with acute accent	ý
08/14	15/14	Latin small letter thorn (Icelandic)	þ
08/15	11/01	Plus-minus sign	±

EBCDIC	ISO	ISO Name	SYM
09/00	11/00	Degree sign	°
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	10/10	Feminine ordinal indicator	ª
09/11	11/10	Masculine ordinal indicator	º
09/12	14/06	Latin small letter a and e	æ
09/13	11/08	Cedilla	¸
09/14	12/06	Latin capital letter A with E	Æ
09/15	10/04	Euro sign	€

**Figure 12-71. Latin1EBCDIC and Latin1ISO Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	11/05	Micro sign	μ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	10/01	Inverted exclamation mark	¡
10/11	11/15	Inverted question mark	¿
10/12	13/00	Latin capital letter eth (Icelandic)	Ð
10/13	13/13	Latin capital letter Y with acute accent	Ý
10/14	13/14	Latin capital letter thorn (Icelandic)	Þ
10/15	10/14	Registered trademark sign	®

EBCDIC	ISO	ISO Name	SYM
11/00	10/02	Cent sign	¢
11/01	10/03	Pound sign	£
11/02	10/05	Yen sign	¥
11/03	10/07	Middle dot	•
11/04	10/09	Copyright sign	©
11/05	10/07	Paragraph sign	§
11/06	11/06	Pilcrow sign	¶
11/07	11/12	Vulgar fraction one quarter	¼
11/08	11/13	Vulgar fraction one half	½
11/09	11/14	Vulgar fraction three quarters	¾
11/10	10/12	Not sign	¬
11/11	10/06	Broken bar	
11/12	10/15	Macron	—
11/13	10/08	Diaeresis	¨
11/14	11/04	Acute accent	´
11/15	13/07	Multiplication sign	×

**Figure 12-72. Latin1EBCDIC and Latin1ISO Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hyphen	SHY
12/11	15/04	Latin small letter o with circumflex accent	ô
12/12	15/06	Latin small letter o with diaeresis	ö
12/13	15/02	Latin small letter o with grave accent	ò
12/14	15/03	Latin small letter o with acute accent	ó
12/15	15/05	Latin small letter o with tilde	õ

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	11/09	Superscript 1	¹
13/11	15/11	Latin small letter u with circumflex accent	û
13/12	15/12	Latin small letter u with diaeresis	ü
13/13	15/09	Latin small letter u with grave accent	ù
13/14	15/10	Latin small letter u with acute accent	ú
13/15	15/15	Latin small letter y with diaeresis	ÿ

Figure 12-73. Latin1EBCDIC and Latin1ISO Column Listing Columns 12 and 13

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	15/07	Division sign	÷
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	11/02	Superscript 2	²
14/11	13/04	Latin capital letter O with circumflex accent	Ô
14/12	13/06	Latin capital letter O with diaeresis	Ö
14/13	13/02	Latin capital letter O with grave accent	Ò
14/14	13/03	Latin capital letter O with acute accent	Ó
14/15	13/05	Latin capital letter O with tilde	Õ

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	11/03	Superscript 3	³
15/11	13/11	Latin capital letter U with circumflex accent	Û
15/12	13/12	Latin capital letter U with diaeresis	Ü
15/13	13/09	Latin capital letter U with grave accent	Ù
15/14	13/10	Latin capital letter U with acute accent	Ú
15/15	07/15	Delete	DEL

**Figure 12-74. Latin1EBCDIC and Latin1ISO Column Listing
Columns 14 and 15**

**Latin2EBCDIC (8CS27.01) and Latin2ISO (8CS27.00)
Coded Character Sets**

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	˘	˘	°	g	·	{	}	\	0
01	SOH	DC1		PU1	NBSP	é	/	É	a	j	˜	À	A	J	÷	1
02	STX	DC2	BPH	SYN	â	ę	Â	Ę	b	k	s	ż	B	K	S	2
03	ETX	DC3	NBH	STS	ä	ë	Ä	Ë	c	l	t	ı	C	L	T	3
04	ST	OSC	IND	CCH	ı	ü	”	Û	d	m	u	ž	D	M	U	4
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v	ş	E	N	V	5
06	SSA	BS	ETB	SPA	ǎ	î	Ă	Î	f	o	w	ž	F	O	W	6
07	APC	ESA	ESC	EOT	č	ř	Č	Ř	g	p	x	ž	G	P	X	7
08	EPA	CAN	HTS	SOS	ç	í	Ç	Í	h	q	y	ž	H	Q	Y	8
09	RI	EM	HTJ		ć	β	Ć	`	i	r	z	ž	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	ś	ŕ	ś	ł	SHY	Ě	ď	Ď
11	VT	SS3	PLD	CSI	·	\$,	#	ň	ń	Ň	Ń	ô	ú	Ô	Ú
12	FF	IS4	PLU	DC4	<	*	%	@	ď	š	Đ	Š	ö	ü	Ö	Ü
13	CR	IS3	ENQ	NAK	()	-	'	ý	ı	Ý	”	í	ř	Ř	ř
14		IS2	ACK	PM	+	;	>	=	ř	ı	Ř	'	ó	ú	Ó	Ú
15		IS1	BEL	SUB	!	^	?	"	ş	€	Ş	×	ó	ě	Ó	DEL

Figure 12-75. Latin2EBCDIC Code Table (8CS27.01)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	°	´	Đ	ı	đ
01	SOH	DC1	!	1	A	Q	a	q		PU1	À	à	Á	Ñ	á	ñ
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	˘	˙	Â	Ň	â	ň
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	Ł	ł	Ǻ	Ó	ǻ	ó
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	€	·	Ǽ	Ô	ǿ	ô
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	Ĺ	ĺ	Ľ	Ó	í	ó
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA	Ś	ś	Ć	Ö	ć	ö
07	BEL	ETB	,	7	G	W	g	w	ESA	EPA	Ş	ş	Ç	x	ç	÷
08	BS	CAN	(8	H	X	h	x	HTS	SOS	ˆ	˚	Č	Ř	č	ř
09	HT	EM)	9	I	Y	i	y	HTJ		Š	š	É	Ů	é	ů
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	Ş	ş	Ę	Ú	ę	ú
11	VT	ESC	+	;	K	[k	{	PLD	CSI	ř	ř	Ě	Ú	ě	ů
12	FF	IS4	,	<	L	\	l		PLU	ST	Ž	ž	Ě	Ü	ě	ü
13	CR	IS3	-	=	M]	m	}	RI	OSC	SHY	ˆ	Í	Ý	í	ý
14	SO	IS2	.	>	N	^	n	~	SS2	PM	Ž	ž	Î	Ť	î	ť
15	SI	IS1	/	?	O	-	o	DEL	SS3	APC	Ž	ž	Ď	ß	ď	·

Figure 12-76. Latin2ISO Code Table (8CS27.00)

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Shift out	SO
00/15	00/15	Shift in	SI

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

**Figure 12-77. Latin2EBCDIC and Latin2ISO Column Listing
Columns 0 and 1**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	08/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledge	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substitute character	SUB

**Figure 12-78. Latin2EBCDIC and Latin2ISO Column Listing
Columns 2 and 3**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	14/02	Latin small letter a with circumflex accent	â
04/03	14/04	Latin small letter a with diaeresis	ä
04/04	15/14	Latin small letter t with cedilla	ţ
04/05	14/01	Latin small letter a with acute accent	á
04/06	14/03	Latin small letter a with breve	ă
04/07	14/08	Latin small letter c with caron	č
04/08	14/07	Latin small letter c with cedilla	ç
04/09	14/06	Latin small letter c with acute accent	ć
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	14/09	Latin small letter e with acute accent	é
05/02	14/10	Latin small letter e with ogonek	ę
05/03	14/11	Latin small letter e with diaeresis	ë
05/04	15/09	Latin small letter u with ring above	û
05/05	14/13	Latin small letter i with acute accent	í
05/06	14/14	Latin small letter i with circumflex accent	î
05/07	11/05	Latin small letter l with caron	ĺ
05/08	14/05	Latin small letter l with acute accent	ĺ
05/09	13/15	Latin small letter sharp s (German)	ß
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

Figure 12-79. Latin2EBCDIC and Latin2ISO Column Listing Columns 4 and 5

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	12/02	Latin capital letter A with circumflex accent	Â
06/03	12/04	Latin capital letter A with diaeresis	Ä
06/04	11/13	Double acute accent	˝
06/05	12/01	Latin capital letter A with acute accent	Á
06/06	12/03	Latin capital letter A with breve	Ă
06/07	12/08	Latin capital letter C with caron	Č
06/08	12/07	Latin capital letter C with cedilla	Ç
06/09	12/06	Latin capital letter C with acute accent	Ć
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	11/07	Caron accent	ˇ
07/01	12/09	Latin capital letter E with acute accent	É
07/02	12/10	Latin capital letter E with ogonek	Ę
07/03	12/11	Latin capital letter E with diaeresis	Ë
07/04	13/09	Latin capital letter U with ring above	Û
07/05	12/13	Latin capital letter I with acute accent	Í
07/06	12/14	Latin capital letter I with circumflex accent	Î
07/07	10/05	Latin capital letter L with caron	Ľ
07/08	12/05	Latin capital letter L with acute accent	Ł
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

**Figure 12-80. Latin2EBCDIC and Latin2ISO Column Listing
Columns 6 and 7**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00	10/02	Breve accent	˘
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	11/06	Latin small letter s with acute accent	ś
08/11	15/02	Latin small letter n with caron	ň
08/12	15/00	Latin small letter d with stroke	đ
08/13	15/13	Latin small letter y with acute accent	ý
08/14	15/08	Latin small letter r with caron	ř
08/15	11/10	Latin small letter s with cedilla	ș

EBCDIC	ISO	ISO Name	SYM
09/00	11/00	Degree sign	°
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	11/03	Latin small letter l with stroke	ł
09/11	15/01	Latin small letter n with acute accent	ñ
09/12	11/09	Latin small letter s with caron	š
09/13	11/02	Cedilla	¸
09/14	12/06	Ogonek accent	˛
09/15	10/04	Euro Sign	€

**Figure 12-81. Latin2EBCDIC and Latin2ISO Column Listing
Columns 8 and 9**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	11/01	Latin small letter a with ogonek	ą
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	10/06	Latin capital letter S with acute accent	Ŝ
10/11	13/02	Latin capital letter N with caron	Ñ
10/12	13/00	Latin capital letter D with stroke	Đ
10/13	13/13	Latin capital letter Y with acute accent	Ÿ
10/14	13/08	Latin capital letter R with caron	Ř
10/15	10/10	Latin capital letter S with cedilla	Ş

EBCDIC	ISO	ISO Name	SYM
11/00	15/15	Dot above	•
11/01	10/01	Latin capital letter A with ogonek	Ą
11/02	11/15	Latin small letter z with dot above	ž
11/03	13/14	Latin capital letter T with cedilla	Ț
11/04	10/15	Latin capital letter Z with dot above	Ž
11/05	10/07	Paragraph sign	§
11/06	11/14	Latin small letter z with caron	ž
11/07	11/12	Latin small letter z with acute accent	ż
11/08	10/14	Latin capital letter Z with caron	Ž
11/09	10/12	Latin capital letter Z with acute accent	Ż
11/10	10/03	Latin capital letter L with stroke	Ł
11/11	13/01	Latin capital letter N with acute accent	Ń
11/12	10/09	Latin capital letter S with caron	Š
11/13	10/08	Diaeresis	¨
11/14	11/04	Acute accent	´
11/15	13/07	Multiplication sign	×

**Figure 12-82. Latin2EBCDIC and Latin2ISO Column Listing
Columns 10 and 11**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hyphen	SHY
12/11	15/04	Latin small letter o with circumflex accent	ô
12/12	15/06	Latin small letter o with diaeresis	ö
12/13	14/00	Latin small letter r with acute accent	ř
12/14	15/03	Latin small letter o with acute accent	ó
12/15	15/05	Latin small letter o with double acute accent	ő

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	12/12	Latin capital letter E with caron accent	Ě
13/11	15/11	Latin small letter u with double acute accent	ű
13/12	15/12	Latin small letter u with diaeresis	ü
13/13	11/11	Latin small letter t with caron	ť
13/14	15/10	Latin small letter u with acute accent	ú
13/15	14/12	Latin small letter e with caron	ě

**Figure 12-83. Latin2EBCDIC and Latin2ISO Column Listing
Columns 12 and 13**

Eight-Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	15/07	Division sign	÷
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	14/15	Latin small letter d with caron	ď
14/11	13/04	Latin capital letter O with circumflex accent	Ô
14/12	13/06	Latin capital letter O with diaeresis	Ö
14/13	12/00	Latin capital letter R with acute accent	Ř
14/14	13/03	Latin capital letter O with acute accent	Ó
14/15	13/05	Latin capital letter O with double acute accent	Ő

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	12/15	Latin capital letter D with caron	Ď
15/11	13/11	Latin capital letter U with double acute accent	Ű
15/12	13/12	Latin capital letter U with diaeresis	Ü
15/13	10/11	Latin capital letter T with caron accent	Ť
15/14	13/10	Latin capital letter U with acute accent	Ú
15/15	07/15	Delete	DEL

**Figure 12-84. Latin2EBCDIC and Latin2ISO Column Listing
Columns 14 and 15**

**Latin5EBCDIC (8CS29.01) and Latin5ISO (8CS29.00)
Coded Character Sets**

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ø	Ø	°	μ	ϕ	{	}	\	0
01	SOH	DC1		PU1	NBSP	é	/	É	á	j	~	£	A	J	÷	1
02	STX	DC2	BPH	SYN	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03	ETX	DC3	NBH	STS	ä	ë	Ä	Ë	c	l	†	•	C	L	T	3
04	ST	OSC	IND	CCH	à	è	À	È	d	m	u	©	D	M	U	4
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v	§	E	N	V	5
06	SSA	BS	ETB	SPA	ã	ï	Ã	Ï	f	o	w	¶	F	O	W	6
07	APC	ESA	ESC	EOT	ä	†	Å	ï	g	p	x	¼	G	P	X	7
08	EPA	CAN	HTS	SOS	ç	†	Ç	ì	h	q	y	½	H	Q	Y	8
09	RI	EM	HTJ		ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	«	ª	¡	¬	SHY	¹	²	³
11	VT	SS3	PLD	CSI	.	\$,	#	»	º	¿	¡	ô	ù	Ô	Ù
12	FF	IS4	PLU	DC4	<	*	%	@	ǧ	œ	Ǵ	-	ö	ü	Ö	Ü
13	CR	IS3	ENQ	NAK	()	_	'	ı	ı	ı	ı	ò	ù	Ò	Ù
14		IS2	ACK	PM	+	;	>	=	ş	Æ	Ş	'	ó	ú	Ó	Ú
15		IS1	BEL	SUB	!	^	?	"	±	€	®	×	õ	ÿ	Õ	DEL

Figure 12-85. Latin5EBCDIC Code Table (8CS29.01)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ø	Ø	°	μ	ϕ	{	}	\	0
01	SOH	DC1		PU1	NBSP	é	/	É	á	j	~	£	A	J	÷	1
02	STX	DC2	BPH	SYN	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03	ETX	DC3	NBH	STS	ä	ë	Ä	Ë	c	l	†	•	C	L	T	3
04	ST	OSC	IND	CCH	à	è	À	È	d	m	u	©	D	M	U	4
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v	§	E	N	V	5
06	SSA	BS	ETB	SPA	ã	ï	Ã	Ï	f	o	w	¶	F	O	W	6
07	APC	ESA	ESC	EOT	ä	†	Ä	ï	g	p	x	¼	G	P	X	7
08	EPA	CAN	HTS	SOS	ç	†	Ç	ì	h	q	y	½	H	Q	Y	8
09	RI	EM	HTJ		ñ	ß	Ñ	`	i	r	z	¾	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	«	ª	¡	¬	SHY	¹	²	³
11	VT	SS3	PLD	CSI	.	\$,	#	»	º	¿	¦	ô	ù	ô	û
12	FF	IS4	PLU	DC4	<	*	%	@	ǧ	œ	Ǧ	-	ö	ü	ö	ü
13	CR	IS3	ENQ	NAK	()	_	'	i	¸	¡	¨	ò	ù	ò	ù
14		IS2	ACK	PM	+	;	>	=	§	Æ	§	'	ó	ú	ó	ú
15		IS1	BEL	SUB	!	^	?	"	±	€	®	×	õ	ÿ	õ	DEL

Figure 12-86. Latin5ISO Code Table (8CS29.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Shift out	SO
00/15	00/15	Shift in	SI

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

Figure 12-87. Latin5EBCDIC and Latin5ISO Column Listing Columns 0 and 1

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledge	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substitute character	SUB

**Figure 12-88. Latin5EBCDIC and Latin5ISO Column Listing
Columns 2 and 3**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	14/02	Latin small letter a with circumflex accent	â
04/03	14/04	Latin small letter a with diaeresis	ä
04/04	14/00	Latin small letter a with grave accent	à
04/05	14/01	Latin small letter a with acute accent	á
04/06	14/03	Latin small letter a with tilde	ã
04/07	14/05	Latin small letter a with ring above	å
04/08	14/07	Latin small letter c with cedilla	ç
04/09	15/01	Latin small letter n with tilde	ñ
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	14/09	Latin small letter e with acute accent	é
05/02	14/10	Latin small letter e with circumflex accent	ê
05/03	14/11	Latin small letter e with diaeresis	ë
05/04	14/08	Latin small letter e with grave accent	è
05/05	14/13	Latin small letter i with acute accent	í
05/06	14/14	Latin small letter i with circumflex accent	î
05/07	14/15	Latin small letter i with diaeresis	ï
05/08	14/12	Latin small letter i with grave accent	ì
05/09	13/15	Latin small letter sharp s (German)	ß
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

**Figure 12–89. Latin5EBCDIC and Latin5ISO Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	12/02	Latin capital letter A with circumflex accent	Â
06/03	12/04	Latin capital letter A with diaeresis	Ä
06/04	12/00	Latin capital letter A with grave accent	À
06/05	12/01	Latin capital letter A with acute accent	Á
06/06	12/03	Latin capital letter A with tilde	Ã
06/07	12/05	Latin capital letter A with ring above	Å
06/08	12/07	Latin capital letter C with cedilla	Ç
06/09	13/01	Latin capital letter N with tilde	Ñ
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	15/08	Latin small letter o with oblique stroke	ø
07/01	12/09	Latin capital letter E with acute accent	É
07/02	12/10	Latin capital letter E with circumflex accent	Ê
07/03	12/11	Latin capital letter E with diaeresis	Ë
07/04	12/08	Latin capital letter E with grave accent	È
07/05	12/13	Latin capital letter I with acute accent	Í
07/06	12/14	Latin capital letter I with circumflex accent	Î
07/07	12/15	Latin capital letter I with diaeresis	Ï
07/08	12/12	Latin capital letter I with grave accent	Ì
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

**Figure 12–90. Latin5EBCDIC and Latin5ISO Column Listing
Columns 6 and 7**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00	13/08	Latin capital letter O with oblique stroke	Ø
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	10/11	Left-pointing double angle quotation mark	«
08/11	11/11	Right-pointing double angle quotation mark	»
08/12	15/00	Latin small letter g with breve	ğ
08/13	15/13	Latin small letter i without dot above	ı
08/14	15/14	Latin small letter s with cedilla	ş
08/15	11/01	Plus-minus sign	±

EBCDIC	ISO	ISO Name	SYM
09/00	11/00	Degree sign	°
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	10/10	Feminine ordinal indicator	ª
09/11	11/10	Masculine ordinal indicator	º
09/12	14/06	Latin small letter a with e	æ
09/13	11/08	Cedilla	¸
09/14	12/06	Latin capital letter A with E	Æ
09/15	10/04	Euro sign	€

**Figure 12-91. Latin5EBCDIC and Latin5ISO Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	11/05	Micro sign	μ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	10/01	Inverted exclamation mark	¡
10/11	11/15	Inverted question mark	¿
10/12	13/00	Latin capital letter G with breve	Ğ
10/13	13/13	Latin capital letter I with dot above	İ
10/14	13/14	Latin capital letter S with cedilla	Ş
10/15	10/14	Registered trademark sign	®

EBCDIC	ISO	ISO Name	SYM
11/00	10/02	Cent sign	¢
11/01	10/03	Pound sign	£
11/02	10/05	Yen sign	¥
11/03	11/07	Middle dot	•
11/04	10/09	Copyright sign	©
11/05	10/07	Paragraph sign	§
11/06	11/06	Pilcrow sign	¶
11/07	11/12	Vulgar fraction one quarter	¼
11/08	11/13	Vulgar fraction one half	½
11/09	11/14	Vulgar fraction three quarters	¾
11/10	10/12	Not sign	¬
11/11	10/06	Broken bar	
11/12	10/15	Macron	—
11/13	10/08	Diaeresis	¨
11/14	11/04	Acute accent	´
11/15	13/07	Multiplication sign	×

**Figure 12–92. Latin5EBCDIC and Latin5ISO Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hyphen	SHY
12/11	15/04	Latin small letter o with circumflex accent	ô
12/12	15/06	Latin small letter o with diaeresis	ö
12/13	15/02	Latin small letter o with grave accent	ò
12/14	15/03	Latin small letter o with acute accent	ó
12/15	15/05	Latin small letter o with tilde	õ

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	11/09	Superscript 1	¹
13/11	15/11	Latin small letter u with circumflex accent	û
13/12	15/12	Latin small letter u with diaeresis	ü
13/13	15/09	Latin small letter u with grave accent	ù
13/14	15/10	Latin small letter u with acute accent	ú
13/15	15/15	Latin small letter y with diaeresis	ÿ

**Figure 12-93. Latin5EBCDIC and Latin5ISO Column Listing
Columns 12 and 13**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	15/07	Division sign	÷
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	11/02	Superscript 2	²
14/11	13/04	Latin capital letter O with circumflex accent	Ô
14/12	13/06	Latin capital letter O with diaeresis	Ö
14/13	13/02	Latin capital letter O with grave accent	Ò
14/14	13/03	Latin capital letter O with acute accent	Ó
14/15	13/05	Latin capital letter O with tilde	Õ

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	11/03	Superscript 3	³
15/11	13/11	Latin capital letter U with circumflex accent	Û
15/12	13/12	Latin capital letter U with diaeresis	Ü
15/13	13/09	Latin capital letter U with grave accent	Ù
15/14	13/10	Latin capital letter U with acute accent	Ú
15/15	07/15	Delete	DEL

**Figure 12-94. Latin5EBCDIC and Latin5ISO Column Listing
Columns 14 and 15**

Latin9EBCDIC (8CS00.00) and Latin9ISO (8CS00.00) Coded Character Sets

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ø	Ø	°	μ	φ	{	}	\	0
01	SOH	DC1		PU1	NBSF	é	/	É	a	j	~	£	A	J	÷	1
02	STX	DC2	BPH	SYN	â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03	ETX	DC3	NBH	STS	ä	ë	Ä	Ë	c	l	t	.	C	L	T	3
04	ST	OSC	IND	CCH	à	è	À	È	d	m	u	©	D	M	U	4
05	HT	NEL	LF	MW	á	í	Á	Í	e	n	v	§	E	N	V	5
06	SSA	BS	ETB	SPA	ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07	APC	ESA	ESC	EOT	à	ï	À	Ï	g	p	x	Œ	G	P	X	7
08	EPA	CAN	HTS	SOS	ç	ì	Ç	Ì	h	q	y	œ	H	Q	Y	8
09	RI	EM	HTJ		ñ	ß	Ñ	·	i	r	z	ÿ	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	«	ª	ı	¬	SHY	'	²	³
11	VT	SS3	PLD	CSI	·	\$,	#	»	º	ı	Š	ô	û	Ô	Û
12	FF	IS4	PLU	DC4	<	*	%	@	ð	æ	Ð	—	ö	ü	Ö	Ü
13	CR	IS3	ENQ	NAK	()	_	'	ý	ž	Ý	š	ò	ù	Ò	Ù
14		IS2	ACK	PM	+	;	>	=	þ	Æ	Þ	ž	ó	ú	Ó	Ú
15		IS1	BEL	SUB	!	^	?	"	±	€	®	×	õ	ÿ	Õ	DEL

Figure 12-95. Latin9EBCDIC Code Table (8CS00.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	°	À	Ğ	à	ğ
01	SOH	DC1	!	1	A	Q	a	q		PU1	i	±	Á	Ñ	á	ñ
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	ç	²	Â	Ò	â	ò
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	£	³	Ã	Ó	ã	ó
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	€	´	Ä	Ô	ä	ô
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	¥	µ	Å	Õ	å	õ
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA	¡	¶	Æ	Ö	æ	ö
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA	§	•	Ç	x	ç	÷
08	BS	CAN	(8	H	X	h	x	HTS	SOS	¨	¸	È	Ø	è	ø
09	HT	EM)	9	I	Y	i	y	HTJ		©	¹	É	Ù	é	ù
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	ª	º	Ê	Ú	ê	ú
11	VT	ESC	+	;	K	[k	{	PLD	CSI	«	»	Ë	Û	ë	û
12	FF	IS4	,	<	L	\	l		PLU	ST	¬	¼	Ì	Ü	ì	ü
13	CR	IS3	-	=	M]	m	}	RI	OSC	SHY	½	Í	İ	ı	ı
14	SO	IS2	.	>	N	^	n	~	SS2	PM	®	¾	Î	Ş	î	ş
15	SI	IS1	/	?	O	_	o	DEL	SS3	APC	-	¿	Ï	ß	ï	ÿ

Figure 12-96. Latin9ISO Code Table (8CS00.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SO H
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse Index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Shift out	SO
00/15	00/15	Shift in	SI

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operation system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

**Figure 12-97. Latin9EBCDIC and Latin9ISO Column Listing
Columns 0 and 1**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	08/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	LF
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledgement	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substituted character	SUB

**Figure 12-98. Latin9EBCDIC and Latin9ISO Column Listing
Columns 2 and 3**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	14/02	Latin small letter a with circumflex accent	â
04/03	14/04	Latin small letter a with diaeresis	ä
04/04	14/00	Latin small letter a with grave accent	à
04/05	14/01	Latin small letter a with acute accent	á
04/06	14/03	Latin small letter a with tilde	ã
04/07	14/05	Latin small letter a with ring above	å
04/08	14/07	Latin small letter c with cedilla	ç
04/09	15/01	Latin small letter n with tilde	ñ
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	14/09	Latin small letter e with acute accent	é
05/02	14/10	Latin small letter e with circumflex accent	ê
05/03	14/11	Latin small letter e with diaeresis	ë
05/04	14/08	Latin small letter e with grave accent	è
05/05	14/13	Latin small letter i with acute accent	í
05/06	14/14	Latin small letter i with circumflex accent	î
05/07	14/15	Latin small letter i with diaeresis	ï
05/08	14/12	Latin small letter i with grave accent	ì
05/09	13/15	Latin small letter sharp s (German)	ß
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

**Figure 12-99. Latin9EBCDIC and Latin9ISO Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	12/02	Latin capital letter A with circumflex accent	Â
06/03	12/04	Latin capital letter A with diaeresis	Ä
06/04	12/00	Latin capital letter A with grave accent	À
06/05	12/01	Latin capital letter A with acute accent	Á
06/06	12/03	Latin capital letter A with tilde	Ã
06/07	12/05	Latin capital letter A with ring above	Å
06/08	12/07	Latin capital letter C with cedilla	Ç
06/09	13/01	Latin capital letter N with tilde	Ñ
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	15/08	Latin small letter o with oblique stroke	ø
07/01	12/09	Latin capital letter E with acute accent	É
07/02	12/10	Latin capital letter E with circumflex accent	Ê
07/03	12/11	Latin capital letter E with diaeresis	Ë
07/04	12/08	Latin capital letter E with grave accent	È
07/05	12/13	Latin capital letter I with acute accent	Í
07/06	12/14	Latin capital letter I with circumflex accent	Î
07/07	12/15	Latin capital letter I with diaeresis	Ï
07/08	12/12	Latin capital letter I with grave accent	Ì
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

**Figure 12-100. Latin9EBCDIC and Latin9ISO Column Listing
Columns 6 and 7**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00	13/08	Latin capital letter O with oblique stroke	Ø
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	10/11	Left-pointing double angle quotation mark	«
08/11	11/11	Right-pointing double angle quotation mark	»
08/12	15/00	Latin small letter eth (Icelandic)	ð
08/13	15/13	Latin small letter y with acute accent	ý
08/14	15/14	Latin small letter thorn (Icelandic)	þ
08/15	11/01	Plus-minus sign	±

EBCDIC	ISO	ISO Name	SYM
09/00	11/00	Degree sign	°
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	10/10	Feminine ordinal indicator	ª
09/11	11/10	Masculine ordinal indicator	º
09/12	14/06	Latin small letter a and e	æ
09/13	11/08	Latin small letter z with caron	ž
09/14	12/06	Latin capital letter A with E	Æ
09/15	10/04	Euro sign	€

**Figure 12-101. Latin9EBCDIC and Latin9ISO Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	11/05	Micro sign	μ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	10/01	Inverted exclamation mark	¡
10/11	11/15	Inverted question mark	¿
10/12	13/00	Latin capital letter eth (Icelandic)	Ð
10/13	13/13	Latin capital letter Y with acute accent	Ý
10/14	13/14	Latin capital letter thorn (Icelandic)	Þ
10/15	10/14	Registered trademark sign	®

EBCDIC	ISO	ISO Name	SYM
11/00	10/02	Cent sign	¢
11/01	10/03	Pound sign	£
11/02	10/05	Yen sign	¥
11/03	11/07	Middle dot	•
11/04	10/09	Copyright sign	©
11/05	10/07	Paragraph sign	§
11/06	11/06	Pilcrow sign	¶
11/07	11/12	Latin capital ligature Œ	Œ
11/08	11/13	Latin small ligature œ	œ
11/09	11/14	Latin capital letter Y with diaeresis	ÿ
11/10	10/12	Not sign	¬
11/11	10/06	Latin capital letter S with caron	Š
11/12	10/15	Macron	—
11/13	10/08	Latin small letter s with caron	š
11/14	11/04	Latin capital letter Z with caron	Ž
11/15	13/07	Multiplication sign	×

**Figure 12-102. Latin9EBCDIC and Latin9ISO Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hyphen	SHY
12/11	15/04	Latin small letter o with circumflex accent	ô
12/12	15/06	Latin small letter o with diaeresis	ö
12/13	15/02	Latin small letter o with grave accent	ò
12/14	15/03	Latin small letter o with acute accent	ó
12/15	15/05	Latin small letter o with tilde	õ

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	11/09	Super script I	ı
13/11	15/11	Latin small letter u with circumflex	û
13/12	15/12	Latin small letter u with diaeresis	ü
13/13	15/09	Latin small letter u with grave accent	ù
13/14	15/10	Latin small letter u with acute accent	ú
13/15	15/15	Latin small letter y with diaeresis	ÿ

Figure 12-103. Latin9EBCDIC and Latin9ISO Column Listing Columns 12 and 13

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	15/07	Division sign	÷
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	11/02	Superscript 2	²
14/11	13/04	Latin capital letter O with circumflex accent	Ô
14/12	13/06	Latin capital letter O with diaeresis	Ö
14/13	13/02	Latin capital letter O with grave accent	Ò
14/14	13/03	Latin capital letter O with acute accent	Ó
14/15	13/05	Latin capital letter O with tilde	Õ

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	11/03	Superscript 3	³
15/11	13/11	Latin capital letter U with circumflex accent	Û
15/12	13/12	Latin capital letter U with diaeresis	Ü
15/13	13/09	Latin capital letter U with grave accent	Ù
15/14	13/10	Latin capital letter U with acute accent	Ú
15/15	07/15	Delete	DEL

**Figure 12-104. Latin9EBCDIC and Latin9ISO Column Listing
Columns 14 and 15**

LatinCyrillicEBCDIC (8CS55.01) and LatinCyrillicISO (8CS55.00) Coded Character Sets

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ђ	ѓ	є	ѕ	Ђ	{	}	\	0
01	SOH	DC1		PU1	NBSP	№	/	ё	а	ј	~	Ё	А	Ј	§	1
02	STX	DC2	BPH	SYN	а	и	р	ш	б	к	ѕ	Т	В	К	Ѕ	2
03	ETX	DC3	NBH	STS	б	й	с	щ	с	l	t	у	С	Л	Т	3
04	ST	OSC	IND	CCH	в	к	т	ь	d	m	u	Ф	D	M	U	4
05	HT	NEL	LF	MW	г	л	у	ы	e	n	v	Х	E	N	V	5
06	SSA	BS	ETB	SPA	д	м	ф	ь	f	o	w	Ц	F	O	W	6
07	APC	ESA	ESC	EOT	е	н	х	э	g	p	x	Ч	G	P	X	7
08	EPA	CAN	HTS	SOS	ж	о	ц	ю	h	q	y	Ш	H	Q	Y	8
09	RI	EM	HTJ		з	п	ч	`	i	r	z	Щ	I	R	Z	9
10	SS2	PU2	VTS	SCI	[]		:	А	Ж	М	Ь	SHY	ћ	Ј	Ц
11	VT	SS3	PLD	CSI	.	\$,	#	Б	З	Н	Ы	i	ќ	Љ	Ѓ
12	FF	IS4	PLU	DC4	<	*	%	@	В	И	О	Ь	ï	ÿ	Њ	Є
13	CR	IS3	ENQ	NAK	()	_	'	Г	И	П	Э	j	и	ћ	Ѕ
14		IS2	ACK	PM	+	;	>	=	Д	К	Р	Ю	љ	І	Ќ	Я
15		IS1	BEL	SUB	!	^	?	"	Е	Л	С	Я	њ	Ї	Ў	DEL

Figure 12-105. LatinCyrillicEBCDIC Code Table (8CS55.01)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	А	Р	а	р	№
01	SOH	DC1	!	1	A	Q	a	q		PU1	Ё	Б	С	б	с	ё
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	Ђ	В	Т	в	т	ђ
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	Ѓ	Г	У	г	у	ѓ
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	Є	Д	Ф	д	ф	є
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW	Ѕ	Е	Х	е	х	ѕ
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA	І	Ж	Ц	ж	ц	і
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA	Ї	З	Ч	з	ч	ї
08	BS	CAN	(8	H	X	h	x	HTS	SOS	Ј	И	Ш	и	ш	ј
09	HT	EM)	9	I	Y	i	y	HTJ		Љ	И	Щ	й	щ	љ
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI	Њ	К	Ъ	к	ъ	њ
11	VT	ESC	+	;	K	[k	{	PLD	CSI	Ћ	Л	Ы	л	ы	ћ
12	FF	IS4	,	<	L	\	l		PLU	ST	Ќ	М	Ь	м	ь	ќ
13	CR	IS3	-	=	M]	m	}	RI	OSC	SHY	Н	Э	н	э	§
14	SO	IS2	.	>	N	^	n	~	SS2	PM	Ў	О	Ю	о	ю	ў
15	SI	IS1	/	?	O	_	o	DEL	SS3	APC	Џ	П	Я	п	я	џ

Figure 12-106. LatinCyrillicISO Code Table (8CS55.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	09/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Locking-shift 0	LS0
00/15	00/15	Locking-shift 1	LS1

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

Figure 12-107. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 0 and 1

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTB
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledge	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substitute character	SUB

Figure 12-108. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 2 and 3

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	13/00	Cyrillic small letter a	а
04/03	13/01	Cyrillic small letter be	б
04/04	13/02	Cyrillic small letter ye	в
04/05	13/03	Cyrillic small letter ghe	г
04/06	13/04	Cyrillic small letter de	д
04/07	13/05	Cyrillic small letter ie	е
04/08	13/06	Cyrillic small letter zhe	ж
04/09	13/07	Cyrillic small letter ze	з
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	15/00	Number acronym	№
05/02	13/08	Cyrillic small letter i	и
05/03	13/09	Cyrillic small letter short i	й
05/04	13/10	Cyrillic small letter ka	к
05/05	13/11	Cyrillic small letter ei	л
05/06	13/12	Cyrillic small letter em	м
05/07	13/13	Cyrillic small letter en	н
05/08	13/14	Cyrillic small letter o	о
05/09	13/15	Cyrillic small letter pe	п
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

Figure 12-109. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 4 and 5

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	14/00	Cyrillic small letter er	р
06/03	14/01	Cyrillic small letter es	с
06/04	14/02	Cyrillic small letter te	т
06/05	14/03	Cyrillic small letter u	у
06/06	14/04	Cyrillic small letter ef	ф
06/07	14/05	Cyrillic small letter ha	х
06/08	14/06	Cyrillic small letter tse	ц
06/09	14/07	Cyrillic small letter che	ч
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	_
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	15/02	Cyrillic small letter Serbo-Croatian dje	ђ
07/01	15/01	Cyrillic small letter io	ѐ
07/02	14/08	Cyrillic small letter sha	ш
07/03	14/09	Cyrillic small letter shcha	щ
07/04	14/10	Cyrillic small letter hard sign	ъ
07/05	14/11	Cyrillic small letter yeru	ѣ
07/06	14/12	Cyrillic small letter soft sign	ь
07/07	14/13	Cyrillic small letter e	ѐ
07/08	14/14	Cyrillic small letter yu	ю
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

**Figure 12-110. LatinCyrillicEBC and LatinCyrillicISO Column Listing
Columns 6 and 7**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00	15/03	Cyrillic small letter Macedonian gje	ѓ
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	11/00	Cyrillic capital letter A	А
08/11	11/01	Cyrillic capital letter GE	Б
08/12	11/02	Cyrillic capital letter VE	В
08/13	11/03	Cyrillic capital letter GHE	Г
08/14	11/04	Cyrillic capital letter DE	Д
08/15	11/05	Cyrillic capital letter IE	Е

EBCDIC	ISO	ISO Name	SYM
09/00	15/04	Cyrillic small letter Ukrainian ie	є
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	11/06	Cyrillic capital letter ZHE	Ж
09/11	11/07	Cyrillic capital letter ZE	З
09/12	11/08	Cyrillic capital letter I	И
09/13	11/09	Cyrillic capital short I	Й
09/14	11/10	Cyrillic capital letter KA	К
09/15	11/11	Cyrillic capital letter EL	Л

Figure 12-111. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 8 and 9

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	15/05	Cyrillic small letter Macedonian dze	ѕ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	11/12	Cyrillic capital letter EM	М
10/11	11/13	Cyrillic capital letter EN	Н
10/12	11/14	Cyrillic capital letter O	О
10/13	11/15	Cyrillic capital letter PE	П
10/14	12/00	Cyrillic capital letter ER	Р
10/15	12/01	Cyrillic capital letter ES	С

EBCDIC	ISO	ISO Name	SYM
11/00	10/02	Cyrillic capital letter Serbo-Croatian DJE	Ђ
11/01	10/01	Cyrillic capital letter IO	Ё
11/02	12/02	Cyrillic capital letter TE	Т
11/03	12/03	Cyrillic capital letter U	У
11/04	12/04	Cyrillic capital letter EF	Ф
11/05	12/05	Cyrillic capital letter HA	Х
11/06	12/06	Cyrillic capital letter TSE	Ц
11/07	12/07	Cyrillic capital letter CHE	Ч
11/08	12/08	Cyrillic capital letter SHA	Ш
11/09	12/09	Cyrillic capital letter SHCHA	Щ
11/10	12/10	Cyrillic capital letter HARD SIGN	Ъ
11/11	12/11	Cyrillic capital letter YERU	Ы
11/12	12/12	Cyrillic capital letter SOFT SIGN	Ь
11/13	12/13	Cyrillic capital letter E	Э
11/14	12/14	Cyrillic capital letter YU	Ю
11/15	12/15	Cyrillic capital letter YA	Я

**Figure 12-112. LatinCyrillicEBC and LatinCyrillicISO Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	10/13	Soft hypen	SHY
12/11	15/06	Cyrillic small letter Byelorussian-Ukrainian i	i
12/12	15/07	Cyrillic small letter Ukrainian yi	ï
12/13	15/08	Cyrillic small letter je	j
12/14	15/09	Cyrillic small letter lje	Љ
12/15	15/10	Cyrillic small letter nje	Њ

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	15/11	Cyrillic small letter Serbo-Croatian	Ѡ
13/11	15/12	Cyrillic small letter Macedonian kje	ќ
13/12	15/14	Cyrillic small letter Byelorussian short u	ў
13/13	15/15	Cyrillic small letter dzhe	Ѣ
13/14	10/06	Cyrillic capital letter byelorussian-Ukrainian I	І
13/15	10/07	Cyrillic capital letter Ukrainian YI	Й

Figure 12-113. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 12 and 13

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01	15/13	Paragraph sign	§
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	10/08	Cyrillic capital letter JE	Ј
14/11	10/09	Cyrillic capital letter LJE	Љ
14/12	10/10	Cyrillic capital letter NJE	Њ
14/13	10/11	Cyrillic capital letter Serbo-Croatian TSHE	Ћ
14/14	10/12	Cyrillic capital letter Macedonian KJE	Ќ
14/15	10/14	Cyrillic capital letter Byelorussian short U	Ў

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	10/15	Cyrillic capital letter DZHE	Ѣ
15/11	10/03	Cyrillic capital letter Macedonian GJE	Ѓ
15/12	10/04	Cyrillic capital letter Ukrainian IE	Є
15/13	10/05	Cyrillic capital letter Macedonian DZE	Ѕ
15/14	14/15	Cyrillic small letter ya	я
15/15	07/15	Delete	DEL

Figure 12-114. LatinCyrillicEBC and LatinCyrillicISO Column Listing Columns 14 and 15

LatinGreekBTOS Coded Character Set (8CS41.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	¼	SP	0	@	P	`	p		—	NBSP	°	í	Π	ύ	π
01	↑	†	!	1	A	Q	a	q		+	'	±	A	P	α	ρ
02	▲	→	"	2	B	R	b	r		⊥	'	²	B		β	ς
03	♣	⊥	#	3	C	S	c	s		⊥	£	³	Γ	Σ	γ	σ
04	■	▪	\$	4	D	T	d	t				'	Δ	T	δ	τ
05	□	÷	%	5	E	U	e	u		⊥		“	E	Υ	ε	υ
06	½		&	6	F	V	f	v		⊥	:	‘A	Z	Φ	ζ	φ
07	⚖	§	'	7	G	W	g	w		⊥	§	•	H	X	η	χ
08	⊥	≠	(8	H	X	h	x		⊥	“	‘E	Θ	Ψ	θ	ψ
09	⊥	≈)	9	I	Y	i	y		⊥	©	‘H	I	Ω	ι	ω
10	↙	¶	*	:	J	Z	j	z		⊥		‘I	K	Ï	κ	ï
11	↓	•	+	;	K	[k	{		=	«	»	Λ	ÿ	λ	ÿ
12	↘	¬	,	<	L	\	l			—	¬	‘O	M	ά	μ	ό
13	▲	≤	—	=	M]	m	}			SHY	½	N	έ	ν	ύ
14	←	±	.	>	N	^	n	~				‘Y	Ξ	ή	ξ	ώ
15	‡	≥	/	?	O	_	o	■		⋮	-	Ω	O	ί	ο	MESI

Figure 12-115. LatinGreekBTOS Code Table (8CS41.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00	Null	NUL
00/01	Upward arrow	↑
00/02	Lower left corner triangle solid	▴
00/03	Cent sign	¢
00/04	Square solid	■
00/05	Square empty	□
00/06	Vulgar fraction one half	½
00/07	Bell sign	🔔
00/08	Left tab sign	⌠
00/09	Right tab sign	⌡
00/10	Left return sign	↶
00/11	Downward arrow	↓
00/12	Right return sign	↷
00/13	Lower right corner triangle solid	▾
00/14	Leftward arrow	←
00/15	Double dagger	‡

COL/ROW	ISO Name	SYM
01/00	Vulgar fraction one quarter	¼
01/01	Dagger	†
01/02	Rightward arrow	→
01/03	Space symbol	␣
01/04	Box drawing solid small square	◼
01/05	Division sign	÷
01/06	Vertical line	
01/07	Paragraph sign	§
01/08	Not equal-to sign	≠
01/09	Asymptotically equal sign	≈
01/10	Pilcrow sign	¶
01/11	Middle dot	•
01/12	Not sign	¬
01/13	Less-than or equal-to sign	≤
01/14	Plus-minus sign	±
01/15	Greater-than or equal-to sign	≥

**Figure 12-116. LatinGreekBTOS Column Listing
Columns 0 and 1**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 12-117. LatinGreekBTOS Column Listing
Columns 2 and 3**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 12-118. LatinGreekBTOS Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

COL/ROW	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Box drawing solid upper half block	▀

Figure 12-119. LatinGreekBTOS Column Listing Columns 6 and 7

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00		
08/01		
08/02		
08/03		
08/04		
08/05		
08/06		
08/07		
08/08		
08/09		
08/10		
08/11		
08/12		
08/13		
08/14		
08/15		

COL/ROW	ISO Name	
09/00	Box drawing light horizontal	—
09/01	Box drawing light horizontal and vertical	+
09/02	Box drawing light horizontal and down	┴
09/03	Box drawing light horizontal and up	┬
09/04	Box drawing light vertical	
09/05	Box drawing light vertical and right	┤
09/06	Box drawing light vertical and left	├
09/07	Box drawing light up and right	┐
09/08	Box drawing light up and left	┑
09/09	Box drawing light down and right	┒
09/10	Box drawing light down and left	┓
09/11	Box drawing double horizontal	==
09/12	Box drawing heavy horizontal	—
09/13	Box drawing double vertical	
09/14	Box drawing heavy vertical	!
09/15	Box drawing light shade	⋮

**Figure 12–120. LatinGreekBTOS Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	No-break space	NBSP
10/01	Left single quotation mark	‘
10/02	Right single quotation mark	’
10/03	Pound sign	£
10/04		
10/05		
10/06	Broken bar	∣
10/07	Paragraph sign	§
10/08	Diaeresis	¨
10/09	Copyright sign	©
10/10		
10/11	Left-pointing double angle quotation mark	«
10/12	Not sign	¬
10/13	Soft hyphen	SHY
10/14		
10/15	Horizontal bar	–

COL/ROW	ISO Name	
11/00	Degree sign	°
11/01	Plus-minus sign	±
11/02	Superscript 2	²
11/03	Superscript 3	³
11/04	Greek accent	´
11/05	Diaeresis and Greek accent	¨
11/06	Greek capital letter alpha with Greek accent	Ἄ
11/07	Middle dot	•
11/08	Greek capital letter epsilon with Greek accent	Ἐ
11/09	Greek capital letter eta with Greek accent	Ἠ
11/10	Greek capital letter iota with Greek accent	Ἰ
11/11	Right-pointing double angle quotation mark	»
11/12	Greek capital letter omicron with Greek accent	Ὀ
11/13	Vulgar fraction one half	½
11/14	Greek capital letter upsilon with Greek accent	Ἦ
11/15	Greek capital letter omega with Greek accent	Ὠ

**Figure 12–121. LatinGreekBTOS Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Greek small letter iota with diaeresis and Greek accent	í
12/01	Greek capital letter alpha	Α
12/02	Greek capital letter beta	Β
12/03	Greek capital letter gamma	Γ
12/04	Greek capital letter delta	Δ
12/05	Greek capital letter epsilon	Ε
12/06	Greek capital letter zeta	Ζ
12/07	Greek capital letter eta	Η
12/08	Greek capital letter theta	Θ
12/09	Greek capital letter iota	Ι
12/10	Greek capital letter kappa	Κ
12/11	Greek capital letter lamda	Λ
12/12	Greek capital letter mu	Μ
12/13	Greek capital letter nu	Ν
12/14	Greek capital letter xi	Ξ
12/15	Greek capital letter omicron	Ο

COL/ROW	ISO Name	
13/00	Greek capital letter pi	Π
13/01	Greek capital letter rho	Ρ
13/02		
13/03	Greek capital letter sigma	Σ
13/04	Greek capital letter tau	Τ
13/05	Greek capital letter upsilon	Υ
13/06	Greek capital letter phi	Φ
13/07	Greek capital letter chi	Χ
13/08	Greek capital letter psi	Ψ
13/09	Greek capital letter omega	Ω
13/10	Greek capital letter iota with diaeresis	ϊ
13/11	Greek capital letter upsilon with diaeresis	ϋ
13/12	Greek small letter alpha with Greek accent	ά
13/13	Greek small letter epsilon with Greek accent	έ
13/14	Greek small letter eta with Greek accent	ή
13/15	Greek small letter iota with Greek accent	ί

**Figure 12-122. LatinGreekBTOS Column Listing
Columns 12 and 13**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Greek small letter upsilon with diaeresis and Greek accent	ύ
14/01	Greek small letter alpha	α
14/02	Greek small letter beta	β
14/03	Greek small letter gamma	γ
14/04	Greek small letter delta	δ
14/05	Greek small letter epsilon	ε
14/06	Greek small letter zeta	ζ
14/07	Greek small letter eta	η
14/08	Greek small letter theta	θ
14/09	Greek small letter iota	ι
14/10	Greek small letter kappa	κ
14/11	Greek small letter lamda	λ
14/12	Greek small letter mu	μ
14/13	Greek small letter nu	ν
14/14	Greek small letter xi	ξ
14/15	Greek small letter omicron	ο

COL/ROW	ISO Name	SYM
15/00	Greek small letter pi	π
15/01	Greek small letter rho	ρ
15/02	Greek small letter final sigma	ς
15/03	Greek small letter sigma	σ
15/04	Greek small letter tau	τ
15/05	Greek small letter upsilon	υ
15/06	Greek small letter phi	φ
15/07	Greek small letter chi	χ
15/08	Greek small letter psi	ψ
15/09	Greek small letter omega	ω
15/10	Greek small letter iota with diaeresis	ϊ
15/11	Greek small letter upsilon with diaeresis	ϋ
15/12	Greek small letter omicron with Greek accent	ό
15/13	Greek small letter upsilon with Greek accent	ύ
15/14	Greek small letter omega with Greek accent	ώ
15/15	Multibyte escape sequence introducer	MESI

**Figure 12-123. LatinGreekBTOS Column Listing
Columns 14 and 15**

LatinGreekEBCDIC (8CS30.01) and LatinGreekISO (8CS30.00) Coded Character Sets

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE		DCS	SP	&	-	ı	Γ	K	P	Ψ	{	}	\	0
01	SOH	DC1		PU1	NBSP	©	/	»	a	j	~	Ω	A	J		1
02	STX	DC2	BPH	SYN	`		²	ˆ	b	k	s	İ	B	K	S	2
03	ETX	DC3	NBH	STS	'	«	³	½	c	l	t	ÿ	C	L	T	3
04	ST	OSC	IND	CCH	£	¬	'	Υ	d	m	u	ά	D	M	U	4
05	HT	NEL	LF	MW	€	SHY	ˆ	Ω	e	n	v	έ	E	N	V	5
06	SSA	BS	ETB	SPA			À	í	f	o	w	ή	F	O	W	6
07	APC	ESA	ESC	EOT		-	•	A	g	p	x	í	G	P	X	7
08	EPA	CAN	HTS	SOS	§	°	È	B	h	q	y	ύ	H	Q	Y	8
09	RI	EM	HTJ		¨	±	ˆ	ı	r	z	α	ı	R	Z		9
10	SS2	PU2	VTS	SCI	[]		:	Δ	Λ		β	θ	ξ	τ	ı
11	VT	SS3	PLD	CSI	.	\$,	#	E	M	Σ	γ	ι	ο	υ	ü
12	FF	IS4	PLU	DC4	<	*	%	@	Z	N	T	δ	κ	π	φ	ó
13	CR	IS3	ENQ	NAK	()	_	'	H	Ξ	Υ	ε	λ	ρ	χ	ύ
14		IS2	ACK	PM	+	;	>	=	Θ	O	Φ	ζ	μ	ς	Ψ	ώ
15		IS1	BEL	SUB	!	^	?	"	I	Π	X	η	ν	σ	ω	DEL

Figure 12-124. LatinGreekEBCDIC Code Table (8CS30.01)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE	SP	0	@	P	`	p		DCS	NBSP	°	í	Π	ύ	π
01	SOH	DC1	!	1	A	Q	a	q		PU1	`	±	A	P	α	ρ
02	STX	DC2	"	2	B	R	b	r	BPH	PU2	'	²	B		β	ς
03	ETX	DC3	#	3	C	S	c	s	NBH	STS	£	³	Γ	Σ	γ	σ
04	EOT	DC4	\$	4	D	T	d	t	IND	CCH	€	'	Δ	T	δ	τ
05	ENQ	NAK	%	5	E	U	e	u	NEL	MW		“	E	Υ	ε	υ
06	ACK	SYN	&	6	F	V	f	v	SSA	SPA		’	Z	Φ	ζ	φ
07	BEL	ETB	'	7	G	W	g	w	ESA	EPA	§	•	H	X	η	χ
08	BS	CAN	(8	H	X	h	x	HTS	SOS	..	’	Θ	Ψ	θ	ψ
09	HT	EM)	9	I	Y	i	y	HTJ		©	’	I	Ω	ι	ω
10	LF	SUB	*	:	J	Z	j	z	VTS	SCI		’	K	’	κ	ï
11	VT	ESC	+	;	K	[k	{	PLD	CSI	«	»	Λ	Ÿ	λ	ü
12	FF	IS4	,	<	L	\	l		PLU	ST	–	’	M	ά	μ	ό
13	CR	IS3	-	=	M]	m	}	RI	OSC	SHY	½	N	έ	ν	ύ
14	LSO	IS2	.	>	N	^	n	~	SS2	PM		Υ	Ξ	ή	ξ	ώ
15	LS1	IS1	/	?	O	_	o	DEL	SS3	APC	-	Ω	O	ί	ο	

Figure 12-125. LatinGreekISO Code Table (8CS30.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
00/00	00/00	Null	NUL
00/01	00/01	Start of heading	SOH
00/02	00/02	Start of text	STX
00/03	00/03	End of text	ETX
00/04	09/12	String terminator	ST
00/05	00/09	Horizontal tabulation	HT
00/06	08/06	Start of selected area	SSA
00/07	07/15	Application program command	APC
00/08	09/07	End of guarded protected area	EPA
00/09	08/13	Reverse index	RI
00/10	08/14	Single shift 2	SS2
00/11	00/11	Vertical tabulation	VT
00/12	00/12	Form feed	FF
00/13	00/13	Carriage return	CR
00/14	00/14	Locking-shift 0	LS0
00/15	00/15	Locking-shift 1	LS1

EBCDIC	ISO	ISO Name	SYM
01/00	01/00	Data link escape	DLE
01/01	01/01	Device control 1	DC1
01/02	01/02	Device control 2	DC2
01/03	01/03	Device control 3	DC3
01/04	09/13	Operating system command	OSC
01/05	08/05	Next line	NEL
01/06	00/08	Backspace	BS
01/07	08/07	End of selected area	ESA
01/08	01/08	Cancel	CAN
01/09	01/09	End medium	EM
01/10	09/02	Private use 2	PU2
01/11	08/15	Single shift 3	SS3
01/12	01/12	Information separator 4	IS4
01/13	01/13	Information separator 3	IS3
01/14	01/14	Information separator 2	IS2
01/15	01/15	Information separator 1	IS1

Figure 12-126. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 0 and 1

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
02/00	08/00		
02/01	08/01		
02/02	08/02	Break permitted here	BPH
02/03	08/03	No break here	NBH
02/04	008/04	Index	IND
02/05	00/10	Line feed	LF
02/06	01/07	End-of-transmission block	ETB
02/07	01/11	Escape	ESC
02/08	08/08	Horizontal tabulation set	HTS
02/09	08/09	Horizontal tabulation with justification	HTJ
02/10	08/10	Vertical tabulation set	VTS
02/11	08/11	Partial line down	PLD
02/12	08/12	Partial line up	PLU
02/13	00/05	Enquiry	ENQ
02/14	00/06	Acknowledgement	ACK
02/15	00/07	Bell	BEL

EBCDIC	ISO	ISO Name	SYM
03/00	09/00	Device control string	DCS
03/01	09/01	Private use 1	PU1
03/02	01/06	Synchronous idle	SYN
03/03	09/03	Set transmission state	STS
03/04	09/04	Cancel character	CCH
03/05	09/05	Message waiting	MW
03/06	09/06	Start of guarded protected area	SPA
03/07	00/04	End of transmission	EOT
03/08	09/08	Start of string	SOS
03/09	09/09		
03/10	09/10	Single character introducer	SCI
03/11	09/11	Control sequence introducer	CSI
03/12	01/04	Device control 4	DC4
03/13	01/05	Negative acknowledge	NAK
03/14	09/14	Privacy message	PM
03/15	01/10	Substituted character	SUB

Figure 12-127. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 2 and 3

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
04/00	02/00	Space	SP
04/01	10/00	No-break space	NBSP
04/02	10/01	Left single quotation mark	`
04/03	10/02	Right single quotation mark	'
04/04	10/03	Pound sign	£
04/05	10/04	Euro sign	€
04/06	10/05		
04/07	10/06	Broken bar	
04/08	10/07	Paragraph sign	§
04/09	10/08	Diaeresis	¨
04/10	05/11	Left square bracket	[
04/11	02/14	Full stop	.
04/12	03/12	Less-than sign	<
04/13	02/08	Left parenthesis	(
04/14	02/11	Plus sign	+
04/15	02/01	Exclamation mark	!

EBCDIC	ISO	ISO Name	SYM
05/00	02/06	Ampersand	&
05/01	10/09	Copyright sign	©
05/02	10/10		
05/03	10/11	Left-pointing double angle quotation mark	«
05/04	10/12	Not sign	¬
05/05	10/13	Soft hyphen	SHY
05/06	10/14		
05/07	10/15	Horizontal bar	-
05/08	11/00	Degree sign	°
05/09	11/01	Plus-minus sign	±
05/10	05/13	Right square bracket]
05/11	02/04	Dollar sign	\$
05/12	02/10	Asterisk	*
05/13	02/09	Right parenthesis)
05/14	03/11	Semicolon	;
05/15	05/14	Circumflex accent	^

**Figure 12-128. LatinGreekEBCDIC and LatinGreekISO Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
06/00	02/13	Hyphen-minus	-
06/01	02/15	Solidus	/
06/02	11/02	Superscript 2	²
06/03	11/03	Superscript 3	³
06/04	11/04	Greek accent	´
06/05	11/05	Diaeresis and Greek accent	¨
06/06	11/06	Greek capital letter alpha with Greek accent	Ἄ
06/07	11/07	Middle dot	•
06/08	11/08	Greek capital letter epsilon with Greek accent	Ἐ
06/09	11/09	Greek capital letter eta with Greek accent	Ἠ
06/10	07/12	Vertical line	
06/11	02/12	Comma	,
06/12	02/05	Percent sign	%
06/13	05/15	Low line	—
06/14	03/14	Greater-than sign	>
06/15	03/15	Question mark	?

EBCDIC	ISO	ISO Name	SYM
07/00	11/10	Greek capital letter iota with Greek accent	Ἰ
07/01	11/11	Right-pointing double angle quotation mark	»
07/02	11/12	Greek capital letter omicron with Greek accent	Ὀ
07/03	11/13	Vulgar fraction one half	½
07/04	11/14	Greek capital letter upsilon with Greek accent	Ἦ
07/05	11/15	Greek capital letter omega with Greek accent	Ὠ
07/06	12/00	Greek small letter iota with diaeresis and Greek accent	ϊ
07/07	12/01	Greek capital letter alpha	Α
07/08	12/02	Greek capital letter beta	Β
07/09	06/00	Grave accent	`
07/10	03/10	Colon	:
07/11	02/03	Number sign	#
07/12	04/00	Commercial at	@
07/13	02/07	Apostrophe	'
07/14	03/13	Equal sign	=
07/15	02/02	Quotation mark	"

Figure 12–129. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 6 and 7

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
08/00	12/03	Greek capital letter gamma	Γ
08/01	06/01	Latin small letter a	a
08/02	06/02	Latin small letter b	b
08/03	06/03	Latin small letter c	c
08/04	06/04	Latin small letter d	d
08/05	06/05	Latin small letter e	e
08/06	06/06	Latin small letter f	f
08/07	06/07	Latin small letter g	g
08/08	06/08	Latin small letter h	h
08/09	06/09	Latin small letter i	i
08/10	12/04	Greek capital letter delta	Δ
08/11	12/05	Greek capital letter epsilon	Ε
08/12	12/06	Greek capital letter zeta	Ζ
08/13	12/07	Greek capital letter eta	Η
08/14	12/08	Greek capital letter theta	Θ
08/15	12/09	Greek capital letter iota	Ι

EBCDIC	ISO	ISO Name	SYM
09/00	12/10	Greek capital letter kappa	Κ
09/01	06/10	Latin small letter j	j
09/02	06/11	Latin small letter k	k
09/03	06/12	Latin small letter l	l
09/04	06/13	Latin small letter m	m
09/05	06/14	Latin small letter n	n
09/06	06/15	Latin small letter o	o
09/07	07/00	Latin small letter p	p
09/08	07/01	Latin small letter q	q
09/09	07/02	Latin small letter r	r
09/10	12/11	Greek capital letter lamda	Λ
09/11	12/12	Greek capital letter mu	Μ
09/12	12/13	Greek capital letter nu	Ν
09/13	12/14	Greek capital letter xi	Ξ
09/14	12/15	Greek capital letter omicron	Ο
09/15	13/00	Greek capital letter pi	Π

**Figure 12-130. LatinGreekEBCDIC and LatinGreekISO Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
10/00	13/01	Greek capital letter rho	Ρ
10/01	07/14	Tilde	~
10/02	07/03	Latin small letter s	s
10/03	07/04	Latin small letter t	t
10/04	07/05	Latin small letter u	u
10/05	07/06	Latin small letter v	v
10/06	07/07	Latin small letter w	w
10/07	07/08	Latin small letter x	x
10/08	07/09	Latin small letter y	y
10/09	07/10	Latin small letter z	z
10/10	13/02		
10/11	13/03	Greek capital letter sigma	Σ
10/12	13/04	Greek capital letter tau	Τ
10/13	13/05	Greek capital letter epsilon	Υ
10/14	13/06	Greek capital letter phi	Φ
10/15	13/07	Greek capital letter chi	Χ

EBCDIC	ISO	ISO Name	SYM
11/00	13/08	Greek capital letter psi	Ψ
11/01	13/09	Greek capital letter omega	Ω
11/02	13/10	Greek capital letter iota with diaeresis	Ï
11/03	13/11	Greek capital letter upsilon with diaeresis	Û
11/04	13/12	Greek small letter alpha with Greek accent	ά
11/05	13/13	Greek small letter epsilon with Greek accent	έ
11/06	13/14	Greek small letter eta with Greek accent	ή
11/07	13/15	Greek small letter iota with Greek accent	ί
11/08	14/00	Greek small letter upsilon with diaeresis and Greek accent	ύ
11/09	14/01	Greek small letter alpha	α
11/10	14/02	Greek small letter beta	β
11/11	14/03	Greek small letter gamma	γ
11/12	14/04	Greek small letter delta	δ
11/13	14/05	Greek small letter epsilon	ε
11/14	14/06	Greek small letter zeta	ζ
11/15	14/07	Greek small letter eta	η

Figure 12-131. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 10 and 11

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
12/00	07/11	Left curly bracket	{
12/01	04/01	Latin capital letter A	A
12/02	04/02	Latin capital letter B	B
12/03	04/03	Latin capital letter C	C
12/04	04/04	Latin capital letter D	D
12/05	04/05	Latin capital letter E	E
12/06	04/06	Latin capital letter F	F
12/07	04/07	Latin capital letter G	G
12/08	04/08	Latin capital letter H	H
12/09	04/09	Latin capital letter I	I
12/10	14/08	Greek small letter theta	θ
12/11	14/09	Greek small letter iota	ι
12/12	14/10	Greek small letter kappa	κ
12/13	14/11	Greek small letter lamda	λ
12/14	14/12	Greek small letter mu	μ
12/15	14/13	Greek small letter nu	ν

EBCDIC	ISO	ISO Name	SYM
13/00	07/13	Right curly bracket	}
13/01	04/10	Latin capital letter J	J
13/02	04/11	Latin capital letter K	K
13/03	04/12	Latin capital letter L	L
13/04	04/13	Latin capital letter M	M
13/05	04/14	Latin capital letter N	N
13/06	04/15	Latin capital letter O	O
13/07	05/00	Latin capital letter P	P
13/08	05/01	Latin capital letter Q	Q
13/09	05/02	Latin capital letter R	R
13/10	14/14	Greek small letter xi	ξ
13/11	14/15	Greek small letter omicron	ο
13/12	15/00	Greek small letter pi	π
13/13	15/01	Greek small letter rho	ρ
13/14	15/02	Greek small letter final sigma	ς
13/15	15/03	Greek small letter sigma	σ

**Figure 12-132. LatinGreekEBCDIC and LatinGreekISO Column Listing
Columns 12 and 13**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

EBCDIC	ISO	ISO Name	SYM
14/00	05/12	Reverse solidus	\
14/01			
14/02	05/03	Latin capital letter S	S
14/03	05/04	Latin capital letter T	T
14/04	05/05	Latin capital letter U	U
14/05	05/06	Latin capital letter V	V
14/06	05/07	Latin capital letter W	W
14/07	05/08	Latin capital letter X	X
14/08	05/09	Latin capital letter Y	Y
14/09	05/10	Latin capital letter Z	Z
14/10	15/04	Greek small letter tau	τ
14/11	15/05	Greek small letter upsilon	υ
14/12	15/06	Greek small letter phi	φ
14/13	15/07	Greek small letter chi	χ
14/14	15/08	Greek small letter psi	ψ
14/15	15/09	Greek small letter omega	ω

EBCDIC	ISO	ISO Name	SYM
15/00	03/00	Digit 0	0
15/01	03/01	Digit 1	1
15/02	03/02	Digit 2	2
15/03	03/03	Digit 3	3
15/04	03/04	Digit 4	4
15/05	03/05	Digit 5	5
15/06	03/06	Digit 6	6
15/07	03/07	Digit 7	7
15/08	03/08	Digit 8	8
15/09	03/09	Digit 9	9
15/10	15/10	Greek small letter iota with diaeresis	ï
15/11	15/11	Greek small letter upsilon with diaeresis	ÿ
15/12	15/12	Greek small letter omicron with Greek accent	ó
15/13	15/13	Greek small letter upsilon with Greek accent	ύ
15/14	15/14	Greek small letter omega with Greek accent	ώ
15/15	07/15	Delete	DEL

Figure 12-133. LatinGreekEBCDIC and LatinGreekISO Column Listing Columns 14 and 15

LocalEBCDIC Coded Character Set (8500.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	DLE			SP	&	-					{	}	\	0	
01	SOH	DC1					/		a	j	~		A	J	APC	1
02	STX	DC2		SYN					b	k	s		B	K	S	2
03	ETX	DC3							c	l	t		C	L	T	3
04									d	m	u		D	M	U	4
05	HT		LF						e	n	v		E	N	V	5
06		BS	ETB						f	o	w		F	O	W	6
07	DEL		ESC	EOT					g	p	x		G	P	X	7
08		CAN							h	q	y		H	Q	Y	8
09		EM						`	i	r	z		I	R	Z	9
10					[]		:								
11	VT		SDO		.	\$,	#								
12	FF	FS	EDO	DC4	<	*	%	@								
13	CR	GS	ENQ	NAK	()	-	'								
14	SO	RS	ACK		+	;	>	=								
15	SI	US	BEL	SUB	!	^	?	"								MD

Figure 12-134. LocalEBCDIC Code Table (8500.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
00/00	Null	NUL
00/01	Start of heading	SO H
00/02	Start text	STX
00/03	End text	ETX
00/04		
00/05	Horizontal tabulation	HT
00/06		
00/07	Delete	DEL
00/08		
00/09		
00/10		
00/11	Vertical tabulation	VT
00/12	Form feed	FF
00/13	Carriage return	CR
00/14	Shift out	SO
00/15	Shift in	SI

Col/Row	ISO Name	Sym
01/00	Data link escape	DLE
01/01	Device control 1	DC1
01/02	Device control 2	DC2
01/03	Device control 3	DC3
01/04		
01/05		
01/06	Backspace	BS
01/07		
01/08	Cancel	CAN
01/09	End medium	EM
01/10		
01/11		
01/12	File separator	FS
01/13	Group separator	GS
01/14	Record separator	RS
01/15	Unit separator	US

**Figure 12-135. LocalEBCDIC Column Listing
Columns 0 and 1**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
02/00		
02/01		
02/02		
02/03		
02/04		
02/05	Line feed	LF
02/06	End-of-transmission block	ETB
02/07	Escape	ESC
02/08		
02/09		
02/10		
02/11	Start double octet	SD O
02/12	End double octet	ED O
02/13	Enquiry	EN Q
02/14	Acknowledge	ACK
02/15	Bell	BEL

Col/Row	ISO Name	Sym
03/00		
03/01		
03/02	Synchronous file	SYN
03/03		
03/04		
03/05		
03/06		
03/07	End of transmission	EOT
03/08		
03/09		
03/10		
03/11		
03/12	Device control 4	DC4
03/13	Negative acknowledge	NAK
03/14		
03/15	Substituted character	SUB

**Figure 12-136. LocalEBCDIC Column Listing
Columns 2 and 3**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
04/00	Space character	SP
04/01		
04/02		
04/03		
04/04		
04/05		
04/06		
04/07		
04/08		
04/09		
04/10	Left square bracket	[
04/11	Full stop	.
04/12	Less-than sign	<
04/13	Left parenthesis	(
04/14	Plus sign	+
04/15	Exclamation mark	!

Col/Row	ISO Name	Sym
05/00	Ampersand	&
05/01		
05/02		
05/03		
05/04		
05/05		
05/06		
05/07		
05/08		
05/09		
05/10	Right square bracket]
05/11	Dollar sign	\$
05/12	Asterisk	*
05/13	Right parenthesis)
05/14	Semicolon	;
05/15	Circumflex accent	^

**Figure 12-137. LocalEBCDIC Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
06/00	Hyphen-minus	-
06/01	Solidus	/
06/02		
06/03		
06/04		
06/05		
06/06		
06/07		
06/08		
06/09		
06/10	Vertical line	
06/11	Comma	,
06/12	Percent sign	%
06/13	Low line	_
06/14	Greater-than sign	>
06/15	Question mark	?

Col/Row	ISO Name	Sym
07/00		
07/01		
07/02		
07/03		
07/04		
07/05		
07/06		
07/07		
07/08		
07/09	Grave accent	`
07/10	Colon	:
07/11	Number sign	#
07/12	Commercial at	@
07/13	Apostrophe	'
07/14	Equal sign	=
07/15	Quotation mark	"

**Figure 12-138. LocalEBCDIC Column Listing
Columns 6 and 7**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
08/00		
08/01	Latin small letter a	a
08/02	Latin small letter b	b
08/03	Latin small letter c	c
08/04	Latin small letter d	d
08/05	Latin small letter e	e
08/06	Latin small letter f	f
08/07	Latin small letter g	g
08/08	Latin small letter h	h
08/09	Latin small letter i	i
08/10		
08/11		
08/12		
08/13		
08/14		
08/15		

Col/Row	ISO Name	Sym
09/00		
09/01	Latin small letter j	j
09/02	Latin small letter k	k
09/03	Latin small letter l	l
09/04	Latin small letter m	m
09/05	Latin small letter n	n
09/06	Latin small letter o	o
09/07	Latin small letter p	p
09/08	Latin small letter q	q
09/09	Latin small letter r	r
09/10		
09/11		
09/12		
09/13		
09/14		
09/15		

**Figure 12-139. LocalEBCDIC Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
10/00		
10/01	Tilde	~
10/02	Latin small letter s	s
10/03	Latin small letter t	t
10/04	Latin small letter u	u
10/05	Latin small letter v	v
10/06	Latin small letter w	w
10/07	Latin small letter x	x
10/08	Latin small letter y	y
10/09	Latin small letter z	z
10/10		
10/11		
10/12		
10/13		
10/14		
10/15		

Col/Row	ISO Name	Sym
11/00		
11/01		
11/02		
11/03		
11/04		
11/05		
11/06		
11/07		
11/08		
11/09		
11/10		
11/11		
11/12		
11/13		
11/14		
11/15		

**Figure 12-140. LocalEBCDIC Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
12/00	Left curly bracket	{
12/01	Latin capital letter A	A
12/02	Latin capital letter B	B
12/03	Latin capital letter C	C
12/04	Latin capital letter D	D
12/05	Latin capital letter E	E
12/06	Latin capital letter F	F
12/07	Latin capital letter G	G
12/08	Latin capital letter H	H
12/09		
12/10		
12/11		
12/12		
12/13		
12/14		
12/15		

Col/Row	ISO Name	Sym
13/00	Right curly bracket	}
13/01	Latin capital letter J	J
13/02	Latin capital letter K	K
13/03	Latin capital letter L	L
13/04	Latin capital letter M	M
13/05	Latin capital letter N	N
13/06	Latin capital letter O	O
13/07	Latin capital letter P	P
13/08	Latin capital letter Q	Q
13/09	Latin capital letter R	R
13/10		
13/11		
13/12		
13/13		
13/14		
13/15		

**Figure 12-141. LocalEBCDIC Column Listing
Columns 12 and 13**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
14/00	Reverse solidus	\
14/01	Application program command	APC
14/02	Latin capital letter S	S
14/03	Latin capital letter T	T
14/04	Latin capital letter U	U
14/05	Latin capital letter V	V
14/06	Latin capital letter W	W
14/07	Latin capital letter X	X
14/08	Latin capital letter Y	Y
14/09	Latin capital letter Z	Z
14/10		
14/11		
14/12		
14/13		
14/14		
14/15		

Col/Row	ISO Name	Sym
15/00	Digit 0	0
15/01	Digit 1	1
15/02	Digit 2	2
15/03	Digit 3	3
15/04	Digit 4	4
15/05	Digit 5	5
15/06	Digit 6	6
15/07	Digit 7	7
15/08	Digit 8	8
15/09	Digit 9	9
15/10		
15/11		
15/12		
15/13		
15/14		
15/15	Mark deleted	MD

**Figure 12-142. LocalEBCDIC Column Listing
Columns 14 and 15**

MACROMAN Coded Character Set (8CS61.03)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	NUL	☺	SP	0	@	P	`	p	Ä	ê	†	∞	ı	-	‡	🍏
01	☹	⌘	!	1	A	Q	a	q	Å	ë	°	±	i	-	·	Ò
02	☺	✓	"	2	B	R	b	r	Ç	í	ç	≤	¬	"	,	Ú
03	☹	◆	#	3	C	S	c	s	É	ì	é	≥	√	"	„	Û
04	☺	⇒	\$	4	D	T	d	t	Ñ	î	ñ	¥	f	`	%	Ù
05	☺	☒	%	5	E	U	e	u	Ö	ï	•	μ	≈	'	Â	ı
06	☺	⬅	&	6	F	V	f	v	Ü	ñ	¶	∂	△	÷	Ê	^
07	☹	➡	'	7	G	W	g	w	á	ó	ß	Σ	«	◇	Á	~
08	☒	☞	(8	H	X	h	x	à	ò	®	Ò	»	ÿ	Ë	-
09)	9	I	Y	i	y	â	ô	©	π	...	ÿ	È	˘
10	➡		*	:	J	Z	j	z	ä	ö	™	∫	NBSP	/	Í	´
11	☞		+	;	K	[k	{	ã	õ	'	ª	À	€	Î	°
12	☺		,	<	L	\	l		â	ú	“	°	Ã	<	Ï	¸
13			-	=	M]	m	}	ç	ù	≠	Ω	Õ	>	ì	‘
14	☹		.	>	N	^	n	~	é	û	Æ	œ	Œ	fi	Ó	¸
15	☹		/	?	O	_	o	Δ	è	ü	∅	ø	œ	fi	Ô	˘

Figure 12-143. MACROMAN Code Table (8CS61.03)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00	NUL	
00/01	SOH	
00/02	STX	
00/03	ETX	
00/04	EOT	
00/05	ENQ	
00/06	ACK	
00/07	BEL	
00/08	BS	
00/09	HT	
00/10	LF	
00/11	VT	
00/12	FF	
00/13	CR	
00/14	SO	
00/15	SI	

COL/ROW	ISO Name	SYM
01/00	DLE	
01/01	DC1	
01/02	DC2	
01/03	DC3	
01/04	DC4	
01/05	NAK	
01/06	SYN	
01/07	ETB	
01/08	CAN	
01/09	EM	
01/10	SUB	
01/11	ESC	
01/12	FS	
01/13	GS	
01/14	RS	
01/15	US	

**Figure 12-144. MACROMAN Column Listing
Columns 0 and 1**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	SYM
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 12-145. MACROMAN Column Listing
Columns 2 and 3**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	SYM
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 12-146. MACROMAN Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

Col/Row	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Vertical line	
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Upward-pointing triangle empty	△

**Figure 12-147. MACROMAN Column Listing
Columns 6 and 7**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00	Latin capital letter A with diaeresis	Ä
08/01	Latin capital letter A with ring above	Å
08/02	Latin capital letter C with cedilla	Ç
08/03	Latin capital letter E with acute accent	É
08/04	Latin capital letter N with tilde	Ñ
08/05	Latin capital letter O with diaeresis	Ö
08/06	Latin capital letter U with diaeresis	Ü
08/07	Latin small letter a with acute accent	á
08/08	Latin small letter a with grave accent	à
08/09	Latin small letter a with circumflex accent	â
08/10	Latin small letter a with diaeresis	ä
08/11	Latin small letter a with tilde	ã
08/12	Latin small letter a with ring above	å
08/13	Latin small letter c with cedilla	ç
08/14	Latin small letter e with acute accent	é
08/15	Latin small letter e with grave accent	è

COL/ROW	ISO Name	SYM
09/00	Latin small letter e with circumflex accent	ê
09/01	Latin small letter e with diaeresis	ë
09/02	Latin small letter i with acute accent	í
09/03	Latin small letter i with grave accent	ì
09/04	Latin small letter i with circumflex accent	î
09/05	Latin small letter i with diaeresis	ï
09/06	Latin small letter n with tilde	ñ
09/07	Latin small letter o with acute accent	ó
09/08	Latin small letter o with grave accent	ò
09/09	Latin small letter o with circumflex accent	ô
09/10	Latin small letter o with diaeresis	ö
09/11	Latin small letter o with tilde	õ
09/12	Latin small letter u with acute accent	ú
09/13	Latin small letter u with grave accent	ù
09/14	Latin small letter u with circumflex accent	û
09/15	Latin small letter u with diaeresis	ü

**Figure 12-148. MACROMAN Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	Dagger	†
10/01	Degree sign	°
10/02	Cent sign	¢
10/03	Pound sign	£
10/04	Section sign	§
10/05	Bullet	•
10/06	Pilcrow sign	¶
10/07	Latin small letter sharp s	ß
10/08	Registered sign	®
10/09	Copyright sign	©
10/10	Trademark sign	™
10/11	Acute accent	´
10/12	Diaeresis	¨
10/13	Not equal to sign	≠
10/14	Latin capital letter AE	Æ
10/15	Latin capital letter O with oblique stroke	Ø

COL/ROW	ISO Name	SYM
11/00	Infinity sign	∞
11/01	Plus-minus sign	±
11/02	Less-than-or-equal-to sign	≤
11/03	Greater-than-or-equal-to sign	≥
11/04	Yen sign	¥
11/05	Micro sign	μ
11/06	Partial Differential	∂
11/07	Summation	∑
11/08	Product	∏
11/09	Greek small letter pi	π
11/10	Integral	∫
11/11	Feminine ordinal indicator	ª
11/12	Masculine ordinal indicator	º
11/13	Greek capital letter omega	Ω
11/14	Latin small letter ae	æ
11/15	Latin small letter o with oblique stroke	ø

**Figure 12-149. MACROMAN Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISOName	SYM
12/00	Inverted question mark	¿
12/01	Inverted exclamation mark	¡
12/02	Not sign	α
12/03	Square root sign	√
12/04	Function sign	f
12/05	Almost equal to sign	≈
12/06	Greek capital letter delta	Δ
12/07	Left-pointing double angle quotation mark	«
12/08	Right-pointing double angle quotation mark	»
12/09	Horizontal ellipsis	...
12/10	No-break space	NBSP
12/11	Latin capital letter A with grave accent	À
12/12	Latin capital letter A with tilde	Ã
12/13	Latin capital letter O with tilde	Õ
12/14	Latin capital ligature OE	Œ
12/15	Latin small ligature oe	œ

COL/ROW	ISOName	SYM
13/00	En dash	-
13/01	Em dash	—
13/02	Left double quotation mark	“
13/03	Right double quotation mark	”
13/04	Left single quotation mark	‘
13/05	Right single quotation mark	’
13/06	Division sign	÷
13/07	Lozenge	◇
13/08	Latin small letter y with diaeresis	ÿ
13/09	Latin capital letter Y with diaeresis	ÿ
13/10	Fraction slash	/
13/11	Euro sign	€
13/12	Single left-pointing angle quotation mark	<
13/13	Single right-pointing angle quotation mark	>
13/14	Latin small ligature fi	fi
13/15	Latin small ligature fl	fl

**Figure 12-150. MACROMAN Column Listing
Columns 12 and 13**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Double dagger	‡
14/01	Middle dot	•
14/02	Single low-9 quotation mark	,
14/03	Double low-9 quotation mark	”
14/04	Per mille sign	‰
14/05	Latin capital letter A with circumflex accent	Â
14/06	Latin capital letter E with circumflex accent	Ê
14/07	Latin capital letter A with acute accent	Á
14/08	Latin capital letter E with diaeresis	Ë
14/09	Latin capital letter E with grave accent	È
14/10	Latin capital letter I with acute accent	Í
14/11	Latin capital letter I with circumflex accent	Î
14/12	Latin capital letter I with diaeresis	Ï
14/13	Latin capital letter I with grave accent	Ì
14/14	Latin capital letter O with acute accent	Ó
14/15	Latin capital letter O with circumflex accent	Ô

COL/ROW	ISO Name	SYM
15/00	Apple Logo (Macintosh F0)	🍏
15/01	Latin capital letter O with grave accent	Ò
15/02	Latin capital letter U with acute accent	Ú
15/03	Latin capital letter U with circumflex accent	Û
15/04	Latin capital letter U with grave accent	Ù
15/05	Latin small letter i dotless	ı
15/06	Circumflex	ˆ
15/07	Tilde	˜
15/08	Macron	ˉ
15/09	Breve	˘
15/10	Dot above	˙
15/11	Ring above	◌̥
15/12	Cedilla	¸
15/13	Double acute accent	˝
15/14	Ogonek	˛
15/15	Caron	ˇ

**Figure 12-151. MACROMAN Column Listing
Columns 14 and 15**

NorwayBTOS Coded Character Set (8CS50.00)

Row	Column															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00		¼	SP	0	@	P	`	p		⁴	Ü	¨	†	‡	⌞	⌟
01	↑	†	!	1	A	Q	a	q	l	⁵	ü	£	‡	‡	⌞	‡
02	▲	→	"	2	B	R	b	r	l ₀	⁶	Æ	§	‡	‡	†	‡
03	¢	⌞	#	3	C	S	c	s	l ₁	⁷	æ	´	‡	‡	†	‡
04	■	▪	\$	4	D	T	d	t	l ₂	⁸	Ä	à	†	‡	⌞	‡
05	□	÷	%	5	E	U	e	u	l ₃	⁹	ä	è	‡	‡	†	†
06	½	l	&	6	F	V	f	v	l ₄	⁰	Ø	l ₀	l	‡	‡	†
07	⌚	§	'	7	G	W	g	w	l ₅	¹	ø	l ₁	‡	‡	⌞	‡
08	†	≠	(8	H	X	h	x	l ₆	²	Ö	l ₂	‡	=	⌞	†
09	†	≈)	9	l	Y	i	y	l ₇	³	ö	l ₃	†	‡	⌞	†
10	↙	¶	*	:	J	Z	j	z	l ₈	⁴	Å	l ₄	†	—	⌞	†
11	↓	•	+	;	K	[k	{	l ₉	⁵	å	l ₅	‡	‡	⌞	‡
12	↘	⌞	,	<	L	\	l	;	⁰	⁶	é	l ₆	†	†	⌞	†
13	▲	≤	—	=	M]	m	}	¹	⁷	á	l ₇	†	†	⌞	▨
14	†	±	.	>	N	^	n	~	²	⁸	½	l ₈	—	‡	⌞	▨
15	‡	≥	/	?	O	_	o	—	³	⁹	·	l ₉	‡	†	⌞	▨

Figure 12-152. NorwayBTOS Code Table (8CS50.00)

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
00/00		
00/01	Upward arrow	↑
00/02	Lower left corner triangle solid	▴
00/03	Cent sign	¢
00/04	Square solid	■
00/05	Square empty	□
00/06	Vulgar fraction one half	½
00/07	Bell sign	🔔
00/08	Left tab sign	⌠
00/09	Right tab sign	⌡
00/10	Left return sign	↶
00/11	Downward arrow	↓
00/12	Right return sign	↷
00/13	Lower right corner triangle solid	▾
00/14	Leftward arrow	←
00/15	Double dagger	‡

COL/ROW	ISO Name	
01/00	Vulgar fraction one quarter	¼
01/01	Dagger	†
01/02	Rightward arrow	→
01/03	Space symbol	␣
01/04	Box drawing solid small square	▣
01/05	Division sign	÷
01/06	Vertical line	
01/07	Paragraph sign	§
01/08	Not equal-to sign	≠
01/09	Asymptotically equal sign	≈
01/10	Pilcrow sign	¶
01/11	Middle dot	•
01/12	Not sign	¬
01/13	Less-than or equal-to sign	≤
01/14	Plus-minus sign	±
01/15	Greater-than or equal-to sign	≥

**Figure 12-153. NorwayBTOS Column Listing
Columns 0 and 1**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
02/00	Space	SP
02/01	Exclamation mark	!
02/02	Quotation mark	"
02/03	Number sign	#
02/04	Dollar sign	\$
02/05	Percent sign	%
02/06	Ampersand	&
02/07	Apostrophe	'
02/08	Left parenthesis	(
02/09	Right parenthesis)
02/10	Asterisk	*
02/11	Plus sign	+
02/12	Comma	,
02/13	Hyphen-minus	-
02/14	Full stop	.
02/15	Solidus	/

Col/Row	ISO Name	Sym
03/00	Digit 0	0
03/01	Digit 1	1
03/02	Digit 2	2
03/03	Digit 3	3
03/04	Digit 4	4
03/05	Digit 5	5
03/06	Digit 6	6
03/07	Digit 7	7
03/08	Digit 8	8
03/09	Digit 9	9
03/10	Colon	:
03/11	Semicolon	;
03/12	Less-than sign	<
03/13	Equal sign	=
03/14	Greater-than sign	>
03/15	Question mark	?

**Figure 12-154. NorwayBTOS Column Listing
Columns 2 and 3**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

Col/Row	ISO Name	Sym
04/00	Commercial at	@
04/01	Latin capital letter A	A
04/02	Latin capital letter B	B
04/03	Latin capital letter C	C
04/04	Latin capital letter D	D
04/05	Latin capital letter E	E
04/06	Latin capital letter F	F
04/07	Latin capital letter G	G
04/08	Latin capital letter H	H
04/09	Latin capital letter I	I
04/10	Latin capital letter J	J
04/11	Latin capital letter K	K
04/12	Latin capital letter L	L
04/13	Latin capital letter M	M
04/14	Latin capital letter N	N
04/15	Latin capital letter O	O

Col/Row	ISO Name	Sym
05/00	Latin capital letter P	P
05/01	Latin capital letter Q	Q
05/02	Latin capital letter R	R
05/03	Latin capital letter S	S
05/04	Latin capital letter T	T
05/05	Latin capital letter U	U
05/06	Latin capital letter V	V
05/07	Latin capital letter W	W
05/08	Latin capital letter X	X
05/09	Latin capital letter Y	Y
05/10	Latin capital letter Z	Z
05/11	Left square bracket	[
05/12	Reverse solidus	\
05/13	Right square bracket]
05/14	Circumflex accent	^
05/15	Low line	_

**Figure 12-155. NorwayBTOS Column Listing
Columns 4 and 5**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
06/00	Grave accent	`
06/01	Latin small letter a	a
06/02	Latin small letter b	b
06/03	Latin small letter c	c
06/04	Latin small letter d	d
06/05	Latin small letter e	e
06/06	Latin small letter f	f
06/07	Latin small letter g	g
06/08	Latin small letter h	h
06/09	Latin small letter i	i
06/10	Latin small letter j	j
06/11	Latin small letter k	k
06/12	Latin small letter l	l
06/13	Latin small letter m	m
06/14	Latin small letter n	n
06/15	Latin small letter o	o

COL/ROW	ISO Name	SYM
07/00	Latin small letter p	p
07/01	Latin small letter q	q
07/02	Latin small letter r	r
07/03	Latin small letter s	s
07/04	Latin small letter t	t
07/05	Latin small letter u	u
07/06	Latin small letter v	v
07/07	Latin small letter w	w
07/08	Latin small letter x	x
07/09	Latin small letter y	y
07/10	Latin small letter z	z
07/11	Left curly bracket	{
07/12	Broken bar	⏏
07/13	Right curly bracket	}
07/14	Tilde	~
07/15	Box drawing solid upper half block	▀

**Figure 12-156. NorwayBTOS Column Listing
Columns 6 and 7**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
08/00		
08/01	Vertical line	
08/02	Vertical line with subscript 0	₀
08/03	Vertical line with subscript 1	₁
08/04	Vertical line with subscript 2	₂
08/05	Vertical line with subscript 3	₃
08/06	Vertical line with subscript 4	₄
08/07	Vertical line with subscript 5	₅
08/08	Vertical line with subscript 6	₆
08/09	Vertical line with subscript 7	₇
08/10	Vertical line with subscript 8	₈
08/11	Vertical line with subscript 9	₉
08/12	Superscript 0	⁰
08/13	Superscript 1	¹
08/14	Superscript 2	²
08/15	Superscript 3	³

COL/ROW	ISO Name	
09/00	Superscript 4	⁴
09/01	Superscript 5	⁵
09/02	Superscript 6	⁶
09/03	Superscript 7	⁷
09/04	Superscript 8	⁸
09/05	Superscript 9	⁹
09/06	Subscript 0	₀
09/07	Subscript 1	₁
09/08	Subscript 2	₂
09/09	Subscript 3	₃
09/10	Subscript 4	₄
09/11	Subscript 5	₅
09/12	Subscript 6	₆
09/13	Subscript 7	₇
09/14	Subscript 8	₈
09/15	Subscript 9	₉

**Figure 12-157. NorwayBTOS Column Listing
Columns 8 and 9**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
10/00	Latin capital letter U with diaeresis	Ü
10/01	Latin small letter u with diaeresis	ü
10/02	Latin capital letter A with E	Æ
10/03	Latin small letter a with e	æ
10/04	Latin capital letter A with diaeresis	Ä
10/05	Latin small letter a with diaeresis	ä
10/06	Latin capital letter O with oblique stroke	Ø
10/07	Latin small letter o with oblique stroke	ø
10/08	Latin capital letter O with diaeresis	Ö
10/09	Latin small letter o with diaeresis	ö
10/10	Latin capital letter A with ring above	Å
10/11	Latin small letter a with ring above	å
10/12	Latin small letter e with acute accent	é
10/13	Latin small letter a with acute accent	á
10/14	Vulgar fraction one half	½
10/15	Grave accent	`

COL/ROW	ISO Name	
11/00	Diaeresis	¨
11/01	Pound sign	£
11/02	Paragraph sign	§
11/03	Acute accent	´
11/04	Latin small letter a with grave accent	à
11/05	Latin small letter e with grave accent	è
11/06	Short vertical line with subscript 0	₀
11/07	Short vertical line with subscript 1	₁
11/08	Short vertical line with subscript 2	₂
11/09	Short vertical line with subscript 3	₃
11/10	Short vertical line with subscript 4	₄
11/11	Short vertical line with subscript 5	₅
11/12	Short vertical line with subscript 6	₆
11/13	Short vertical line with subscript 7	₇
11/14	Short vertical line with subscript 8	₈
11/15	Short vertical line with subscript 9	₉

**Figure 12-158. NorwayBTOS Column Listing
Columns 10 and 11**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
12/00	Box drawing heavy vertical and light right	┃
12/01	Box drawing heavy vertical and light left	┃
12/02	Box drawing heavy vertical and light horizontal	┃┃
12/03	Box drawing heavy vertical and horizontal	┃┃
12/04	Box drawing heavy vertical and right	┃┃
12/05	Box drawing heavy vertical and left	┃┃
12/06	Box drawing heavy vertical	┃
12/07	Box drawing heavy vertical and left and light right	┃┃
12/08	Box drawing heavy vertical and right and light left	┃┃
12/09	Box drawing heavy horizontal and light down	┃
12/10	Box drawing light horizontal and heavy up	┃
12/11	Box drawing heavy horizontal and light vertical	┃
12/12	Box drawing heavy horizontal and down	┃
12/13	Box drawing heavy horizontal and up	┃
12/14	Box drawing heavy horizontal	┃
12/15	Box drawing heavy horizontal and up and light down	┃

COL/ROW	ISO Name	
13/00	Box drawing heavy horizontal and down and light up	┃
13/01	Box drawing double vertical and light right	┃┃
13/02	Box drawing double vertical and light left	┃┃
13/03	Box drawing double vertical	┃┃
13/04	Box drawing double vertical and light horizontal	┃┃
13/05	Box drawing double vertical and horizontal	┃┃
13/06	Box drawing double horizontal and light down	┃┃
13/07	Box drawing double horizontal and light up	┃┃
13/08	Box drawing double horizontal	┃┃
13/09	Box drawing double horizontal and light vertical	┃┃
13/10	Box drawing light horizontal	┃
13/11	Box drawing light horizontal and vertical	┃
13/12	Box drawing light horizontal and down	┃
13/13	Box drawing light horizontal and up	┃
13/14	Box drawing light horizontal and double up	┃
13/15	Box drawing light horizontal and double down	┃

**Figure 12-159. NorwayBTOS Column Listing
Columns 12 and 13**

Eight Bit Coded Character Sets: FrenchArabicEBCDIC to NorwayBTOS

COL/ROW	ISO Name	SYM
14/00	Box drawing double up and right	⌞
14/01	Box drawing light vertical	
14/02	Box drawing light vertical and right	┤
14/03	Box drawing light vertical and left	├
14/04	Box drawing double down and right	⌋
14/05	Box drawing light vertical and double right	⌋
14/06	Box drawing light vertical and double left	⌋
14/07	Box drawing double up and left	⌞
14/08	Box drawing double down and left	⌋
14/09	Box drawing heavy up and right	┘
14/10	Box drawing heavy up and left	┐
14/11	Box drawing heavy down and right	┘
14/12	Box drawing heavy down and left	┐
14/13	Box drawing light up and right	┘
14/14	Box drawing light up and left	┐
14/15	Box drawing light down and right	┘

COL/ROW	ISO Name	SYM
15/00	Box drawing light down and left	└
15/01	Box drawing double vertical and right	⌋
15/02	Box drawing double horizontal and down	⌋
15/03	Box drawing double vertical and left	⌋
15/04	Box drawing double horizontal and up	⌋
15/05	Box drawing heavy vertical and double right	⌋
15/06	Box drawing heavy horizontal and double down	⌋
15/07	Box drawing heavy vertical and double left	⌋
15/08	Box drawing heavy horizontal and double up	⌋
15/09	Box drawing double vertical and heavy right	⌋
15/10	Box drawing double horizontal and heavy down	⌋
15/11	Box drawing double vertical and heavy left	⌋
15/12	Box drawing double horizontal and heavy up	⌋
15/13	BTOS control 1	▨
15/14	BTOS control 2	▩
15/15	BTOS control 3	▪

**Figure 12-160. NorwayBTOS Column Listing
Columns 14 and 15**

Section 13

Double-Byte and Mixed, Multibyte Coded Character Sets

This section includes double-byte and mixed, multibyte coded character set tables arranged alphabetically from ASTASCIIGB2312 to UCS2NT. You can find 8-bit coded character sets in Sections 10, 11, and 12. Ccsversions are in Section 14 and standard convention definitions are in Section 15. Refer to Section 9, “Coded Character Sets and Coded Character Set Mapping Tables,” for a complete list of the coded character sets and coded character set mapping tables supported in the current release.

Multibyte coded character sets, which contain either pure 16-bit characters or a mix of 8- and 16-bit characters, are more difficult to represent graphically than their 8-bit counterparts. Eight-bit coded character sets encode a maximum of 256 characters. Sixteen-bit coded character sets, on the other hand, can encode a maximum of 65,536 characters, although most 16-bit character sets do not use all 65,536 code positions.

Rather than displaying and defining every character of every 16-bit coded character set supported on the release media (many sets are simply variations of each other), this section provides a brief definition of each supported coded character set and describes where the code points are encoded within the 16-bit range of hexadecimal (hex) x0000–xFFFF (0–65535 decimal) values.

Table 13–1 lists the available pure 16-bit and mixed, multibyte coded character sets by the nationality supported. Character set numbers beginning with “16” denote pure 16-bit tables; numbers beginning with “816” denote mixed, multibyte tables.

Table 13-1. Double-Byte and Mixed, Multibyte Coded Character Sets by Nationality Supported

Nationality	Coded Character Set Name	Coded Character Set Number	System Compatibility
China	ASTASCIIGB2312	816CS00.00	ASCII-based terminals
China	CODEPAGE936	816CS00.00	PC-based systems
China	EBCDICGB2312	816CS00.00	ClearPath NX and A Series servers
Korea	ASTASCIIKSC5601	816CS00.00	ASCII-based terminals and UNIX systems
Korea	CODEPAGE949	816CS00.00	PC-based systems
Korea	EBCDICKSC5601	816CS00.00	ClearPath NX and A Series servers
Japan	CODEPAGE932	816CS00.00	PC-based systems
Japan	EUCJp	816CS00.00	UNIX systems
Japan	JapanASCIIJBIS7	816CS00.00	ASCII-based terminals
Japan	JapanEBCDICJBIS8	816CS00.00	ClearPath NX and A Series servers
Japan	JapanV24JBIS8	816CS00.00	ClearPath NX and A Series servers
Japan	LetsJ	816CS00.00	ClearPath IX servers
Japan	JBIS7	16CS00.00	ASCII-based terminals
Japan	JBIS8	16CS00.00	ClearPath NX and A Series servers
International	UCS2	16CS00.00	Various servers
International	UCS2NT	16CS00.00	PC-based systems
Taiwan	ASUTL	16CS00.00	ClearPath NX and A Series servers
Taiwan	ASTUTL	16CS00.00	ASCII-based terminals
Taiwan	ASTASCIUUTL	816CS00.00	ASCII-based terminals
Taiwan	CODEPAGE950	816CS00.00	PC-based systems
Taiwan	EBCDICUTL	816CS00.00	ClearPath NX and A Series servers

How to Read Coded Character Set Diagrams

Double-byte and mixed, multibyte coded character sets are represented by a different format from 8-bit coded character set tables. The following paragraphs explain how to read the notation used in the diagrams.

- All examples use hexadecimal notation. The characters 0–9 and A–F are used to represent decimal values 0 to 15. To identify hex notation, a lowercase “x” precedes each value.
- Eight-bit characters are defined by their hex encoding. For example, in ASeriesEBCDIC, x2B is the hex representation for the character at column 2, row 11 (decimal value 43), which is the start double octet (SDO) character. A range of 8-bit characters is denoted by x00–x7F, which represents the first 128 character positions of an 8-bit coded character set. In a mixed, multibyte coded character set diagram, the 8-bit coded character set components are listed and described above the diagram.
- Sixteen-bit characters are also defined by their hex encoding. For example, x0000 is the first character position (decimal value 0), while xFFFF is one of the last character positions (decimal value 65533). Sixteen-bit coded character sets are represented graphically within pure 16-bit and mixed, multibyte coded character set diagrams.
- Most of the coded character set descriptions use byte pair range values to describe the active areas of the 16-bit plane. Leading bytes are shown by row values while trailing bytes are shown by column values. The leading byte range and trailing byte range each can have a value of x00–xFF. For example, the byte pair range (xA1–xFE, x41–x9E) shows that the first byte may range from xA1 to xFE, while the second byte may range from x41 to x9E. The first character in this range is xA141 and the last is xFE9E. Note that xA141–xFE9E does not represent the same range as it implies that xA200 falls within the range, which is not the case.

Double-Byte and Mixed, Multibyte Coded Character Sets

Example 13-1 provides the definition and graphical representation of a fictional mixed, multibyte coded character set.

This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F.
- A 16-bit coded character set encoded in the byte pair range (xA1–xFE, x41–x9E).

8-bit ASCII (x00–x7F) +

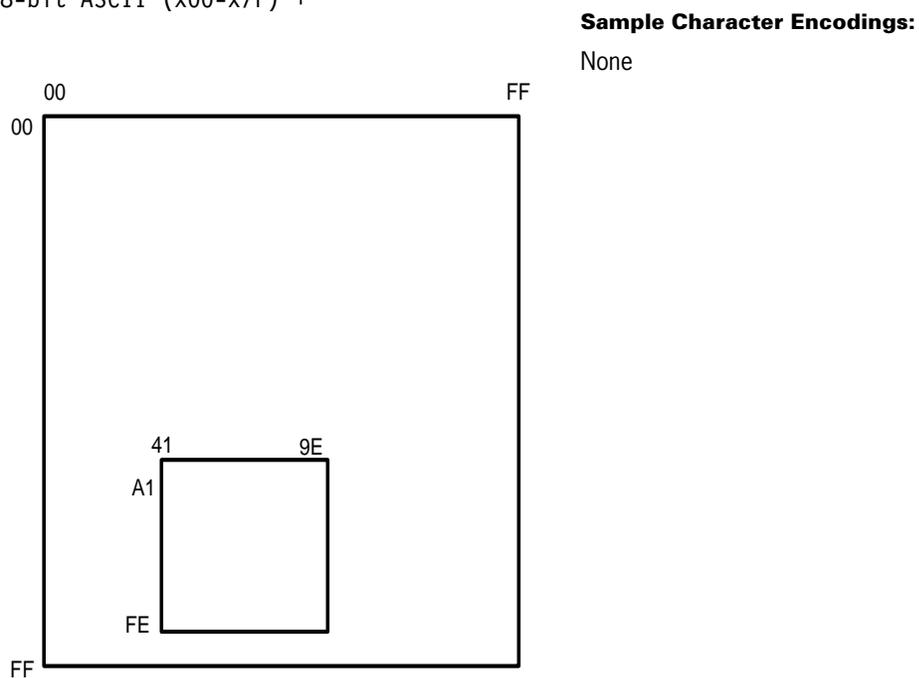


Figure 13-1. Reading Mixed, Multibyte Coded Character Sets

ASTASCIIGB2312 Coded Character Set (816CS00.00)

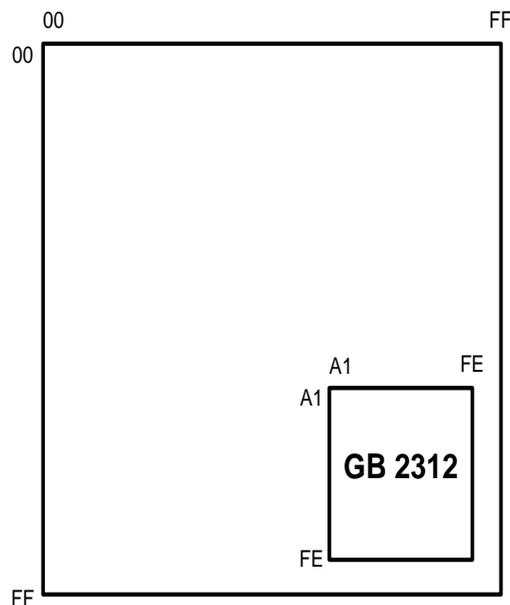
ASTASCIIGB2312 is a mixed, multibyte coded character set used on Chinese ASCII-based terminals. This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSupplISO,” for a complete description of the ASCII coded character set.
- A 16-bit coded character set based on the Guo Biao (“National Standard”) GB 2312-80 Simplified Chinese standard coded character set, which represents over 7,000 simplified Chinese characters. The ASCII terminals encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the GB 2312-80 coded character set is located at code position xA1A1.

Within a sequence of characters, all of the 16-bit data is bracketed by the ASCII encoding of the 8-bit SDO and EDO characters, byte values x9E and x9F respectively, to signify the start and end of the 16-bit data.

8-bit ASCII (x00–x7F) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A3 Ideographic full stop
- xA1A4 Katakana middle dot
- xA1A5 Macron
- xA1A6 Caron
- xA1A7 Diaeresis
- xA1A8 Ditto mark

Figure 13-2. ASTASCIIGB2312 Coded Character Set (816CS00.00)

ASTASCIIKSC5601 Coded Character Set (816CS00.00)

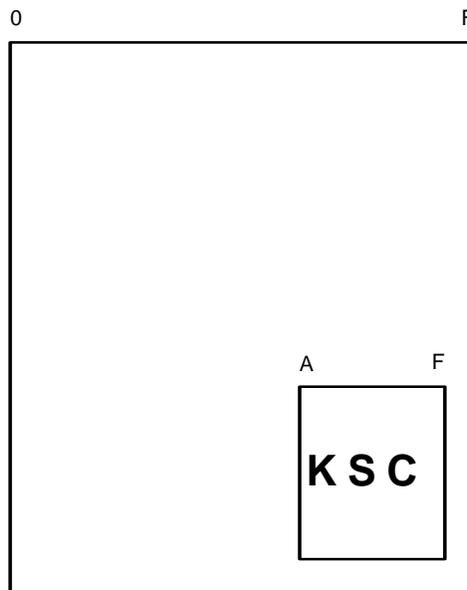
ASTASCIIKSC5601 is a mixed, multibyte coded character set used on Korean ASCII-based terminals and UNIX systems. This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSupplISO,” for a complete description of the ASCII coded character set.
- A 16-bit character set based on the Korean standard character set KSC 5601-1987, which represents over 8,000 Korean characters including Hangul syllables and Kanja ideographs. The ASCII terminals and UNIX systems encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the KSC 5601-1987 coded character set is located at code position xA1A1.

Within a sequence of characters, there are no delimiters between 8-bit and 16-bit characters. They are simply identified by their leading byte value.

8-bit ASCII (x00–x7F) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A3 Fullwidth full stop
- xA1A4 Katakana middle dot
- xA1A5 Two dot leader
- xA1A6 Horizontal ellipsis
- xA1A7 Diaeresis
- xA1A8 Ditto mark

Figure 13-3. ASTASCIIKSC5601 Coded Character Set (816CS00.00)

ASTASCIUUTL Coded Character Set (816CS00.00)

ASTASCIUUTL is a mixed, multibyte coded character set used on Taiwanese ASCII-based terminals. This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSupplISO,” for a complete description of the ASCII coded character set.
- A 16-bit coded character set defined by Unisys Taiwan Limited (UTL) that represents over 19,000 Taiwanese characters. The ASCII terminal encoding places these characters throughout the byte pair ranges (x21–x7E, x21–x7E), (x21–x7E, xA1–xFE), (xA1–xFB, x21–7E), and (xA1–xFB, xA1–xFE).

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the UTL coded character set is located at code position x21A1.

Within a sequence of characters, all of the 16-bit data is bracketed by the ASCII encoding of the 8-bit SDO and EDO characters, byte values x9E and x9F respectively, to signify the start and end of the 16-bit data.

The code positions for ASTASCIUUTL are not currently defined. No translation tables exist to support mappings to or from this character set.

8-bit ASCII (x00–x7F) +

Sample Character Encodings:

None

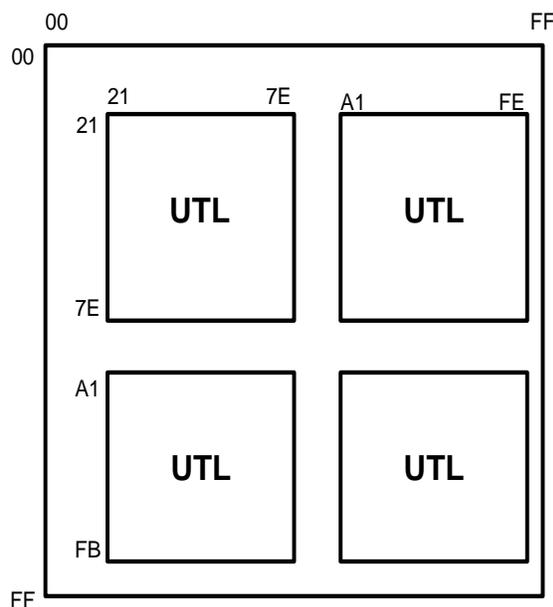


Figure 13-4. ASTASCIUUTL Coded Character Set (816CS00.00)

ASUTL Coded Character Set (16CS00.00)

ASUTL is a pure 16-bit coded character set for use on Taiwanese ClearPath NX and A Series servers. This coded character set consists of

- The 16-bit coded character set defined by Unisys Taiwan Limited (UTL) that represents over 19,000 characters for the Taiwanese market. The server systems encoding places these characters in the byte pair ranges (x41–x9E, x41–x9E), (x41–x9E, xA1–xFE), (xA1–xFB, x41–x9E), and (xA1–xFB, xA1–xFE).

The 16-bit space character (ideographic space) in the UTL coded character set is located at code position x41A1.

The ASUTL coded character set is used when double-byte characters are needed without the surrounding bracket characters SDO and EDO, such as when double-byte characters are embedded in a COBOL85 record and defined as National characters.

The code positions for ASUTL are not currently defined. No translation tables exist to support mappings to or from this character set.

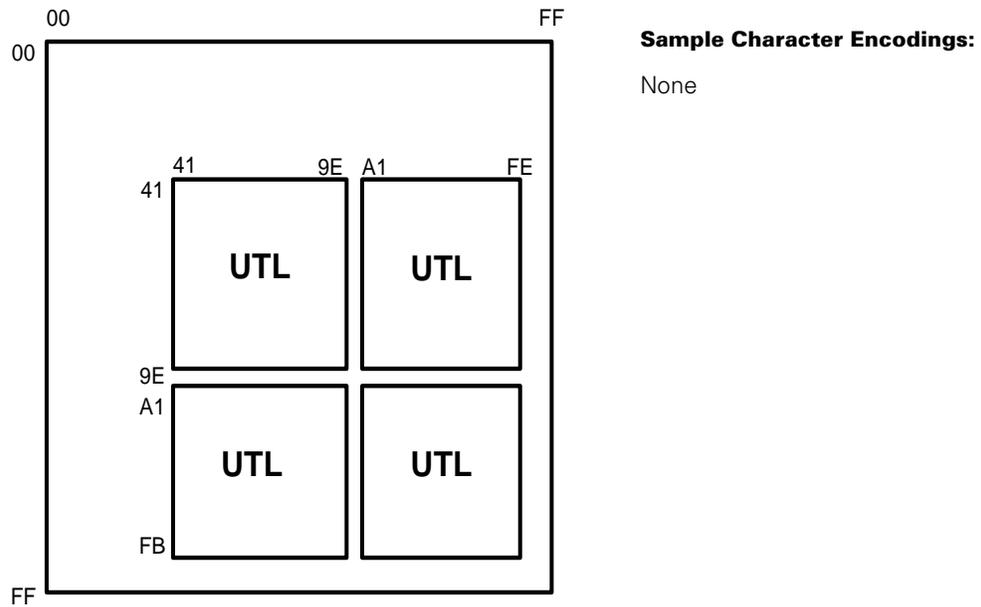


Figure 13-6. ASUTL Coded Character Set (16CS00.00)

CODEPAGE932 Coded Character Set (816CS00.00)

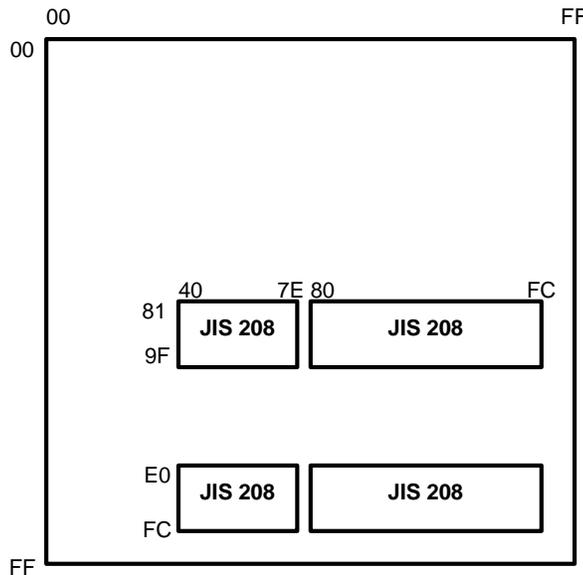
CODEPAGE932 is a mixed, multibyte coded character set, also known as Shift Japanese Industrial Standard, that is used on Japanese PC-based systems. This coded character set consists of

- The 8-bit Japanese coded character set defined as JISASCII. JISASCII is a combination of the ASCII coded character set and the Japanese Industrial Standard JISX0201 coded character set, which represents 63 Katakana characters placed where no ASCII characters are defined. The 8-bit ASCII characters are encoded in the byte range x00–x7F. The 8-bit Katakana characters are encoded in the byte range xA1–xDF. Refer to Section 12, “Eight-Bit Coded Character Sets: FrenchArabicE to NorwayBTOS,” for a complete description of the JISASCII coded character set.
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The PC encoding places these characters in the byte pair ranges (x81–x9F, x40–x7E), (x81–x9F, x80–xFC), (xE0–xFC, x40–x7E), and (xE0–xFC, x80–xFC).

The 8-bit space character in the JISASCII coded character set is located at code position x20. The 16-bit ideographic space character in the JISX0208 coded character set is located at code position x8140.

Within a sequence of characters, there are no delimiters between 8-bit and 16-bit characters. They are simply identified by their leading byte value.

8-bit JISASCII (x00–x7F and xA1–xDF) +



Sample Character Encodings:

- x8140 Ideographic space
- x8141 Ideographic comma
- x8142 Ideographic full stop
- x8143 Fullwidth comma
- x8144 Fullwidth full stop
- x8145 Katakana middle dot
- x8146 Fullwidth colon
- x8147 Fullwidth semicolon

Figure 13–7. CODEPAGE932 Coded Character Set (816CS00.00)

CODEPAGE936 Coded Character Set (816CS00.00)

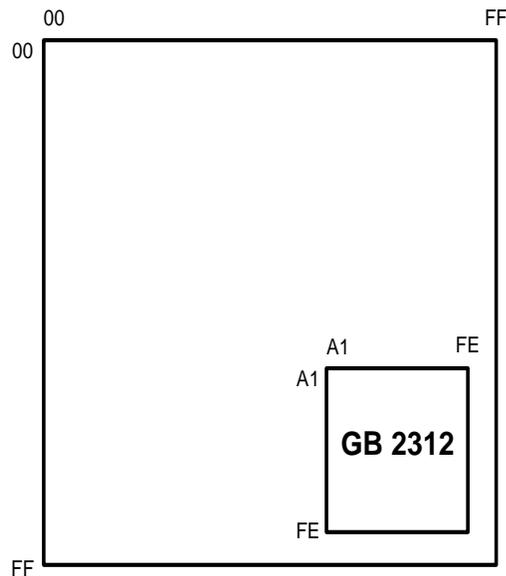
CODEPAGE936 is a mixed, multibyte coded character set used on Chinese PC-based systems. This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CanSupplISO,” for a complete description of the ASCII coded character set.
- A 16-bit coded character set based on the Guo Biao (“National Standard”) GB 2312-80 Simplified Chinese standard coded character set, which represents over 7,000 simplified Chinese characters. The PC-based systems encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the GB 2312-80 coded character set is located at code position xA1A1.

Within a sequence of characters, there are no delimiters between 8-bit and 16-bit characters. They are simply identified by their leading byte value.

8-bit ASCII (x00–x7F) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A3 Ideographic full stop
- xA1A4 Katakana middle dot
- xA1A5 Macron
- xA1A6 Caron
- xA1A7 Diaeresis
- xA1A8 Ditto mark

Figure 13–8. CODEPAGE936 Coded Character Set (816CS00.00)

CODEPAGE949 Coded Character Set (816CS00.00)

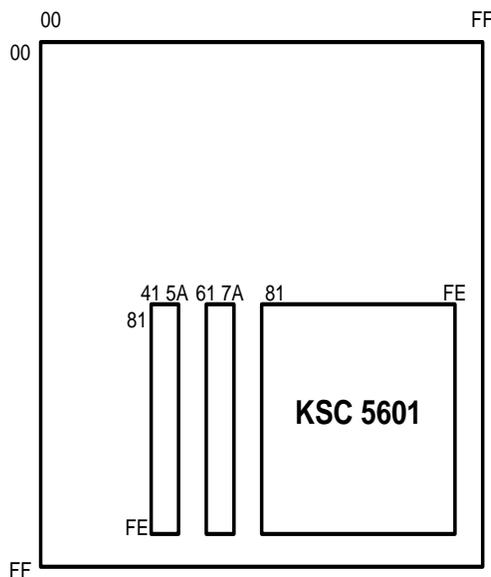
CODEPAGE949 is a mixed, multibyte coded character set used on Korean PC-based systems. This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CanSupplISO,” for a complete description of the ASCII coded character set.
- A 16-bit coded character set based on the Korean standard character set KSC 5601-1987, which represents over 17,000 Korean characters including Hanguk syllables and Kanja ideographs. The Korean PC encoding places these characters in the byte pair ranges (x81–xFE, x41–x5A), (x81–xFE, x61–x7A), and (x81–xFE, x81–xFE).

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the KSC 5601-1987 coded character set is located at code position xA1A1.

Within a sequence of characters, there are no delimiters between 8-bit and 16-bit characters. They are simply identified by their leading byte value.

8-bit ASCII (x00–x7F) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A3 Fullwidth full stop
- xA1A4 Middle dot
- xA1A5 Two dot leader
- xA1A6 Horizontal ellipsis
- xA1A7 Diaeresis
- xA1A8 Ditto mark

Figure 13-9. CODEPAGE949 Coded Character Set (816CS00.00)

CODEPAGE950 Coded Character Set (816CS00.00)

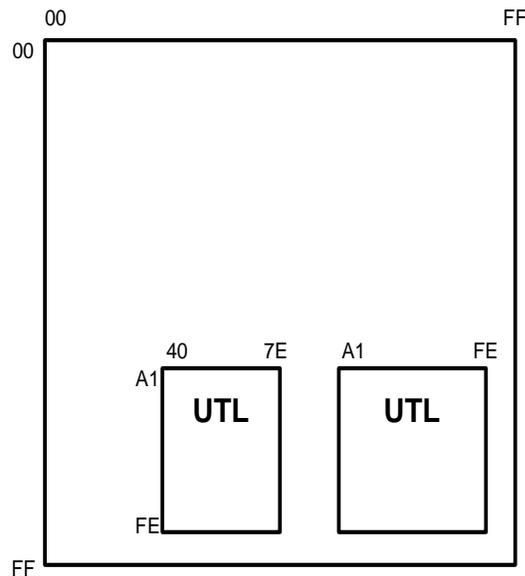
CODEPAGE950 is a mixed, multibyte coded character set used on Taiwanese PC-based systems. This coded character set consists of

- The 8-bit coded character set ASCII, which has characters encoded in the byte range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CanSupplISO,” for a complete description of the ASCII coded character set.
- A 16-bit coded character set based on the Big-5 Traditional Chinese standard coded character set plus Unisys Taiwan Limited (UTL) extensions, which together represent over 13,000 Taiwanese characters. The PC systems encoding places these characters in the byte pair ranges (xA1–xFE, x40–x7E) and (xA1–xFE, xA1–xFE).

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the UTL coded character set is located at code position xA140.

Within a sequence of characters, there are no delimiters between 8-bit and 16-bit characters. They are simply identified by their leading byte value.

8-bit ASCII (x00–x7F) +



Sample Character Encodings:

- xA140 Ideographic space
- xA141 Fullwidth comma
- xA142 Ideographic comma
- xA143 Ideographic full stop
- xA244 Fullwidth full stop
- xA245 Bullet
- xA246 Fullwidth semicolon
- xA247 Fullwidth colon

Figure 13–10. CODEPAGE950 Coded Character Set (816CS00.00)

EBCDICGB2312 Coded Character Set (816CS00.00)

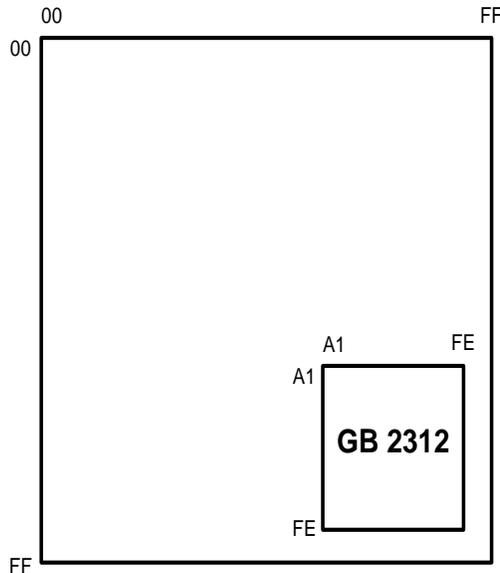
EBCDICGB2312 is a mixed, multibyte coded character set used on Chinese ClearPath NX and A Series servers. This coded character set consists of

- The 8-bit coded character set ASeriesEBCDIC, which has characters scattered throughout byte range x00–xFF. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSuppIIISO,” for a complete description of the ASeriesEBCDIC coded character set.
- A 16-bit coded character set based on the Guo Biao (“National Standard”) GB 2312–80 Simplified Chinese standard coded character set, which represents over 7,000 simplified Chinese characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).

The 8-bit space character in the ASeriesEBCDIC coded character set is located at code position x40. The 16-bit ideographic space character in the GB 2312-80 coded character set is located at code position xA1A1.

Within a sequence of characters, all of the 16-bit data is bracketed by the 8-bit SDO and EDO characters, byte values x2B and x2C respectively, to signify the start and end of the 16-bit data.

8-bit ASeriesEBCDIC (x00–xFF) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A3 Ideographic full stop
- xA1A4 Katakana middle dot
- xA1A5 Macron
- xA1A6 Caron
- xA1A7 Diaeresis
- xA1A8 Ditto mark

Figure 13–11. EBCDICGB2312 Coded Character Set (816CS00.00)

EBCDICKSC5601 Coded Character Set (816CS00.00)

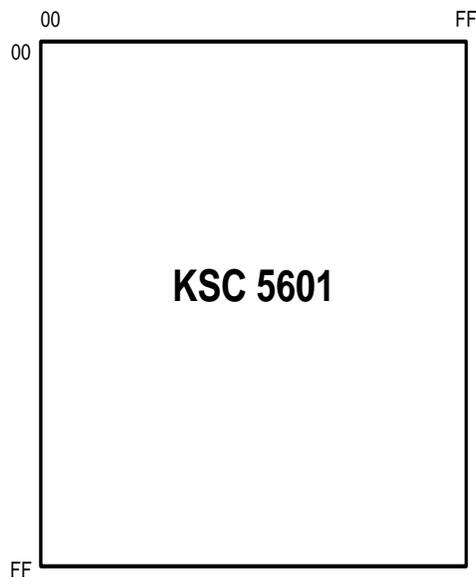
EBCDICKSC5601 is a mixed, multibyte coded character set used on Korean ClearPath NX and A Series servers. This coded character set consists of

- The 8-bit coded character set ASeriesEBCDIC, which has characters scattered throughout byte range x00–xFF. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSuppIIISO,” for a complete description of the ASeriesEBCDIC coded character set.
- A 16-bit coded character set based on the Korean standard character set KSC 5601-1987, which represents over 8,000 Korean characters including Hangul syllables and Kanja ideographs. The server systems encoding scatters these characters throughout the entire 16-bit plane from x0000–xFFFD, with both bytes being empty slots in the ASeriesEBCDIC encoding.

The 8-bit space character in the ASeriesEBCDIC coded character set is located at code position x40. The 16-bit ideographic space character in the KSC 5601-1987 coded character set is located at code position x4242.

Within a sequence of characters, there are no delimiters between 8-bit and 16-bit characters. They are simply identified by their leading byte value.

8-bit ASeriesEBCDIC (x00–xFF) +



Sample Character Encodings:

x4242	Ideographic space
x4243	Ideographic comma
x4244	Fullwidth full stop
x4245	Katakana middle dot
x4246	Two dot leader
x4247	Horizontal ellipsis
x4248	Diaeresis
x4249	Ditto mark

Figure 13–12. EBCDICKSC5601 Coded Character Set (816CS00.00)

EBCDICUTL Coded Character Set (816CS00.00)

EBCDICUTL is a mixed, multibyte coded character set used on Taiwanese ClearPath NX and A Series servers. This coded character set consists of

- The 8-bit coded character set ASeriesEBCDIC, which has characters scattered throughout byte range x00–xFF. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSuppIIISO,” for a complete description of the ASeriesEBCDIC coded character set.
- A 16-bit coded character set defined by Unisys Taiwan Limited (UTL) that represents over 19,000 Taiwanese characters. The server systems encoding places these characters throughout the byte pair ranges (x41–x9E, x41–x9E), (x41–x9E, xA1–xFE), (xA1–xFB, x41–x9E), and (xA1–xFB, xA1–xFE).

The 8-bit space character in the ASeriesEBCDIC coded character set is located at code position x40. The 16-bit ideographic space character in the UTL coded character set is located at code position x41A1.

Within a sequence of characters, all of the 16-bit data is bracketed by the 8-bit SDO and EDO characters, byte values x2B and x2C respectively, to signify the start and end of the 16-bit data.

The code positions for EBCDICUTL are not currently defined. No translation tables exist to support mappings to or from this character set.

8-bit ASeriesEBCDIC (x00–xFF) +

Sample Character Encodings:

None

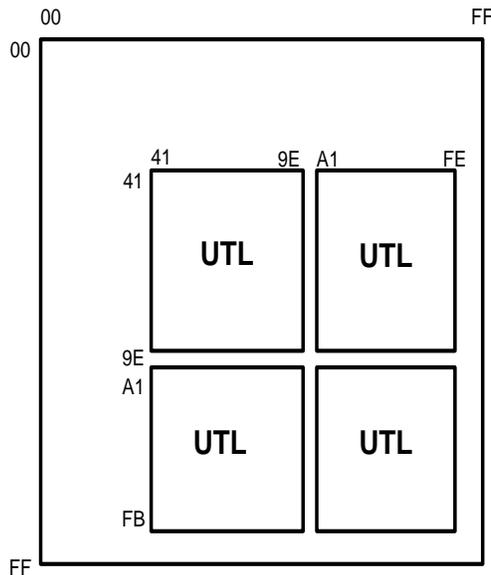


Figure 13–13. EBCDICUTL Coded Character Set (816CS00.00)

EUCJp Coded Character Set (816CS00.00)

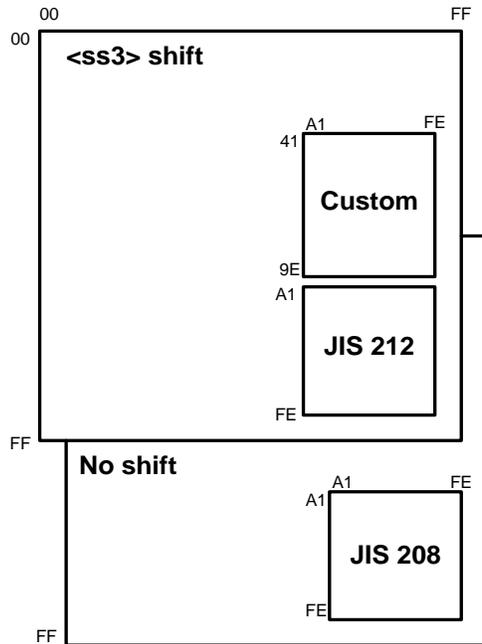
EUCJp is a mixed, multibyte coded character set used to represent the Extended UNIX Code for Japanese, also known as UNIX System AJEC—Advanced Japanese EUC Code. This coded character set consists of

- The 8-bit ASCII coded character set, which has characters encoded in the range x00–x7F. Refer to Section 10, “Eight-Bit Coded Character Sets: Arabic20E to CANSupplISO,” for a complete description of the ASCII coded character set.
- An 8-bit coded character set based on the Japanese Industrial Standard JISX0201 coded character set, which represents 63 Katakana characters encoded in the byte range xA1–xDF. This 8-bit data is preceded by the “SS2” single shift control character, which indicates the width and code set of the next character in the string. The SS2 control character has the byte value x8E and must precede every character to which it applies.
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The UNIX encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The UNIX encoding places the JISX0212 characters in the byte pair range (xA1–xFE, xA1–xFE). This 16-bit data is preceded by the “SS3” single shift control character, which indicates the width and code set of the next character in the string. The SS3 control character has the byte value x8F and must precede every character to which it applies.
- Other 16-bit customer-defined custom Japanese characters. The UNIX encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE). This 16-bit data is preceded by the “SS3” single shift control character, which indicates the width and code set of the next character in the string. The SS3 control character has the byte value x8F and must precede every character to which it applies.

The 8-bit space character in the ASCII coded character set is located at code position x20. The 16-bit ideographic space character in the JISX0208 coded character set is located at code position xA1A1.

Double-Byte and Mixed, Multibyte Coded Character Sets

8-bit ASCII (x00-x7F) +
 <ss2> JISX0201 (x8E followed by xA1-xDF) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A5 Fullwidth full stop
- xA1A6 Katakana middle dot
- xA2BF Breve
- xA2B0 Caron
- xA2B1 Right-pointing double angle quotation mark
- xA2B2 Dot above

Figure 13-14. EUCJp Coded Character Set (816CS00.00)

JapanEBCDICJBIS8 Coded Character Set (816CS00.00)

JapanEBCDICJBIS8 is a mixed, multibyte coded character set that is used on Japanese ClearPath NX and A Series servers. This coded character set consists of

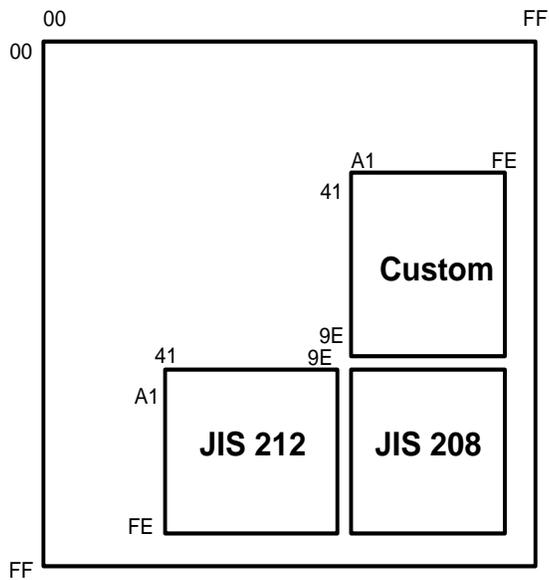
- The 8-bit Japanese coded character set defined as JapanEBCDIC. JapanEBCDIC is a combination of the ASeriesEBCDIC coded character set and the Japanese Industrial Standard JISX0201 coded character set, which represents 63 Katakana characters placed where no EBCDIC characters are defined. The 8-bit ASeriesEBCDIC and Katakana characters are scattered throughout the 8-bit byte range x00–xFF. Refer to Section 12, “Eight-Bit Coded Character Sets: FrenchArabicE to NorwayBTOS,” for a complete description of the JapanEBCDIC coded character set.
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, x41–x9E).
- Other 16-bit customer-defined custom Japanese characters. The server systems encoding places these characters in the byte pair range (x41–x9E, xA1–xFE).

The 8-bit space character in the JapanEBCDIC coded character set is located at code position x40. The 16-bit ideographic space character in the JISX0208 coded character set is located at code position xA1A1.

Within a sequence of characters, all of the 16-bit data is bracketed by the 8-bit SDO and EDO characters, byte values x2B and x2C respectively, to signify the start and end of the 16-bit data.

Double-Byte and Mixed, Multibyte Coded Character Sets

8-bit JapanEBCDIC (x00-xFF) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A5 Fullwidth full stop
- xA1A6 Katakana middle dot
- xA24F Breve
- xA250 Caron
- xA251 Right-pointing double angle quotation mark
- xA252 Dot above

Figure 13-15. JapanEBCDICJIS8 Coded Character Set (816CS00.00)

JapanV24JBIS8 Coded Character Set (816CS00.00)

JapanV24JBIS8 is a mixed, multibyte coded character set that is used on Japanese ClearPath NX and A Series servers. This coded character set consists of

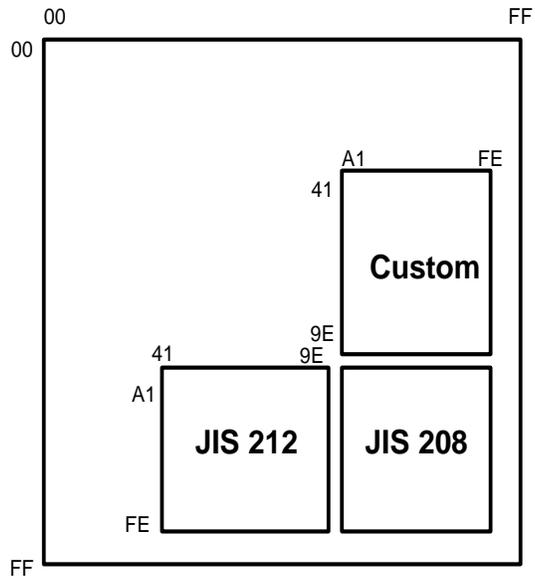
- The 8-bit Japanese coded character set defined as JapanV24. JapanV24 is a combination of EBCDIC characters and 63 Katakana characters. It is a non-standard coded character set which has legacy usage in a lot of Japanese data. It is somewhat based on the ASeriesEBCDIC coded character set and the Japanese Industrial Standard JISX0201 coded character set, except that the Katakana characters in JISX0201 have been placed in code positions such that they replace the EBCDIC lower-case Latin characters “a” through “z”. (These lower-case characters may be transplanted to other positions.) The 8-bit EBCDIC and Katakana characters are scattered throughout the 8-bit byte range x00–xFF. Refer to Section 12, “Eight-Bit Coded Character Sets: FrenchArabicE to NorwayBTOS,” for a complete description of the JapanV24 coded character set.
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, x41–x9E).
- Other 16-bit customer-defined custom Japanese characters. The server systems encoding places these characters in the byte pair range (x41–x9E, xA1–xFE).

The 8-bit space character in the JapanV24 coded character set is located at code position x40. The 16-bit ideographic space character in the JISX0208 coded character set is located at code position xA1A1.

Within a sequence of characters, all of the 16-bit data is bracketed by the 8-bit SDO and EDO characters, byte values x2B and x2C respectively, to signify the start and end of the 16-bit data.

Double-Byte and Mixed, Multibyte Coded Character Sets

8-bit JapanV24 (x00-xFF) +



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A5 Fullwidth full stop
- xA1A6 Katakana middle dot
- xA24F Breve
- xA250 Caron
- xA251 Right-pointing double angle quotation mark
- xA252 Dot above

Figure 13-16. JapanV24JBIS8 Coded Character Set (816CS00.00)

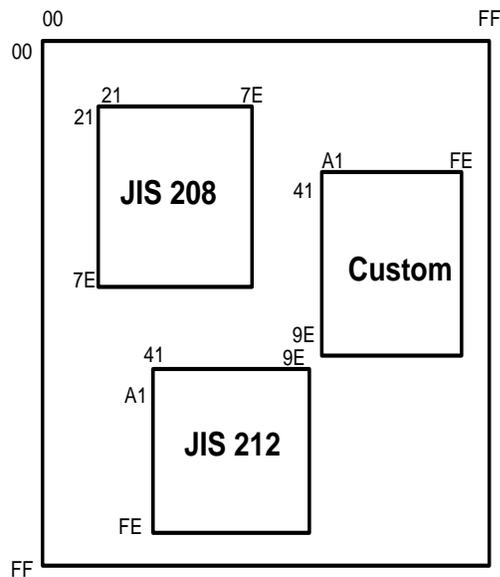
JBIS7 Coded Character Set (16CS00.00)

JBIS7 is a pure 16-bit coded character set for use on Japanese ASCII-based terminals. This coded character set consists of

- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The ASCII terminal encoding places these characters in the byte pair range (x21–x7E, x21–x7E).
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The ASCII terminal encoding places these characters in the byte pair range (xA1–xFE, x41–x9E).
- Other 16-bit customer-defined custom Japanese characters. The ASCII terminal encoding places these characters in the byte pair range (x41–x9E, xA1–xFE).

The 16-bit space character (ideographic space) in the JISX0208 coded character set is located at code position x2121.

The JBIS7 coded character set is used when double-byte characters are needed without the surrounding bracket characters SDO and EDO, such as when double-byte characters are embedded in a COBOL85 record and defined as National characters.



Sample Character Encodings:

x2121	Ideographic space
x2122	Ideographic comma
x2125	Fullwidth full stop
x2126	Katakana middle dot
xA24F	Breve
xA250	Caron
xA251	Right-pointing double angle quotation mark
xA252	Dot above

Figure 13–17. JBIS7 Coded Character Set (16CS00.00)

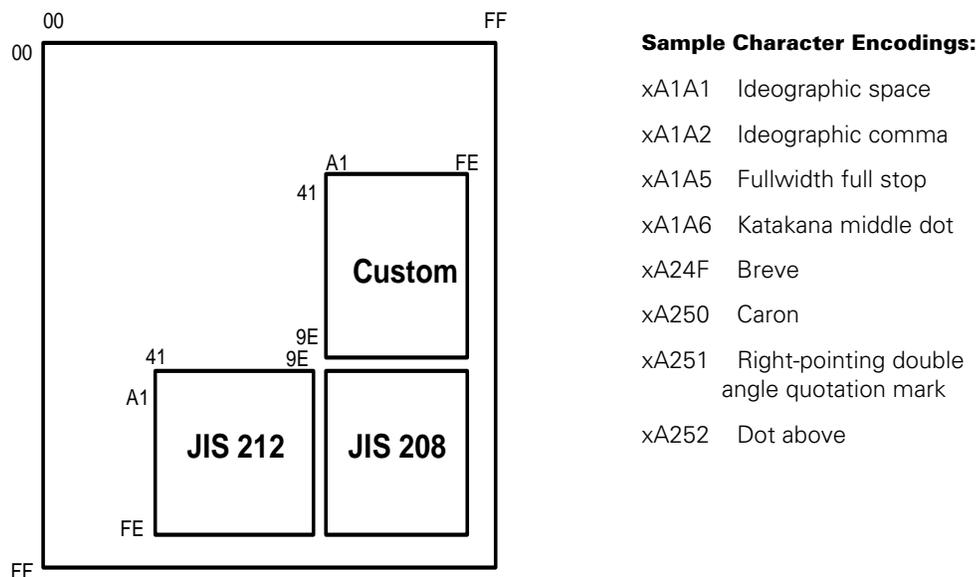
JBIS8 Coded Character Set (16CS00.00)

JBIS8 is a pure 16-bit coded character set for use on Japanese ClearPath NX and A Series server systems. This coded character set consists of

- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, x41–x9E).
- Other 16-bit customer-defined custom Japanese characters. The server systems encoding places these characters in the byte pair range (xA1–xFE, x41–x9E).

The 16-bit space character (ideographic space) in the JISX0208 coded character set is located at code position xA1A1.

The JBIS8 coded character set is used when double-byte characters are needed without the surrounding bracket characters SDO and EDO, such as when double-byte characters are embedded in a COBOL85 record and defined as National characters.



Sample Character Encodings:

- xA1A1 Ideographic space
- xA1A2 Ideographic comma
- xA1A5 Fullwidth full stop
- xA1A6 Katakana middle dot
- xA24F Breve
- xA250 Caron
- xA251 Right-pointing double angle quotation mark
- xA252 Dot above

Figure 13–18. JBIS8 Coded Character Set (16CS00.00)

JISASCIIJBIS7 Coded Character Set (816CS00.00)

JISASCIIJBIS7 is a mixed, multibyte coded character set used on Japanese ASCII-based terminals. This coded character set consists of

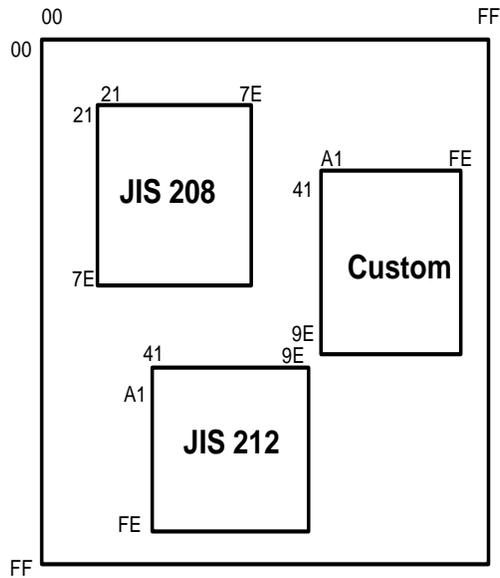
- The 8-bit Japanese coded character set defined as JISASCII. JISASCII is a combination of ASCII and the Japanese Industrial Standard JISX0201 coded character set, which represents 63 Katakana characters placed where no ASCII characters are defined. The 8-bit ASCII characters are encoded in the byte range x00–x7F. The 8-bit Katakana characters are encoded in the byte range xA1–xDF. Refer to Section 12, “Eight-Bit Coded Character Sets: FrenchArabicE to NorwayBTOS,” for a complete description of the JISASCII coded character set.
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The ASCII terminal encoding places these characters in the byte pair range (x21–x7E, x21–x7E).
- A 16-bit character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The ASCII terminal encoding places these characters in the byte pair range (xA1–xFE, x41–x9E).
- Other 16-bit customer-defined custom Japanese characters. The ASCII terminal encoding places these characters in the byte pair range (x41–x9E, xA1–xFE).

The 8-bit space character in the JISASCII coded character set is located at code position x20. The 16-bit ideographic space character in the JISX0208 coded character set is located at code position x2121.

Within a sequence of characters, all of the 16-bit data is bracketed by the ASCII encoding of the 8-bit SDO and EDO characters, byte values x9E and x9F respectively, to signify the start and end of the 16-bit data.

Double-Byte and Mixed, Multibyte Coded Character Sets

8-bit JISASCII (x00-x7F and xA1-xDF) +



Sample Character Encodings:

- x2121 Ideographic space
- x2122 Ideographic comma
- x2125 Fullwidth full stop
- x2126 Katakana middle dot
- xA24F Breve
- xA250 Caron
- xA251 Right-pointing double angle quotation mark
- xA252 Dot above

Figure 13-19. JISASCIIJBIS7 Coded Character Set (816CS00.00)

LetsJ Coded Character Set (816CS00.00)

LetsJ is a mixed, multibyte coded character set used on Japanese ClearPath IX systems to represent 8-bit ASCII and Katakana characters and 16-bit Kanji characters. This coded character set consists of

- The 8-bit Japanese coded character set defined as JISASCII. JISASCII is a combination of the ASCII coded character set and the Japanese Industrial Standard JISX0201 coded character set, which represents 63 Katakana characters placed where no ASCII characters are defined. The 8-bit ASCII characters are encoded in the byte range x00–x7F. The 8-bit Katakana characters are encoded in the byte range xA1–xDF. Refer to Section 12, “Eight-Bit Coded Character Sets: FrenchArabicE to NorwayBTOS,” for a complete description of the JISASCII coded character set.
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0208 coded character set, which represents over 6,000 basic Japanese Kanji and other graphic characters. The LetsJ encoding places these characters in the byte pair range (xA1–xFE, xA1–xFE).
- A 16-bit coded character set based on the Japanese Industrial Standard JISX0212 coded character set, which represents over 6,000 supplementary Japanese Kanji and other graphic characters. The LetsJ encoding places these characters in the byte pair range (xA1–xFE, x21–x7E).
- Other 16-bit customer-defined custom Japanese characters. The LetsJ encoding places these characters in the byte pair range (x21–x7E, xA1–xFE).

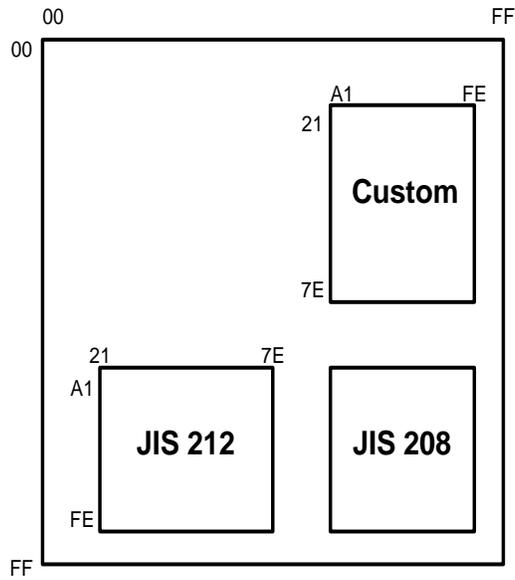
The 8-bit space character in the JISASCII coded character set is located at code position x20. The 16-bit ideographic space character in the JISX0208 coded character set is located at code position x2020.

Within a sequence of characters, all of the 16-bit character data must be preceded by the two-byte lock shift character x93nn, where “nn” is an **even**-valued hex number, to signal the beginning of a 16-bit character string. Once in the 16-bit mode, the two-byte lock shift character x93nn, where “nn” is an **odd**-valued hex number, is required to signal the beginning of 8-bit data.

Unlike the SDO/EDO mechanism of JapanEBCDICJBIS8 and JISASCIIJSBIS7, there is no need to put a lock shift character at the end of a string. A lock shift character is only needed when switching from one mode to another. The current mode of input, 8-bit or 16-bit, continues until the end of the string or until another lock shift character is encountered.

Double-Byte and Mixed, Multibyte Coded Character Sets

8-bit JISASCII (x00-x7F and xA1-xDF) +



Sample Character Encodings:

- x2020 Ideographic space
- xA1A2 Ideographic comma
- xA1A5 Fullwidth full stop
- xA1A6 Katakana middle dot
- xA22F Breve
- xA230 Caron
- xA231 Right-pointing double angle quotation mark
- xA232 Dot above

Figure 13-20. LetsJ Coded Character Set (816CS00.00)

UCS2 Coded Character Set (16CS00.00)

UCS2 is a pure 16-bit coded character set that uses a fixed-width encoding scheme to represent international character codes for written characters, texts for the major scripts of the world, and technical symbols in common use. This coded character set is based on the *International Standard ISO/IEC 10646-1: 1993, Information Technology—Universal Multiple-Octet Coded Character Set (UCS)—Part 1: Architecture and Basic Multilingual Plane*. The characters are encoded in the byte range x0000–xFFFD. For more information, refer to the ISO/IEC 10646-1 standard or to the Unicode Standard provided by the Unicode Consortium.

Various machine architectures have differed in the ordering of the most significant byte and least significant byte of the encoding. For example, the ISO10646 character, Latin capital letter A, can be represented as x0041 or x4100. Leading with the most significant byte is known as big-endian and is the standard definition. Leading with the least significant byte is known as little-endian and is used on Intel processors, Windows 95, and all versions of Windows NT. **UCS2 is the big-endian encoding.** In the example above, x0041 would be the big-endian encoding for U+0041.

The UCS2 coded character set uses the entire x0000–xFFFF plane, although not all code positions are assigned.

The space character in the UCS2 coded character set is located at code position x0020; the ideographic space character is located at code position x3000.

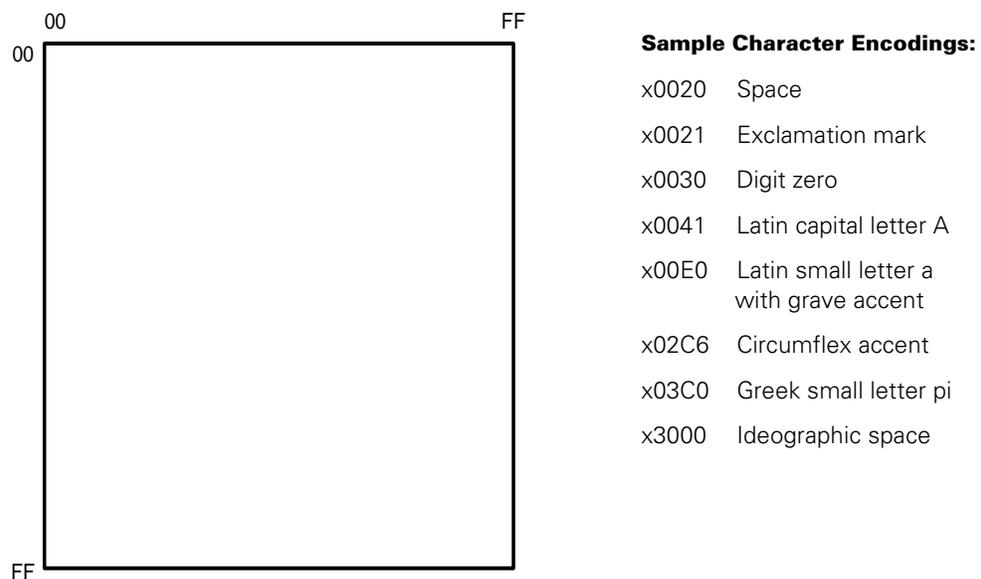


Figure 13–21. UCS2 Coded Character Set (16CS00.00)

UCS2NT Coded Character Set (16CS00.00)

UCS2NT is a pure 16-bit coded character set that uses a fixed-width encoding scheme to represent international character codes for written characters, texts for the major scripts of the world, and technical symbols in common use. This coded character set is based on the *International Standard ISO/IEC 10646-1: 1993, Information Technology—Universal Multiple-Octet Coded Character Set (UCS)—Part 1: Architecture and Basic Multilingual Plane*. The characters are encoded in the byte range x0000–xFFFD. For more information, refer to the ISO/IEC 10646-1 standard or to the Unicode Standard provided by the Unicode Consortium.

Various machine architectures differ in the ordering of the most significant byte and least significant byte of the encoding. For example, the ISO10646 character, Latin capital letter A, can be represented as x0041 or x4100. Leading with the most significant byte is known as big-endian and is the standard definition of the Unicode Standard. Leading with the least significant byte is known as little-endian and is used on Intel processors, Windows 95, and all versions of Windows NT. **UCS2NT is the little-endian encoding.** In the example above, x4100 would be the little-endian encoding for U+0041.

The UCS2NT coded character set uses the entire x0000–xFFFF plane, although not all code positions are assigned.

The space character in the UCS2NT coded character set is located at code position x2000; the ideographic space character is located at code position x0030.

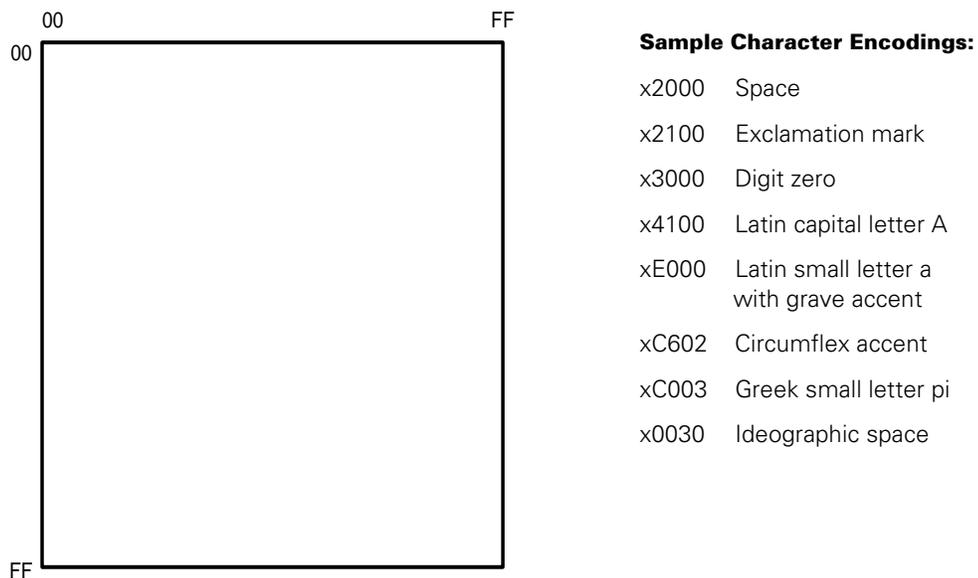


Figure 13-22. UCS2NT Coded Character Set (16CS00.00)

Section 14

Ccsversions

This section presents ccsversion definitions. Table 14–1 alphabetically lists the ccsversions supplied in the current release. Each ccsversion is identified by name and number. You can use the name or number to establish a ccsversion.

Table 14–1 also lists the base coded character set for the ccsversion. Several ccsversions can share a base coded character set. For example, Brazilian, CanadaEBCDIC, Norway, Spanish, Swedish1, and Swiss all use Latin1EBCDIC as the base coded character set.

Table 14–1. Ccsversions in the Current Release

Ccsversion Name	Ccsversion Number	Base Coded Character Set
Arabic20 <i>Note: Arabic 20 EBCDIC was previously referred to as Versin 20 EBCDIC</i>	20	Arabic20EBCDIC
ASERIESNATIVE	0	ASERIESEBCDIC
Brazilian	45	Latin1EBCDIC
CanadaEBCDIC	74	Latin1EBCDIC
CanadaGP	75	CANSupplEBCDIC
Czechoslovakia	53	Latin2EBCDIC
France	35	Latin5EBCDIC
Hungarian	85	Latin2EBCDIC
LatinGreek	49	LatinGreekEBCDIC
Norway	71	Latin1EBCDIC
Polish	86	Latin2EBCDIC
Romanian	96	Latin2EBCDIC
Russian	88	LatinCyrillicEBC
Spanish	98	Latin1EBCDIC

Table 14–1. Ccsversions in the Current Release

Ccsversion Name	Ccsversion Number	Base Coded Character Set
Swedish1	99	Latin1EBCDIC
Swiss	64	Latin1EBCDIC
Turkish	80	Latin5EBCDIC

Tables 14–2 through 14–17 contain ccsversion definitions. Each table contains the ccsversion elements and their corresponding characters. The escapement definitions for each ccsversion are listed after the appropriate table.

Arabic20 Ccsversion Definitions

Table 14–2 lists the definitions for the Arabic20 ccsversion. The ccsversion number for Arabic20 is 20. The coded character set base is Arabic20EBCDIC.

Table 14–2. Arabic20 Ccsversion Definitions

Elements	Definitions
Spaces data class	Space, Arabic space
Alphabetic data class	<p>Lowercase and uppercase forms of the following:</p> <p>Latin letters:</p> <p>a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z,</p> <p>The following Arabic letters:</p> <p>lam alef, hamza, yeh with hamza above, alef, alef maksura, beh, teh, teh marbuta, theh, jeem, hah, khah, dal, thal, reh, zain, seen, sheen, sad, dad, tah, zah, ain, ghain, feh, qaf, kaf, lam, meem, noon, heh, waw, yeh, tatweel</p>
Numeric data class	<p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>The following Arabic-Indic numbers:</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p>
Presentation set	<p>Lowercase and uppercase forms of the following:</p> <p>Latin letters:</p> <p>a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z,</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ , >, ?, grave accent, colon, #, @, apostrophe, =, ", tilde, {, }, \, number acronymn, soft hyphen, paragraph sign</p> <p>The following Arabic characters:</p> <p>lam alef, hamza, yeh with hamza above, alef, alef maksura, beh, teh, teh marbuta, theh, jeem, hah, khah, dal, thal, reh, zain, seen, sheen, sad, dad, tah, zah, ain, ghain, feh, qaf, kaf, lam, meem, noon, heh, waw, yeh, tatweel</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, space, full stop, +, !, numeric comma, *, semicolon, left double angle quote, right double angle quote, -, /, comma, %, ?, colon, =</p>
Lowercase set	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z,
Uppercase set	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z,
Lowercase-to-uppercase mapping	a-A, b-B, c-C, d-D, e-E, f-F, g-G, h-H, i-I, j-J, k-K, l-L, m-M, n-N, o-O, p-P, q-Q, r-R, s-S, t-T, u-U, v-V, w-W, x-X, y-Y, z-Z,
Uppercase-to-lowercase mapping	A-a, B-b, C-c, D-d, E-e, F-f, G-g, H-h, I-i, J-j, K-k, L-l, M-m, N-n, O-o, P-p, Q-q, R-r, S-s, T-t, U-u, V-v, W-w, X-x, Y-y, Z-z,

Table 14-2. Arabic20 Ccsversion Definitions

Elements	Definitions
Character collating	<p>Note: in the character collating sequence, the series of characters within square brackets are given the same rank in equivalent ordering.</p> <p>space, arabic space, [a, A], [b, B], [c, C], [d, D], [e, E], [f, F], [g, G], [h, H], [i, I], [j, J], [k, K], [l, L], [m, M], [n, N], [o, O], [p, P], [q, Q], [r, R], [s, S], [t, T], [u, U], [v, V], [w, W], [x, X], [y, Y], [z, Z]</p> <p>Forms of the following Arabic letters:</p> <p>tatweel, hamza, yeh with hamza above, alef, beh, teh marbuta, teh, theh, jeem, hah, khah, dal, thal, reh, zain, seen, sheen, sad, dad, tah, zah, ain, ghain, feh, qaf, kaf, lam, meem, noon, heh, waw, yeh,</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [full stop, arabic full stop], [comma, arabic comma, arabic numeric comma], [?, arabic question mark], [semicolon, arabic semicolon], [!, arabic exclamation point], [colon, arabic colon], apostrophe, grave accent, [-, arabic hyphenminus], _, vertical line, [/, arabic solidus], \, left double angle quote, right double angle quote, (,), left bracket, right bracket, <, >, {, }, apersand, @, [%, arabic percent sign], \$, [=, arabic equal sign], [+ , arabic plus sign], circumflex accent, *, #, tilde</p>
Substitutions	lam alef is ordered as a lam followed by alef
Alternate digits	0-arabic indic 0, 1-arabic indic 1, 2-arabic indic 2, 3-arabic indic 3, 4-arabic indic 4, 5-arabic indic 5, 6-arabic indic 6, 7-arabic indic 7, 8-arabic indic 8, 9-arabic indic 9

The following escapement definitions apply to the Arabic20 ccsversion:

Character advance direction	All control characters (K), [(K), < (K), left parenthesis (K),] (K), right parenthesis (K), > (K), { (K) } (K), APC (K), MD (K), all other arabic characters right-to-left (-), other characters left-to-right (+)
Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)

ASeriesNative Ccsversion Definitions

Table 14–3 lists the definitions for the ASeriesNative ccsversion. The ccsversion number for ASeriesNative is 0. The coded character set base is ASERIESEBCDIC.

Table 14–3. ASeriesNative Ccsversion Definitions

Elements	Characters
Spaces data class	Space
Alphabetic data class	Lowercase and uppercase forms of the following Latin letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	Lowercase and uppercase forms of the following Latin letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \
Lowercase set	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z
Uppercase set	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z
Lowercase to uppercase mapping	a-A, b-B, c-C, d-D, e-E, f-F, g-G, h-H, i-I, j-J, k-K, l-L, m-M, n-N, o-O, p-P, q-Q, r-R, s-S, t-T, u-U, v-V, w-W, x-X, y-Y, z-Z
Uppercase to lowercase mapping	A-a, B-b, C-c, D-d, E-e, F-f, G-g, H-h, I-i, J-j, K-k, L-l, M-m, N-n, O-o, P-p, Q-q, R-r, S-s, T-t, U-u, V-v, W-w, X-x, Y-y, Z-z
Character collating	NUL, SOH, STX, ETX, HT, DEL, VT, FF, CR, SO, SI DLE, DC1, DC2, DC3, BS, CAN, EM, FS, GS, RS, US LF, ETB, ESC, ENQ, ACK, BEL, SYN, EOT, DC4, NAK SUB, space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ >, ?, grave accent, colon, #, @, apostrophe, =, ", a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, tilde, s, t, u, v, w, x, y, z, {, A, B, C, D, E, F, G, H, I, }, J, K, L, M, N, O, P, Q, R, \, APC, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, MD
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the ASeriesNative ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Brazilian Ccsversion Definitions

Table 14–4 lists the definitions for the Brazilian ccsversion. The ccsversion number for Brazilian is 45. The coded character set base is Latin1EBCDIC.

Table 14–4. Brazilian Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters: a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>Soft hyphen</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters: a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, no-break space, left double-angle quote, right double-angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, soft hyphen, superscript 1, superscript 2, superscript 3, division sign</p>
Lowercase set	a, a circumflex, a grave, a acute, a tilde, a with e, b, c, c cedilla, d, e, e circumflex, e acute, f, g, h, i, i acute, j, k, l, m, n, o, o circumflex, o acute, p, q, r, s, t, u, u diaeresis, u acute, v, w, x, y, z

Table 14-4. Brazilian Ccsversion Definitions

Elements	Characters
Uppercase set	A, A circumflex, A grave, A acute, A tilde, A with E, B, C, C cedilla, D, E, E circumflex, E acute, F, G, H, I, I acute, J, K, L, M, N, O, O circumflex, O acute, P, Q, R, S, T, U, U diaeresis, U acute, V, W, X, Y, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a grave-A grave, a acute-A acute, a tilde-A tilde, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, e-E, e circumflex-E circumflex, e acute-E acute, f-F, g-G, h-H, i-I, i acute-I acute, j-J, k-K, l-L, m-M, n-N, o-O, o circumflex-O circumflex, o acute-O acute, p-P, q-Q, r-R, s-S, t-T, u-U, u diaeresis-U diaeresis, u acute-U acute, v-V, w-W, x-X, y-Y, z-Z
Uppercase to lowercase mapping	A-a, A circumflex-a circumflex, A grave-a grave, A acute-a acute, A tilde-a tilde, A with E-a with e, B-b, C-c, C cedilla-c cedilla, D-d, E-e, E circumflex-e circumflex, E acute-E acute, F-f, G-g, H-h, I-i, I acute-i acute, J-j, K-k, L-l, M-m, N-n, O acute-o acute, P-p, Q-q, R-r, S-s, T-t, U-u, U diaeresis-u diaeresis, U acute-u acute, V-v, W-w, X-x, Y-y, Z-z
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>Space, [A, a, A acute, a acute, A grave, a grave, A circumflex, a circumflex, A diaeresis, a diaeresis, A tilde, a tilde, A ring above, a ring above], [B, b], [C, c, C cedilla, c cedilla], [D, d], [E, e, E acute, e acute, E grave, e grave, E circumflex, e circumflex, E diaeresis, e diaeresis], [F, f], [G, g], [H, h], [I, i, I acute, i acute, I grave, i grave, I circumflex, i circumflex, I diaeresis, i diaeresis], [J, j], [K, k], [L, l], [M, m], [N, n, N tilde, n tilde], [O, o, O acute, o acute, O grave, o grave, O circumflex, o circumflex, O diaeresis, o diaeresis, O tilde, o tilde, O with oblique stroke, o with oblique stroke], [P, p], [Q, q] [R, r], [S, s, German sharp s], [T, t], [U, u, U acute, u acute, U grave, u grave, U circumflex, u circumflex, U diaeresis, u diaeresis], [V, v], [W, w], [X, x], [Y, y, Y acute, y acute, y diaeresis], [Z, z], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, >, ?, grave accent, colon, #, @, apostrophe, =, ", left double angle quote, right double angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, tilde, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, {, }, superscript 1, \, superscript 2, superscript 3, division sign, uppercase Icelandic eth, lowercase Icelandic eth, uppercase Icelandic thorn, lowercase Icelandic thorn</p>

Table 14-4. Brazilian Ccsversion Definitions

Elements	Characters
Substitutions	A with E is ordered as A followed by E. a with e is ordered as a followed by e.
Alternate digits	No concept

The following escapement definitions apply to the Brazilian ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	Circumflex accent 0, grave accent 0, diaeresis 0, acute accent 0, other characters left-to-right (+)

CanadaEBCDIC Ccsversion Definitions

Table 14–5 lists the definitions for the CanadaEBCDIC ccsversion. The ccsversion number for CanadaEBCDIC is 74. The coded character set base is Latin1EBCDIC.

Table 14–5. CanadaEBCDIC Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>Soft hyphen</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, no-break space, left double-angle quote, right double-angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, soft hyphen, superscript 1, superscript 2, superscript 3, division sign</p>

Table 14-5. CanadaEBCDIC Ccsversion Definitions

Elements	Characters
Lowercase set	a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, lowercase Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, lowercase Icelandic thorn, q, r, s, German sharp s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y diaeresis, y acute, z
Uppercase set	A, A circumflex, A diaeresis, A grave, A acute, A tilde, A ring above, A with E, B, C, C cedilla, D, uppercase Icelandic eth, E, E circumflex, E diaeresis, E grave, E acute, F, G, H, I, I circumflex, I diaeresis, I grave, I acute, J, K, L, M, N, N tilde, O, O circumflex, O diaeresis, O grave, O acute, O tilde, O with oblique stroke, P, uppercase Icelandic thorn, Q, R, S, T, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Y acute, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a diaeresis-A diaeresis, a grave-A grave, a acute-A acute, a tilde-A tilde, a ring above-A ring above, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, lowercase Icelandic eth-uppercase Icelandic eth, e-E, e circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, i grave-I grave, i acute-I acute, j-J, k-K, l-L, m-M, n-N, n tilde-N tilde, o-O, o circumflex-O circumflex, o diaeresis-O diaeresis, o grave-O grave, o acute-O acute, o tilde-O tilde, o with oblique stroke-O with oblique stroke, p-P, lowercase Icelandic thorn-uppercase Icelandic thorn, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, u acute-U acute, v-V, w-W, x-X, y-Y, y acute-Y acute, y diaeresis-Y, z-Z
Uppercase to lowercase mapping	A-a, A circumflex-a circumflex, A diaeresis-a diaeresis, A grave-a grave, A acute-a acute, A tilde-a tilde, A ring above-a ring above, A with E-a with e, B-b, C-c, C cedilla-c cedilla, D-d, uppercase Icelandic eth-lowercase Icelandic eth, E-e, E circumflex-e circumflex, E diaeresis-e diaeresis, E grave-e grave, E acute-e acute, F-f, G-g, H-h, I-i, I circumflex-i circumflex, I diaeresis-i diaeresis, I grave-i grave, I acute-i acute, J-j, K-k, L-l, M-m, N-n, N tilde-n tilde, O-o, O circumflex-o circumflex, O diaeresis-o diaeresis, O grave-o grave, O acute-o acute, O tilde-o tilde, O with oblique stroke-o with oblique stroke, P-p, lowercase Icelandic thorn-uppercase Icelandic thorn, Q-q, R-r, S-s, T-t, U-u, U circumflex-u circumflex, U diaeresis-u diaeresis, U grave-u grave, U acute-u acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z

Table 14–5. CanadaEBCDIC Ccsversion Definitions

Elements	Characters
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>NUL, SOH, STX, ETX, ST, HT, SSA, APC, EPA, RI, SS2, VT, FF, CR, DLE, DC1, DC2, DC3, OSC, NEL, BS, ESA, CAN, EM, PU2, SS3, IS4, IS3, IS2, IS1, BPH, NBH, IND, LF, ETB, ESC, HTS, HTJ, VTS, PLD PLU, ENQ, ACK, BEL, DCS, PU1, SYN, STS, CCH, MW, SPA, EOT, SOS, SCI, CSI, DC4, NAK, PM, SUB, space, [A, a, A acute, a acute, A grave, a grave, A circumflex, a circumflex, A diaeresis, a diaeresis, A tilde, a tilde, A ring above, a ring above], [B, b], [C, c, C cedilla, c cedilla], [D, d], [E, e, E acute, e acute, E grave, e grave, E circumflex, e circumflex, E diaeresis, e diaeresis], [F, f] [G, g], [H, h], [I, i, I acute, i acute, I grave, i grave, I circumflex, i circumflex, I diaeresis, i diaeresis], [J, j], [K, k], [L, l], [M, m], [N, n, N tilde, n tilde], [O, o, O acute, o acute, O grave, o grave, O circumflex, o circumflex, O diaeresis, o diaeresis, O tilde, o tilde, O with oblique stroke, o with oblique stroke], [P, p], [Q, q] [R, r], [S, s, German sharp s], [T, t], [U, u, U acute, u acute, U grave, u grave, U circumflex, u circumflex, U diaeresis, u diaeresis], [V, v], [W, w], [X, x], [Y, y, Y acute, y acute, y diaeresis], [Z, z], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, >, ?, grave accent, colon, #, @, apostrophe, =, ", left double angle quote, right double angle quote, [uppercase Icelandic eth, lowercase Icelandic eth], [uppercase Icelandic thorn, lowercase Icelandic thorn], plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, tilde, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, {, }, superscript 1, superscript 2, superscript 3, \, division sign, DEL</p>
Substitutions	<p>A with E is ordered as A followed by E.</p> <p>a with e is ordered as a followed by e.</p>
Alternate digits	No concept

The following escapement definitions apply to the CanadaEBCDIC ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

CanadaGP Ccsversion Definitions

Table 14–6 lists the definitions for the CanadaGP ccsversion. The ccsversion number for CanadaGP is 75. The coded character set base is CANSuppEBCDIC.

Table 14–6. CanadaGP Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a grave, a acute, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i acute, j, k, l, m, n, n tilde, o, o circumflex, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, z</p> <p>Soft hyphen</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a grave, a acute, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i acute, j, k, l, m, n, n tilde, o, o circumflex, o acute, p, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, z</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, no-break space, left double-angle quote, right double-angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, diaeresis, acute accent, soft hyphen, superscript 1, superscript 2, superscript 3, superscript 4, superscript 5, superscript 6, superscript 7, superscript 8, superscript 9, superscript 0, superscript +, superscript -, superscript (, superscript), left single quotation mark, right single quotation mark, left double quotation mark, right double quotation mark, less-than or equal sign, greater-than or equal sign, not equal sign, full block</p> <p>The following box drawings characters:</p> <p>Light horizontal, light vertical, light down and right, light down and left, light up and right, light up and left, light vertical and right, light vertical and left, light down and horizontal, light up and horizontal, light vertical and horizontal</p>

Table 14-6. CanadaGP Ccsversion Definitions

Elements	Characters
Lowercase set	a, a circumflex, a grave, a acute, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i acute, j, k, l, m, n, n tilde, o, o circumflex, o acute, p, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, z
Uppercase set	A, A circumflex, A grave, A acute, B, C, C cedilla, D, E, E circumflex, E diaeresis, E grave, E acute, F, G, H, I, I circumflex, I diaeresis, I acute, J, K, L, M, N, N tilde, O, O circumflex, O acute, P, Q, R, S, T, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a grave-A grave, a acute-A acute, b-B, c-C, c cedilla-C cedilla, d-D, e-E, e-circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, i acute-I acute, j-J, k-K, l-L, m-M, n-N, n tilde-N tilde, o-O, o circumflex-O circumflex, o acute-O acute, p-P, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, u acute-U acute, v-V, w-W, x-X, y-Y, z-Z
Uppercase to lowercase mapping	A-a, A circumflex-a circumflex, A grave-a grave, A acute-a acute, B-b, C-c, C cedilla-c cedilla, D-d, E-e, E-circumflex-e circumflex, E diaeresis-e diaeresis, E grave-e grave, E acute-e acute, F-f, G-g, H-h, I-i, I circumflex-i circumflex, I diaeresis-i diaeresis, I acute-i acute, J-j, K-k, L-l, M-m, N-n, N tilde-n tilde, O-o, O circumflex-o circumflex, O acute-o acute, P-p, Q-q, R-r, S-s, T-t, U-u, U circumflex-u circumflex, U diaeresis-u diaeresis, U grave-u grave, U acute-u acute, V-v, W-w, X-x, Y-y, Z-z
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>NUL, SOH, STX, ETX, ST, HT, SSA, APC, EPA, RI, SS2, VT, FF, CR, DLE, DC1, DC2, DC3, OSC, NEL, BS, ESA, CAN, EM, PU2, SS3, IS4, IS3, IS2, IS1, BPH, NBH, IND, LF, ETB, ESC, HTS, HTJ, VTS, PLD, PLU, ENQ, ACK, BEL, DCS, PU1, SYN, STS, CCH, MW, SPA, EOT, SOS, SCI, CSI, DC4, NAK, PM, SUB,</p> <p>space,</p> <p>[A, a, A acute, a acute, A grave, a grave, A circumflex, a circumflex],</p> <p>[B, b], [C, c, C cedilla, c cedilla], [D, d],</p> <p>[E, e, E acute, e acute, E grave, e grave, E circumflex, e circumflex, E diaeresis, e diaeresis], [F, f], [G, g], [H, h],</p> <p>[I, i, I acute, i acute, I circumflex, i circumflex, I diaeresis, i diaeresis],</p>

Table 14–6. CanadaGP Ccsversion Definitions

Elements	Characters
<p>Character collating (Cont.)</p>	<p>[J, j], [K, k], [L, l], [M, m], [N, n, N tilde, n tilde], [O, o, O acute, o acute, O circumflex, o circumflex], [P, p], [Q, q], [R, r], [S, s], [T, t], [U, u, U acute, u acute, U grave, u grave, U circumflex, u circumflex, U diaeresis, u diaeresis], [V, v], [W, w], [X, x], [Y, y], [Z, z], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, >, ?, grave accent, colon, #, @, apostrophe, =, ", left double angle quote, right double angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, micro sign, tilde, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, diaeresis, acute accent, {, }, superscript 0 (zero), superscript 1, superscript 2, superscript 3, superscript 4, superscript 5, superscript 6, superscript 7, superscript 8, superscript 9, \, DEL, left single quotation mark, left double quotation mark, less- than or equal sign, box drawings light horizontal, not equal sign, right single quotation mark, right double quotation mark, greater-than or equal sign, superscript (, box drawings light vertical, superscript), box drawings light up and right, box drawings light vertical and right, box drawings light vertical and left, box drawings light down and left, box drawings light vertical and horizontal, superscript +, box drawings light up and left, box drawings light up and horizontal, box drawings light down and horizontal, box drawings light down and right, superscript -, full block</p>
<p>Substitutions</p>	<p>No concept</p>
<p>Alternate digits</p>	<p>No concept</p>

The following escapement definitions apply to the CanadaGP ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Czechoslovakia Ccsversion Definitions

Table 14–7 lists the definitions for the Czechoslovakia ccsversion. The ccsversion number for Czechoslovakia is 53. The coded character set base is Latin2EBCDIC.

Table 14–7. Czechoslovakia Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, double acute accent, vertical line, comma, %, _, >, ?, caron, grave accent, colon, #, @, apostrophe, =, ", breve, degree sign, cedilla, ogonek, euro sign, tilde, dot above, paragraph sign, diaeresis, acute accent, multiplication sign, {, soft hyphen, }, \, division sign</p>
Lowercase set	<p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, German sharp s, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p>

Table 14–7. Czechoslovakia Ccsversion Definitions

Elements	Characters
Uppercase set	A, A acute, A circumflex, A breve, A diaeresis, A ogonek, B, C, C acute, C cedilla, C caron, D, D caron, D with stroke, E, E acute, E caron, E diaeresis, E ogonek, F, G, H, I, I acute, I circumflex, J, K, L, L acute, L caron, L with stroke, M, N, N acute, N caron, O, O acute, O circumflex, O diaeresis, O double acute, P, Q, R, R acute, R caron, S, S acute, S cedilla, S caron, T, T caron, T cedilla, U, U acute, U with ring above, U diaeresis, U double acute, V, W, X, Y, Y acute, Z, Z with dot above, Z acute, Z caron
Lowercase to uppercase mapping	a-A, a acute-A acute, a circumflex-A circumflex, a breve-A breve, a diaeresis-A diaeresis, a ogonek-A ogonek, b-B, c-C, c acute-C acute, c cedilla-C cedilla, c caron-C caron, d-D, d caron-D caron, d with stroke-D with stroke, e-E, e acute-E acute, e caron-E caron, e diaeresis-E diaeresis, e ogonek-E ogonek, f-F, g-G, h-H, i-I, i acute-I acute, i circumflex-I circumflex, j-J, k-K, l-L, l acute-L acute, l caron-L caron, l with stroke-L with stroke, m-M, n-N, n acute-N acute, n caron-N caron, o-O, o acute-O acute, o circumflex-O circumflex, o diaeresis-O diaeresis, o double acute-O double acute, p-P, q-Q, r-R, r acute-R acute, r caron-R caron, s-S, s acute-S acute, s cedilla-S cedilla, s caron-S caron, t-T, t caron-T caron, t cedilla-T cedilla, u-U, u acute-U acute, u with ring above-U with ring above, u diaeresis-U diaeresis, u double acute-U double acute, v-V, w-W, x-X, y-Y, y acute-Y acute, z-Z, z with dot above-Z with dot above, z acute-Z acute, z caron-Z caron
Uppercase to lowercase mapping	A-a, A acute-a acute, A circumflex-a circumflex, A breve-a breve, A diaeresis-a diaeresis, A ogonek-a ogonek, B-b, C-c, C acute-c acute, C cedilla-c cedilla, C caron-c caron, D-d, D caron-d caron, D with stroke-d with stroke, E-e, E acute-e acute, E caron-e caron, E diaeresis-e diaeresis, E ogonek-e ogonek, F-f, G-g, H-h, I-i, I acute-i acute, I circumflex-i circumflex, J-j, K-k, L-l, L acute-l acute, L caron-l caron, L with stroke-l with stroke, M-m, N-n, N acute-n acute, N caron-n caron, O-o, O acute-o acute, O circumflex-o circumflex, O diaeresis-o diaeresis, O double acute-o double acute, P-p, Q-q, R-r, R acute-r acute, R caron-r caron, S-s, S acute-s acute, S cedilla-s cedilla, S caron-s caron, T-t, T caron-t caron, T cedilla-t cedilla, U-u, U acute-u acute, U with ring above-u with ring above, U diaeresis-u diaeresis, U double acute-u double acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z, Z with dot above-z with dot above, Z acute-z acute, Z caron-z caron

Table 14–7. Czechoslovakia Ccsversion Definitions

Elements	Characters
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>Space, [a, A, a acute, A acute, a circumflex, A circumflex, a breve, A breve, a diaeresis, A diaeresis, a ogonek, A ogonek], [b, B], [c, C, c acute, C acute, c cedilla, C cedilla], [c caron, C caron], [d, D, d caron, D caron, d with stroke, D with stroke], [e, E, e acute, E acute, e caron, E caron, e diaeresis, E diaeresis, e ogonek, E ogonek], [f, F], [g, G], [h, H], [ch, Ch, CH], [i, I, i acute, I acute, i circumflex, I circumflex], [j, J], [k, K], [l, L, l acute, L acute, l caron, L caron, l with stroke, L with stroke], [m, M], [n, N, n acute, N acute, n caron, N caron], [o, O, o acute, O acute, o circumflex, O circumflex, o diaeresis, O diaeresis, o double acute, O double acute], [p, P], [q, Q], [r, R, r acute, R acute], [r caron, R caron], [s, S, s acute, S acute, s cedilla, S cedilla, German sharp s], [s caron, S caron], [t, T, t caron, T caron, t cedilla, T cedilla], [u, U, u acute, U acute, u with ring above, U with ring above, u diaeresis, U diaeresis, u double acute, U double acute], [v, V], [w, W], [x, X], [y, Y, y acute, Y acute], [z, Z, z with dot above, Z with dot above, z acute, Z acute], [z caron, Z caron], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, full stop, comma, semicolon, ?, !, colon, ", apostrophe, grave accent, -, _, vertical line, /, \, (,), left square bracket, , right square bracket, <, >, {, }, ampersand, @, paragraph sign, %, \$, euro sign, dot above, acute accent, caron, breve, diaeresis, double acute accent, cedilla, ogonek, soft hyphen, =, +, ^, multiplication sign, division sign, *, #, tilde</p>
Substitutions	<p>c followed by h is ordered logically as one character, after h.</p> <p>C followed by h is ordered logically as one character, after ch.</p> <p>C followed by H is ordered logically as one character, after Ch.</p>
Alternate digits	No concept

The following escapement definitions apply to the Czechoslovakia ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	double acute accent 0, caron 0, breve 0, degree sign 0, cedilla 0, ogonek 0, dot above 0, diaeresis 0, acute accent 0, other characters left-to-right (+)

France Ccsversion Definitions

Table 14–8 lists the definitions for the France ccsversion. The ccsversion number for France is 35. The coded character set base is Latin5EBCDIC.

Table 14–8. France Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, g breve, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, q, r, s, s cedilla, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, g breve, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, q, r, s, s cedilla, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, z</p> <p>Lowercase i without dot above, German sharp s, y diaeresis</p> <p>Uppercase I with dot above</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ , colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, no-break space, left double-angle quote, right double-angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, soft hyphen, superscript 1, superscript 2, superscript 3, division sign</p>
Lowercase set	<p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, g breve, h, i, i circumflex, i diaeresis, i grave, i acute, i without dot above, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, q, r, s, German sharp s, s cedilla, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y diaeresis, z</p>

Table 14–8. France Ccsversion Definitions

Elements	Characters
Uppercase set	A, A circumflex, A diaeresis, A grave, A acute, A tilde, A ring above, A with E, B, C, C cedilla, D, E, E circumflex, E diaeresis, E grave, E acute, F, G, G breve, H, I, I circumflex, I diaeresis, I grave, I acute, I with dot above, J, K, L, M, N, N tilde, O, O circumflex, O diaeresis, O grave, O acute, O tilde, O with oblique stroke, P, Q, R, S, S cedilla, T, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a grave-A grave, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, e-E, e circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, j-J, k-K, l-L, m-M, o-O, o circumflex-O circumflex, o with oblique stroke-O with oblique stroke, p-P, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, v-V, w-W, x-X, y-Y, y diaeresis-Y, z-Z
Uppercase to lowercase mapping	No concept
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>[No-break space, space], [0, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters], [1, superscript 1], [2, superscript 2], [3, superscript 3], 4, 5, 6, 7, 8, 9, [a, A, feminine ordinal indicator, a acute, A acute, a grave, A grave, a circumflex, A circumflex, a ring above, A ring above, a diaeresis, A diaeresis, a tilde, A tilde], [b, B], [c, C, c cedilla, C cedilla], [d, D], [e, E, e acute, E acute, e grave, E grave, e circumflex, E circumflex, e diaeresis, E diaeresis], [f, F], [g, G, g breve, G breve], [h, H], [i, I, i acute, I acute, i grave, I grave, i circumflex, I circumflex, i diaeresis, I diaeresis, i without dot above, I with dot above], [j, J], [k, K], [l, L], [m, M], [n, N, n tilde, N tilde], [o, O, masculine ordinal indicator, o acute, O acute, o grave, O grave, o circumflex, O circumflex, o tilde, O tilde, o diaeresis, O diaeresis, o with oblique stroke, O with oblique stroke], [p, P], [q, Q], [r, R], [s, S, s cedilla, S cedilla], [t, T], [u, U, u acute, U acute, u grave, U grave, u circumflex, U circumflex, u diaeresis, U diaeresis], [v, V], [w, W], [x, X], [y, Y, y diaeresis], [z, Z], _, macron, soft hyphen, -, comma, semicolon, colon,!, inverted exclamation mark, ?, inverted question mark, /, full stop, acute accent, grave accent, circumflex accent, diaeresis, tilde, middle dot, cedilla, apostrophe, ", left double angle quote, right double angle quote, (,), left square bracket, right square bracket, {, }, paragraph sign, pilcrow sign, copyright sign, registered trademark sign, @, euro sign, cent sign, \$, pound sign, yen sign, *, \, ampersand, #, %, +, plus-minus sign, division sign, multiplication sign, <, =, >, not sign, vertical line, broken bar, degree sign, micro sign</p>

Table 14-8. France Ccsversion Definitions

Elements	Characters
Substitutions	a with e is ordered as a followed by e. A with E is ordered as A followed by E. German sharp s is ordered as s followed by s.
Alternate digits	No concept

The following escapement definitions apply to the France ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	circumflex accent 0, grave accent 0, diaeresis 0, acute accent 0, other characters left-to-right (+)

Hungarian Ccsversion Definitions

Table 14–9 lists the definitions for the Hungarian ccsversion. The ccsversion number for LatinGreek is 85. The coded character set base is Latin2EBCDIC.

Table 14–9. Hungarian Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, double acute accent, vertical line, comma, %, _, >, ?, caron, grave accent, colon, #, @, apostrophe, =, ", breve, degree sign, cedilla, ogonek, euro sign, tilde, dot above, paragraph sign, diaeresis, acute accent, multiplication sign, {, soft hyphen, }, \, division sign</p>
Lowercase set	<p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, German sharp s, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p>

Table 14–9. Hungarian Ccsversion Definitions

Elements	Characters
Uppercase set	A, A acute, A circumflex, A breve, A diaeresis, A ogonek, B, C, C acute, C cedilla, C caron, D, D caron, D with stroke, E, E acute, E caron, E diaeresis, E ogonek, F, G, H, I, I acute, I circumflex, J, K, L, L acute, L caron, L with stroke, M, N, N acute, N caron, O, O acute, O circumflex, O diaeresis, O double acute, P, Q, R, R acute, R caron, S, S acute, S cedilla, S caron, T, T caron, T cedilla, U, U acute, U with ring above, U diaeresis, U double acute, V, W, X, Y, Y acute, Z, Z with dot above, Z acute, Z caron
Lowercase to uppercase mapping	a-A, a acute-A acute, a circumflex-A circumflex, a breve-A breve, a diaeresis-A diaeresis, a ogonek-A ogonek, b-B, c-C, c acute-C acute, c cedilla-C cedilla, c caron-C caron, d-D, d caron-D caron, d with stroke-D with stroke, e-E, e acute-E acute, e caron-E caron, e diaeresis-E diaeresis, e ogonek-E ogonek, f-F, g-G, h-H, i-I, i acute-I acute, i circumflex-I circumflex, j-J, k-K, l-L, l acute-L acute, l caron-L caron, l with stroke-L with stroke, m-M, n-N, n acute-N acute, n caron-N caron, o-O, o acute-O acute, o circumflex-O circumflex, o diaeresis-O diaeresis, o double acute-O double acute, p-P, q-Q, r-R, r acute-R acute, r caron-R caron, s-S, s acute-S acute, s cedilla-S cedilla, s caron-S caron, t-T, t caron-T caron, t cedilla-T cedilla, u-U, u acute-U acute, u with ring above-U with ring above, u diaeresis-U diaeresis, u double acute-U double acute, v-V, w-W, x-X, y-Y, y acute-Y acute, z-Z, z with dot above-Z with dot above, z acute-Z acute, z caron-Z caron
Uppercase to lowercase mapping	A-a, A acute-a acute, A circumflex-a circumflex, A breve-a breve, A diaeresis-a diaeresis, A ogonek-a ogonek, B-b, C-c, C acute-c acute, C cedilla-c cedilla, C caron-c caron, D-d, D caron-d caron, D with stroke-d with stroke, E-e, E acute-e acute, E caron-e caron, E diaeresis-e diaeresis, E ogonek-e ogonek, F-f, G-g, H-h, I-i, I acute-i acute, I circumflex-i circumflex, J-j, K-k, L-l, L acute-l acute, L caron-l caron, L with stroke-l with stroke, M-m, N-n, N acute-n acute, N caron-n caron, O-o, O acute-o acute, O circumflex-o circumflex, O diaeresis-o diaeresis, O double acute-o double acute, P-p, Q-q, R-r, R acute-r acute, R caron-r caron, S-s, S acute-s acute, S cedilla-s cedilla, S caron-s caron, T-t, T caron-t caron, T cedilla-t cedilla, U-u, U acute-u acute, U with ring above-u with ring above, U diaeresis-u diaeresis, U double acute-u double acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z, Z with dot above-z with dot above, Z acute-z acute, Z caron-z caron

Table 14–9. Hungarian Ccsversion Definitions

Elements	Characters
Character collating	<p>Note: in the character collating sequence, the series of characters within square brackets are given the same rank in equivalent ordering.</p> <p>[Space, No-break space], [a, A, a acute, A acute, a circumflex, A circumflex, a breve, A breve, a diaeresis, A diaeresis, a ogonek, A ogonek], [b, B],</p> <p>[c, C, c acute, C acute, c cedilla, C cedilla, c caron, C caron], [d, D, d caron, D caron, d with stroke, D with stroke], [e, E, e acute, E acute, e caron, E caron, e diaeresis, E diaeresis, e ogonek, E ogonek], [f, F], [g, G], [h, H], [i, I, i acute, I acute, i circumflex, I circumflex],</p> <p>[j, J], [k, K], [l, L, l acute, L acute, l caron, L caron, l with stroke, L with stroke], [m, M], [n, N, n acute, N acute, n caron, N caron], [o, O, o acute, O acute, o circumflex, O circumflex], [o diaeresis, O diaeresis, o double acute, O double acute], [p, P], [q, Q],</p> <p>[r, R, r acute, R acute, r caron, R caron], [s, S, s acute, S acute, s cedilla, S cedilla, German sharp s, s caron, S caron],</p> <p>[t, T, t caron, T caron, t cedilla, T cedilla], [u, U, u acute, U acute, u with ring above, U with ring above], [u diaeresis, U diaeresis, u double acute, U double acute], [v, V], [w, W], [x, X],</p> <p>[y, Y, y acute, Y acute], [z, Z, z with dot above, Z with dot above, z acute, Z acute, z caron, Z caron],</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, full stop, comma, semicolon, ?, !, colon, ", apostrophe, grave accent, -, _, vertical line, /, \, (,), left square bracket, right square bracket, <, >, {, }, ampersand, @, paragraph sign, %, \$, euro sign, dot above, acute accent, caron, breve, diaeresis, double acute accent, cedilla, ogonek, soft hyphen, =, +, ^, multiplication sign, division sign, *, #, tilde</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Hungarian ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	double acute accent 0, caron 0, breve 0, degree sign 0, cedilla 0, ogonek 0, dot above 0, diaeresis 0, acute accent 0, other characters left-to-right (+)

LatinGreek Ccsversion Definitions

Table 14–10 lists the definitions for the LatinGreek ccsversion. The ccsversion number for LatinGreek is 49. The coded character set base is LatinGreekEBCDIC.

Table 14–10. LatinGreek Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z</p> <p>Lowercase and uppercase forms of the following Greek letters: Alpha, alpha with accent, beta, gamma, delta, epsilon, epsilon with accent, zeta, eta, eta with accent, theta, iota, iota with accent, iota with diaeresis, kappa, lamda, mu, nu, xi, omicron, omicron with accent, pi, rho, sigma, tau, upsilon, upsilon with accent, upsilon with diaeresis, phi, chi, psi, omega, omega with accent</p> <p>Lowercase terminal sigma, upsilon with diaeresis and accent</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z</p> <p>Lowercase and uppercase forms of the following Greek letters: alpha, alpha with accent, beta, gamma, delta, epsilon, epsilon with accent, zeta, eta, eta with accent, theta, iota, iota with accent, iota with diaeresis, kappa, lamda, mu, nu, xi, omicron, omicron with accent, pi, rho, sigma, tau, upsilon, upsilon with accent, upsilon with diaeresis, phi, chi, psi, omega, omega with accent</p> <p>Lowercase iota with diaeresis and accent, terminal sigma, upsilon with diaeresis and accent</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, >, ?, grave accent, colon, #, @, apostrophe, =, ", tilde, {, }, \, left single quotation mark, right single quotation mark, pound sign, broken bar, paragraph sign, diaeresis, copyright sign, left double-angle quote, not sign, macron, degree sign, plus-minus sign, superscript 2, superscript 3, Greek accent, diaeresis and Greek accent, middle dot, right double-angle quote, vulgar fraction one half</p>

Table 14–10. LatinGreek Ccsversion Definitions

Elements	Characters
Lowercase set	<p>a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z</p> <p>Lowercase forms of the following Greek letters: alpha, alpha with accent, beta, gamma, delta, epsilon, epsilon with accent, zeta, eta, eta with accent, theta, iota, iota with accent, iota with diaeresis, iota with diaeresis and accent, kappa, lamda, mu, nu, xi, omicron, omicron with accent, pi, rho, terminal sigma, sigma, tau, upsilon, upsilon with accent, upsilon with diaeresis, upsilon with diaeresis and accent, phi, chi, psi, omega, omega with accent</p>
Uppercase set	<p>A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z</p> <p>Uppercase forms of the following Greek letters: alpha, alpha with accent, beta, gamma, delta, epsilon, epsilon with accent, zeta, eta, eta with accent, theta, iota, iota with accent, iota with diaeresis, kappa, lamda, mu, nu, xi, omicron, omicron with accent, pi, rho, sigma, tau, upsilon, upsilon with accent, upsilon with diaeresis, phi, chi, psi, omega, omega with accent</p>

Table 14–10. LatinGreek Ccsversion Definitions

Elements	Characters
Lowercase to uppercase mapping	<p>a-A, b-B, c-C, d-D, e-E, f-F, g-G, h-H, i-I, j-J, k-K, l-L, m-M, n-N, o-O, p-P, q-Q, r-R, s-S, t-T, u-U, v-V, w-W, x-X, y-Y, z-Z</p> <p>For the following Greek letters: lowercase alpha-uppercase alpha lowercase alpha with accent- uppercase alpha uppercase alpha with accent-uppercase alpha lowercase beta-uppercase beta, lowercase gamma- uppercase gamma lowercase delta-uppercase delta, lowercase epsilon-uppercase epsilon lowercase epsilon with accent-uppercase epsilon uppercase epsilon with accent-uppercase epsilon lowercase zeta- uppercase zeta, lowercase eta-uppercase eta lowercase eta with accent-uppercase eta uppercase eta with accent-uppercase eta lowercase theta- uppercase theta, lowercase iota-uppercase iota lowercase iota with accent-uppercase iota lowercase iota with diaeresis-uppercase iota uppercase iota with diaeresis-uppercase iota lowercase iota with diaeresis and accent-uppercase iota lowercase kappa-uppercase kappa, lowercase lamda- uppercase lamda lowercase mu-uppercase mu lowercase nu- uppercase nu lowercase xi-uppercase xi, lowercase omicron- uppercase omicron lowercase omicron with accent- uppercase omicron uppercase omicron with accent- uppercase omicron lowercase pi-uppercase pi, lowercase rho- uppercase rho lowercase terminal sigma-uppercase sigma lowercase sigma-uppercase sigma, lowercase tau-uppercase tau lowercase upsilon-uppercase upsilon lowercase upsilon with accent-uppercase upsilon uppercase upsilon with accent-uppercase upsilon lowercase upsilon with diaeresis-uppercase upsilon uppercase upsilon with diaeresis-uppercase upsilon lowercase upsilon with diaeresis and accent-uppercase upsilon lowercase phi-uppercase phi, lowercase chi-uppercase chi lowercase psi-uppercase psi, lowercase omega-uppercase omega lowercase omega with accent-uppercase omega uppercase omega with accent-uppercase omega</p>
Uppercase to lowercase mapping	No concept

Table 14–10. LatinGreek Ccsversion Definitions

Elements	Characters
Character collating	<p>NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SUB, ESC, IS4, IS3, IS2, IS1, space, !, ", #, \$, %, ampersand, apostrophe, (,), *, +, comma, -, full stop, /, colon, semicolon, <, =, >, ?, @, left square bracket, \, right square bracket, circumflex accent, _, grave accent, {, vertical line, }, tilde, DEL, BPH, NBH, IND, NEL, SSA, ESA, HTS, HTJ, VTS, PLD, PLU, RI, SS2, SS3, DCS, PU1, PU2, STS, CCH, MW, SPA, EPA, SOS, SCI, CSI, ST, OSC, PM, APC, no-break space, left single quotation mark, right single quotation mark, pound sign, broken bar, paragraph sign, diaeresis, copyright sign, left double-angle quote, not sign, soft hyphen, macron, degree sign, plus-minus sign, superscript 2, superscript 3, Greek accent, diaeresis and Greek accent, middle dot, right double-angle quote, vulgar fraction one half, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, a, B, b, C, c, D, d, E, e, F, f, G, g, H, h, I, i, K, k, L, l, M, m, N, n, O, o, P, p, Q, q, R, r, S, s, T, t, U, u, V, v, W, w, X, x, Y, y, Z, z, uppercase alpha, uppercase alpha with accent, lowercase alpha, lowercase alpha with accent, uppercase beta, lowercase beta, uppercase gamma, lowercase gamma, uppercase delta, lowercase delta, uppercase epsilon, uppercase epsilon with accent, lowercase epsilon, lowercase epsilon with accent, uppercase zeta, lowercase zeta, uppercase eta, uppercase eta with accent, lowercase eta, lowercase eta with accent, uppercase theta, lowercase theta, uppercase iota, uppercase iota with accent, uppercase iota with diaeresis, lowercase iota, lowercase iota with accent, lowercase iota with diaeresis, lowercase iota with diaeresis and accent, uppercase kappa, lowercase kappa, uppercase lamda, lowercase lamda, uppercase mu, lowercase mu, uppercase nu, lowercase nu, uppercase xi, lowercase xi, uppercase omicron, uppercase omicron with accent, lowercase omicron, lowercase omicron with accent, uppercase pi, lowercase pi, uppercase rho, lowercase rho, uppercase sigma, lowercase terminal sigma, lowercase sigma, uppercase tau, lowercase tau, uppercase upsilon, uppercase upsilon with accent, uppercase upsilon with diaeresis, lowercase upsilon, lowercase upsilon with accent, lowercase upsilon with diaeresis, lowercase upsilon with diaeresis and accent, uppercase phi, lowercase phi, uppercase chi, lowercase chi, uppercase psi, lowercase psi, uppercase omega, uppercase omega with accent, lowercase omega, lowercase omega with accent</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the LatinGreek ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Norway Ccsversion Definitions

Table 14–11 lists the definitions for the Norway ccsversion. The ccsversion number for LatinGreek is 71. The coded character set base is Latin1EBCDIC.

Table 14–11. Norway Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, left double-angle quote, right double-angle quote, degree sign, feminine ordinal indicator, masculine ordinal indicator, euro sign, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, broken bar, diaeresis, acute accent, multiplication sign, division sign</p>
Lowercase set	<p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, lowercase Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, lowercase Icelandic thorn, q, r, s, German sharp s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y diaeresis, y acute, z</p>

Table 14-11. Norway Ccsversion Definitions

Elements	Characters
Uppercase set	A, A circumflex, A diaeresis, A grave, A acute, A tilde, A ring above, A with E, B, C, C cedilla, D, uppercase Icelandic eth, E, E circumflex, E diaeresis, E grave, E acute, F, G, H, I, I circumflex, I diaeresis, I grave, I acute, J, K, L, M, N, N tilde, O, O circumflex, O diaeresis, O grave, O acute, O tilde, O with oblique stroke, P, uppercase Icelandic thorn, Q, R, S, T, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Y acute, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a diaeresis-A diaeresis, a grave-A grave, a acute-A acute, a tilde-A tilde, a ring above-A ring above, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, lowercase Icelandic eth-uppercase Icelandic eth, e-E, e circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, i grave-I grave, i acute-I acute, j-J, k-K, l-L, m-M, n-N, n tilde-N tilde, o-O, o circumflex-O circumflex, o diaeresis-O diaeresis, o grave-O grave, o acute-O acute, o tilde-O tilde, o with oblique stroke-O with oblique stroke, p-P, lowercase Icelandic thorn-uppercase Icelandic thorn, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, u acute-U acute, v-V, w-W, x-X, y-Y, y acute-Y acute, z-Z
Uppercase to lowercase mapping	No concept

Table 14-11. Norway Ccsversion Definitions

Elements	Characters
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>[NUL, SOH, STX, ETX, EOT, ENQ, ACK, BEL, BS, HT, LF, VT, FF, CR, SO], [SI, DLE, DC1, DC2, DC3, DC4, NAK, SYN, ETB, CAN, EM, SUB, ESC, IS4, IS3], [IS2, IS1, APC, BPH, NBH, IND, NEL, SSA, ESA, HTS, HTJ, VTS, PLD], [PLU, RI, SS2, SS3, DCS, PU1, PU2, STS, CCH, MW, SPA, EPA, SOS, SCI], [CSI, ST, OSC, PM, DEL], [space, no-break space], [!, inverted exclamation mark, ", diaeresis, #, \$, cent sign, pound sign, euro sign, yen sign, paragraph sign, %, ampersand], [apostrophe, acute accent, (,), *, +, multiplication sign, division sign, plus-minus sign, comma], [cedilla, soft hyphen, not sign, <, left double angle quote, =, >, right double angle quote, ?, inverted question mark, @], [copyright sign, registered trademark sign, feminine ordinal indicator, masculine ordinal indicator], degree sign, left square bracket, \, right square bracket, circumflex accent, _, grave accent], [l, vertical line, broken bar, }, tilde], [macron, micro sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters], middle dot, -, full stop, /, 0, [1, superscript 1], [2, superscript 2], [3, superscript 3], 4, 5, 6, 7, 8, 9, [A, A grave, A acute, A circumflex, A tilde, a, a grave, a acute, a circumflex, a tilde], [B, b], [C, C cedilla, c, c cedilla], [D, uppercase Icelandic eth, d, lowercase Icelandic eth], [E, E grave, E acute, E circumflex, E diaeresis, e, e grave, e acute, e circumflex, e diaeresis], [F, f], [G, g], [H, h], [I, I grave, I acute, I circumflex, I diaeresis, i, i grave, i acute, i circumflex, i diaeresis], [J, j], [K, k], [L, l], [M, m], [N, N tilde, n, n tilde], [O, O grave, O acute, O circumflex, O tilde, o, o grave, o acute, o circumflex, o tilde], [P, p], [Q, q] [R, r], [S, s, German sharp s], [T, uppercase Icelandic thorn, t, lowercase Icelandic thorn], [U, U grave, U acute, U circumflex, u, u grave, u acute, u circumflex], [V, v], [W, w], [X, x], [Y, Y acute, U diaeresis, y, y acute, u diaeresis, y diaeresis], [Z, z], [A with E, A diaeresis, a with e, a diaeresis], [O with oblique stroke, O diaeresis, o with oblique stroke, o diaeresis], [A ring above, a ring above]</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Norway ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Polish Ccsversion Definitions

Table 14–12 lists the definitions for the Polish ccsversion. The ccsversion number for Polish is 86. The coded character set base is Latin2EBCDIC.

Table 14–12. Polish Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, double acute accent, vertical line, comma, %, _ , >, ?, caron, grave accent, colon, #, @, apostrophe, =, ", breve, degree sign, cedilla, ogonek, euro sign, tilde, dot above, paragraph sign, diaeresis, acute accent, multiplication sign, {, soft hyphen, }, \, division sign</p>
Lowercase set	<p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, German sharp s, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p>

Table 14–12. Polish Ccsversion Definitions

Elements	Characters
Uppercase set	A, A acute, A circumflex, A breve, A diaeresis, A ogonek, B, C, C acute, C cedilla, C caron, D, D caron, D with stroke, E, E acute, E caron, E diaeresis, E ogonek, F, G, H, I, I acute, I circumflex, J, K, L, L acute, L caron, L with stroke, M, N, N acute, N caron, O, O acute, O circumflex, O diaeresis, O double acute, P, Q, R, R acute, R caron, S, S acute, S cedilla, S caron, T, T caron, T cedilla, U, U acute, U with ring above, U diaeresis, U double acute, V, W, X, Y, Y acute, Z, Z with dot above, Z acute, Z caron
Lowercase to uppercase mapping	a-A, a acute-A acute, a circumflex-A circumflex, a breve-A breve, a diaeresis-A diaeresis, a ogonek-A ogonek, b-B, c-C, c acute-C acute, c cedilla-C cedilla, c caron-C caron, d-D, d caron-D caron, d with stroke-D with stroke, e-E, e acute-E acute, e caron-E caron, e diaeresis-E diaeresis, e ogonek-E ogonek, f-F, g-G, h-H, i-I, i acute-I acute, i circumflex-I circumflex, j-J, k-K, l-L, l acute-L acute, l caron-L caron, l with stroke-L with stroke, m-M, n-N, n acute-N acute, n caron-N caron, o-O, o acute-O acute, o circumflex-O circumflex, o diaeresis-O diaeresis, o double acute-O double acute, p-P, q-Q, r-R, r acute-R acute, r caron-R caron, s-S, s acute-S acute, s cedilla-S cedilla, s caron-S caron, t-T, t caron-T caron, t cedilla-T cedilla, u-U, u acute-U acute, u with ring above-U with ring above, u diaeresis-U diaeresis, u double acute-U double acute, v-V, w-W, x-X, y-Y, y acute-Y acute, z-Z, z with dot above-Z with dot above, z acute-Z acute, z caron-Z caron
Uppercase to lowercase mapping	A-a, A acute-a acute, A circumflex-a circumflex, A breve-a breve, A diaeresis-a diaeresis, A ogonek-a ogonek, B-b, C-c, C acute-c acute, C cedilla-c cedilla, C caron-c caron, D-d, D caron-d caron, D with stroke-d with stroke, E-e, E acute-e acute, E caron-e caron, E diaeresis-e diaeresis, E ogonek-e ogonek, F-f, G-g, H-h, I-i, I acute-i acute, I circumflex-i circumflex, J-j, K-k, L-l, L acute-l acute, L caron-l caron, L with stroke-l with stroke, M-m, N-n, N acute-n acute, N caron-n caron, O-o, O acute-o acute, O circumflex-o circumflex, O diaeresis-o diaeresis, O double acute-o double acute, P-p, Q-q, R-r, R acute-r acute, R caron-r caron, S-s, S acute-s acute, S cedilla-s cedilla, S caron-s caron, T-t, T caron-t caron, T cedilla-t cedilla, U-u, U acute-u acute, U with ring above-u with ring above, U diaeresis-u diaeresis, U double acute-u double acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z, Z with dot above-z with dot above, Z acute-z acute, Z caron-z caron

Table 14–12. Polish Ccsversion Definitions

Elements	Characters
Character collating	<p>Note: in the character collating sequence, the series of characters within square brackets are given the same rank in equivalent ordering.</p> <p>[Space, No-break space], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [A, a], [A ogonek, a ogonek], [A acute, a acute], [A circumflex, a circumflex], [A breve, a breve], [A diaeresis, a diaeresis], [B, b], [C, c], [C acute, c acute], [C cedilla, c cedilla], [C caron, c caron], [D, d], [D caron, d caron], [D with stroke, d with stroke], [E, e], [E acute, e acute], [E ogonek, e ogonek], [E diaeresis, e diaeresis], [E caron, e caron], [F, f], [G, g], [H, h], [I, i], [I acute, i acute], [I circumflex, i circumflex], [J, j], [K, k], [L, l], [L with stroke, l with stroke], [L caron, l caron], [L acute, l acute], [M, m], [N, n], [N acute, n acute], [N caron, n caron], [O, o], [O acute, o acute], [O circumflex, o circumflex], [O double acute, o double acute], [O diaeresis, o diaeresis], [P, p], [Q, q], [R, r], [R acute, r acute], [R caron, r caron], [S, s], [S acute, s acute], [S caron, s caron], [S cedilla, s cedilla, German sharp s], [T, t], [T caron, t caron], [T cedilla, t cedilla], [U, u], [U with ring above, u with ring above], [U acute, u acute], [U double acute, u double acute], [U diaeresis, u diaeresis], [V, v], [W, w], [X, x], [Y, y], [Y acute, y acute], [Z, z], [Z acute, z acute], [Z caron, z caron], [Z with dot above, z with dot above], full stop, comma, semicolon, ?, !, colon, ", apostrophe, grave accent, -, _ , vertical line, /, \, (,), left square bracket, right square bracket, <, >, {, }, ampersand, @, paragraph sign, %, \$, euro sign, dot above, acute accent, caron, breve, diaeresis, double acute accent, cedilla, ogonek, soft hyphen, =, +, ^, multiplication sign, division sign, *, #, tilde</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Polish ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	double acute accent 0, caron 0, breve 0, degree sign 0, cedilla 0, ogonek 0, dot above 0, diaeresis 0, acute accent 0, other characters left-to-right (+)

Romanian Ccsversion Definitions

Table 14–13 lists the definitions for the Romanian ccsversion. The ccsversion number for Romanian is 96. The coded character set base is Latin2EBCDIC.

Table 14–13. Romanian Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p> <p>Lowercase German sharp s</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, double acute accent, vertical line, comma, %, _, >, ?, caron, grave accent, colon, #, @, apostrophe, =, ", breve, degree sign, cedilla, ogonek, euro sign, tilde, dot above, paragraph sign, diaeresis, acute accent, multiplication sign, {, soft hyphen, }, \, division sign</p>
Lowercase set	<p>a, a acute, a circumflex, a breve, a diaeresis, a ogonek, b, c, c acute, c cedilla, c caron, d, d caron, d with stroke, e, e acute, e caron, e diaeresis, e ogonek, f, g, h, i, i acute, i circumflex, j, k, l, l acute, l caron, l with stroke, m, n, n acute, n caron, o, o acute, o circumflex, o diaeresis, o double acute, p, q, r, r acute, r caron, s, s acute, s cedilla, German sharp s, s caron, t, t caron, t cedilla, u, u acute, u with ring above, u diaeresis, u double acute, v, w, x, y, y acute, z, z with dot above, z acute, z caron</p>

Table 14-13. Romanian Ccsversion Definitions

Elements	Characters
Uppercase set	A, A acute, A circumflex, A breve, A diaeresis, A ogonek, B, C, C acute, C cedilla, C caron, D, D caron, D with stroke, E, E acute, E caron, E diaeresis, E ogonek, F, G, H, I, I acute, I circumflex, J, K, L, L acute, L caron, L with stroke, M, N, N acute, N caron, O, O acute, O circumflex, O diaeresis, O double acute, P, Q, R, R acute, R caron, S, S acute, S cedilla, S caron, T, T caron, T cedilla, U, U acute, U with ring above, U diaeresis, U double acute, V, W, X, Y, Y acute, Z, Z with dot above, Z acute, Z caron
Lowercase to uppercase mapping	a-A, a acute-A acute, a circumflex-A circumflex, a breve-A breve, a diaeresis-A diaeresis, a ogonek-A ogonek, b-B, c-C, c acute-C acute, c cedilla-C cedilla, c caron-C caron, d-D, d caron-D caron, d with stroke-D with stroke, e-E, e acute-E acute, e caron-E caron, e diaeresis-E diaeresis, e ogonek-E ogonek, f-F, g-G, h-H, i-I, i acute-I acute, i circumflex-I circumflex, j-J, k-K, l-L, l acute-L acute, l caron-L caron, l with stroke-L with stroke, m-M, n-N, n acute-N acute, n caron-N caron, o-O, o acute-O acute, o circumflex-O circumflex, o diaeresis-O diaeresis, o double acute-O double acute, p-P, q-Q, r-R, r acute-R acute, r caron-R caron, s-S, s acute-S acute, s cedilla-S cedilla, s caron-S caron, t-T, t caron-T caron, t cedilla-T cedilla, u-U, u acute-U acute, u with ring above-U with ring above, u diaeresis-U diaeresis, u double acute-U double acute, v-V, w-W, x-X, y-Y, y acute-Y acute, z-Z, z with dot above-Z with dot above, z acute-Z acute, z caron-Z caron
Uppercase to lowercase mapping	A-a, A acute-a acute, A circumflex-a circumflex, A breve-a breve, A diaeresis-a diaeresis, A ogonek-a ogonek, B-b, C-c, C acute-c acute, C cedilla-c cedilla, C caron-c caron, D-d, D caron-d caron, D with stroke-d with stroke, E-e, E acute-e acute, E caron-e caron, E diaeresis-e diaeresis, E ogonek-e ogonek, F-f, G-g, H-h, I-i, I acute-i acute, I circumflex-i circumflex, J-j, K-k, L-l, L acute-l acute, L caron-l caron, L with stroke-l with stroke, M-m, N-n, N acute-n acute, N caron-n caron, O-o, O acute-o acute, O circumflex-o circumflex, O diaeresis-o diaeresis, O double acute-o double acute, P-p, Q-q, R-r, R acute-r acute, R caron-r caron, S-s, S acute-s acute, S cedilla-s cedilla, S caron-s caron, T-t, T caron-t caron, T cedilla-t cedilla, U-u, U acute-u acute, U with ring above-u with ring above, U diaeresis-u diaeresis, U double acute-u double acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z, Z with dot above-z with dot above, Z acute-z acute, Z caron-z caron

Table 14-13. Romanian Ccsversion Definitions

Elements	Characters
Character collating	<p>Note: in the character collating sequence, the series of characters within square brackets are given the same rank in equivalent ordering.</p> <p>[Space, no-break space], [a, A, a acute, A acute, a breve, A breve, a circumflex, A circumflex, a diaeresis, A diaeresis, a ogonek, A ogonek], [B, b], [c, C, c acute, C acute, c cedilla, C cedilla], [c caron, C caron], [d, D, d caron, D caron, d with stroke, D with stroke], [e, E, e acute, E acute, e caron, E caron, e diaeresis, E diaeresis, e ogonek, E ogonek] [f, F], [g, G], [h, H], [i, I, i acute, I acute, i circumflex, I circumflex], [j, J], [k, K], [l, L], [l acute, L acute, l caron, L caron, l with stroke, L with stroke], [m, M], [n, N, n acute, N acute, n caron, N caron], [o, O, o acute, O acute, o circumflex, O circumflex, o diaeresis, O diaeresis, o double acute, O double acute], [p, P], [q, Q], [r, R, r acute, R acute], [r caron, R caron], [s, S, s acute, S acute, s cedilla, S cedilla, German sharp s], [t, T, t caron, T caron, t cedilla, T cedilla], [u, U, u acute, U acute, u with ring above, U with ring above, u diaeresis, U diaeresis, u double acute, U double acute], [v, V], [w, W], [x, X], [y, Y, y acute, Y acute], [z, Z, z with dot above, Z with dot above, z acute, Z acute], [z caron, Z caron], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, full stop, comma, semicolon, ?, !, colon, ", apostrophe, grave accent, -, _, vertical line, /, \, (,), left square bracket, right square bracket, <, >, {, }, ampersand, @, paragraph sign, %, \$, euro sign, dot above, acute accent, caron, breve, diaeresis, double acute accent, cedilla, ogonek, =, +, ^, multiplication sign, division sign, *, #, tilde</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Romanian ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	double acute accent 0, caron 0, breve 0, degree sign 0, cedilla 0, ogonek 0, dot above 0, diaeresis 0, acute accent 0, other characters left-to-right (+)

Russian Ccsversion Definitions

Table 14–14 lists the definitions for the Russian ccsversion. The ccsversion number for Russian is 88. The coded character set base is LatinCyrillicEBC.

Table 14–14. Russian Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	Lowercase and uppercase forms of the following Latin letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, Lowercase and uppercase forms of the following Cyrillic letters: a, be, ve, ghe, de, ie, zhe, ze, i, short i, ka, el, em, en, o, pe, er, es, te, u, ef, ha, tse, che, io, sha, shcha, hardsign, yeru, softsign, e, yu, ya
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	Lowercase and uppercase forms of the following Latin letters: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, Lowercase and uppercase forms of the following Cyrillic letters: a, be, ve, ghe, de, ie, zhe, ze, i, short i, ka, el, em, en, o, pe, er, es, te, u, ef, ha, tse, che, io, sha, shcha, hardsign, yeru, softsign, e, yu, ya, Serbo-Croatian dje, Macedonian gje, Ukrainian ie, Macedonian dze, Byelorussian-Ukrainian i, Ukrainian yi, je, lje, nje, Serbo-Croatian tshe, Macedonian kje, Byelorussian short u, dzhe 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ , >, ?, grave accent, colon, #, @, apostrophe, =, ", tilde, {, }, \, number acronym, soft hyphen, paragraph sign
Lowercase set	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, Lowercase forms of the following Cyrillic letters: a, be, ve, ghe, de, ie, zhe, ze, i, short i, ka, el, em, en, o, pe, er, es, te, u, ef, ha, tse, che, io, sha, shcha, hardsign, yeru, softsign, e, yu, ya
Uppercase set	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, Uppercase forms of the following Cyrillic letters: a, be, ve, ghe, de, ie, zhe, ze, i, short i, ka, el, em, en, o, pe, er, es, te, u, ef, ha, tse, che, io, sha, shcha, hardsign, yeru, softsign, e, yu, ya

Table 14-14. Russian Ccsversion Definitions

Elements	Characters
<p>Lowercase to uppercase mapping</p>	<p>a-A, b-B, c-C, d-D, e-E, f-F, g-G, h-H, i-I, j-J, k-K, l-L, m-M, n-N, o-O, p-P, q-Q, r-R, s-S, t-T, u-U, v-V, w-W, x-X, y-Y, z-Z,</p> <p>For the following Cyrillic letters:</p> <p>lowercase a-uppercase a, lowercase be-uppercase be, lowercase ve-uppercase ve, lowercase ghe-uppercase ghe, lowercase de-uppercase de, lowercase ie-uppercase ie, lowercase zhe-uppercase zhe, lowercase ze-uppercase ze, lowercase i-uppercase I, lowercase short i-uppercase short i, lowercase ka-uppercase ka, lowercase el-uppercase el, lowercase em-uppercase em, lowercase en-uppercase en, lowercase o-uppercase o, lowercase pe-uppercase pe lowercase er-uppercase er, lowercase es-uppercase es lowercase te-uppercase te, lowercase u-uppercase u, lowercase ef-uppercase ef, lowercase ha-uppercase ha, lowercase tse-uppercase tse, lowercase che-uppercase che, lowercase io-uppercase io, lowercase sha-uppercase sha, lowercase shcha-uppercase shcha, lowercase hardsign-uppercase hardsign, lowercase yeru-uppercase yeru, lowercase softsign-uppercase softsign, lowercase e-uppercase e, lowercase yu-uppercase yu, lowercase ya-uppercase ya</p>
<p>Uppercase to lowercase mapping</p>	<p>A-a, B-b, C-c, D-d, E-e, F-f, G-g, H-h, I-i, J-j, K-k, L-l, M-m, N-n, O-o, P-p, Q-q, R-r, S-s, T-t, U-u, V-v, W-w, X-x, Y-y, Z-z,</p> <p>For the following Cyrillic letters:</p> <p>uppercase a-lowercase a, uppercase be-lowercase be, uppercase ve-lowercase ve, uppercase ghe-lowercase ghe, uppercase de-lowercase de, uppercase ie-lowercase ie, uppercase zhe-lowercase zhe, uppercase ze-lowercase ze, uppercase i-lowercase I, uppercase short i-lowercase short i, uppercase ka-lowercase ka, uppercase el-lowercase el, uppercase em-lowercase em, uppercase en-lowercase en, uppercase o-lowercase o, uppercase pe-lowercase pe uppercase er-lowercase er, uppercase es-lowercase es uppercase te-lowercase te, uppercase u-lowercase u, uppercase ef-lowercase ef, uppercase ha-lowercase ha, uppercase tse-lowercase tse, uppercase che-lowercase che, uppercase io-lowercase io, uppercase sha-lowercase sha, uppercase shcha-lowercase shcha, uppercase hardsign-lowercase hardsign, uppercase yeru-lowercase yeru, uppercase softsign-lowercase softsign, uppercase e-lowercase e, uppercase yu-lowercase yu, uppercase ya-lowercase ya</p>

Table 14-14. Russian Ccsversion Definitions

Elements	Characters
Character collating	<p>Note: in the character collating sequence, the series of characters within square brackets are given the same rank in equivalent ordering.</p> <p>Space, [A, a], [B, b], [C, c], [D, d], [E, e] [F, f], [G, g], [H, h], [I, i], [J, j], [K, k], [L, l], [M, m], [N, n], [O, o], [P, p], [Q, q] [R, r], [S, s], [T, t], [U, u], [V, v], [W, w] [X, x], [Y, y], [Z, z],</p> <p>[Forms of the following Cyrillic letters:</p> <p>[uppercase a, lowercase a], [uppercase be, lowercase be], [uppercase ve, lowercase ve], [uppercase ghe, lowercase ghe], [uppercase de, lowercase de], [uppercase ie, lowercase ie], [uppercase io, lowercase io], [uppercase zhe, lowercase zhe], [uppercase ze, lowercase ze], [uppercase i, lowercase i], [uppercase short i, lowercase short i],[uppercase ka, lowercase ka], [uppercase el, lowercase el], [uppercase em, lowercase em], [uppercase en, lowercase en], [uppercase o, lowercase o], [uppercase pe, lowercase pe], [uppercase er, lowercase er], [uppercase es, lowercase es], [uppercase te, lowercase te], [uppercase u, lowercase u], [uppercase ef, lowercase ef], [uppercase ha, lowercase ha] [uppercase tse, lowercase tse], [uppercase che, lowercase che], [uppercase sha, lowercase sha],[uppercase shcha, lowercase shcha], [uppercase hardsign, lowercase hardsign], [uppercase yeru, lowercase yeru], [uppercase softsign, lowercase softsign], [uppercase e, lowercase e], [uppercase yu, lowercase yu], [uppercase ya, lowercase ya],</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, no-break space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _ , >, ?, grave accent, colon, #, @, apostrophe, =, ", tilde, {, },\, number acronym, soft hyphen, paragraph sign</p> <p>Forms of the following Cyrillic letters:</p> <p>[lowercase Serbo-Croatian dje, lowercase Macedonian gje, lowercase Ukrainian ie, lowercase Macedonian dze, uppercase Serbo-Croatian dje, lowercase Byelorussian-Ukrainian i, lowercase Ukrainian yi, lowercase je, lowercase lje, lowercase nje, lowercase Serbo-Croatian tshe, lowercase Macedonian kje, lowercase Byelorussian short u, lowercase dzhe, uppercase Byelorussian-Ukrainian i]</p> <p>[uppercase Ukrainian yi, uppercase je, uppercase lje, uppercase nje, uppercase Serbo-Croatian tshe, uppercase Macedonian kje, uppercase Byelorussian short u, uppercase dzhe, uppercase Macedonian gje, uppercase Ukrainian ie, uppercase Macedonian dze, DEL]</p>
Substitutions	No concept
Alternate digits	No concept

Russian Ccsversion Definitions

The following escapement definitions apply to the Russian ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Spanish Ccsversion Definitions

Table 14–15 lists the definitions for the Spanish ccsversion. The ccsversion number for Spanish is 98. The coded character set base is Latin1EBCDIC.

Table 14–15. Spanish Ccsversion Definitions

Elements	Characters
Spaces data class	Space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>Space</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, no-break space, left double-angle quote, right double-angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, euro sign, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, broken bar, macron, diaeresis, acute accent, multiplication sign, soft hyphen, superscript 1, superscript 2, superscript 3, division sign</p>

Table 14–15. Spanish Ccsversion Definitions

Elements	Characters
Lowercase set	a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, lowercase Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, lowercase Icelandic thorn, q, r, s, German sharp s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y diaeresis, y acute, z
Uppercase set	A, A circumflex, A diaeresis, A grave, A acute, A tilde, A ring above, A with E, B, C, C cedilla, D, uppercase Icelandic eth, E, E circumflex, E diaeresis, E grave, E acute, F, G, H, I, I circumflex, I diaeresis, I grave, I acute, J, K, L, M, N, N tilde, O, O circumflex, O diaeresis, O grave, O acute, O tilde, O with oblique stroke, P, uppercase Icelandic thorn, Q, R, S, T, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Y acute, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a diaeresis-A diaeresis, a grave-A grave, a acute-A acute, a tilde-A tilde, a ring above-A ring above, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, lowercase Icelandic eth-uppercase Icelandic eth, e-E, e circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, i grave-I grave, i acute-I acute, j-J, k-K, l-L, m-M, n-N, n tilde-N tilde, o-O, o circumflex-O circumflex, o diaeresis-O diaeresis, o grave-O grave, o acute-O acute, o tilde-O tilde, o with oblique stroke-O with oblique stroke, p-P, lowercase Icelandic thorn-uppercase Icelandic thorn, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, u acute-U acute, v-V, w-W, x-X, y-Y, y acute-Y acute, y diaeresis-Y, z-Z
Uppercase to lowercase mapping	A-a, A circumflex-a circumflex, A diaeresis-a diaeresis, A grave-a grave, A acute-a acute, A tilde-a tilde, A ring above-a ring above, A with E-a with e, B-b, C-c, C cedilla-c cedilla, D-d, uppercase Icelandic eth-lowercase Icelandic eth, E-e, E circumflex-e circumflex, E diaeresis-e diaeresis, E grave-e grave, E acute-e acute, F-f, G-g, H-h, I-i, I circumflex-i circumflex, I diaeresis-i diaeresis, I grave-i grave, I acute-i acute, J-j, K-k, L-l, M-m, N-n, N tilde-n tilde, O-o, O circumflex-o circumflex, O diaeresis-o diaeresis, O grave-o grave, O acute-o acute, O tilde-o tilde, O with oblique stroke-o with oblique stroke, P-p, lowercase Icelandic thorn-uppercase Icelandic thorn, Q-q, R-r, S-s, T-t, U-u, U circumflex-u circumflex, U diaeresis-u diaeresis, U grave-u grave, U acute-u acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Y diaeresis-y, Z-z

Table 14–15. Spanish Ccsversion Definitions

Elements	Characters
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>[Space, inverted exclamation mark], !, ", left double-angle quote, right double-angle quote, #, \$, cent sign, pound sign, euro sign, yen sign, %, ampersand, apostrophe, (,), *, +, plus-minus sign, multiplication sign, division sign, comma, -, full stop, middle dot, /, colon, semicolon, <, =, >, inverted question mark, ?, @, paragraph sign, pilcrow sign, left square bracket, \, right square bracket, circumflex accent, _, macron, grave accent, acute accent, diaeresis, {, vertical line, broken bar, }, tilde, DEL, copyright sign, registered trademark sign, feminine ordinal indicator, masculine ordinal indicator, soft hyphen, degree sign, superscript 1, superscript 2, superscript 3, micro sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, [A, A grave, A acute, A circumflex, A tilde, A diaeresis, A ring above], [a, a grave, a acute, a circumflex, a tilde, a diaeresis, a ring above], B, b, [C, C cedilla], [c, c cedilla], D, d, [E, E grave, E acute, E circumflex, E diaeresis], [e, e grave, e acute, e circumflex, e diaeresis], F, f, G, g, H, h, [I, I grave, I acute, I circumflex, diaeresis], [i, i grave, i acute, i circumflex, i diaeresis], J, j, K, k, L, l, M, m, N, n, N tilde, n tilde, [O, O grave, O acute, O circumflex, O tilde, O diaeresis, O with oblique stroke], [o, o grave, o acute, o circumflex, o tilde, o diaeresis, o with oblique stroke], P, p, Q, q, R, r, S, s, T, t, [U, U grave, U acute, U circumflex, U diaeresis], [u, u grave, u acute, u circumflex, u diaeresis], V, v, W, w, X, x, [Y, Y acute], [y, y acute, y diaeresis], Z, z, A with E, a with e, German sharp s, uppercase Icelandic eth, lowercase Icelandic eth, uppercase Icelandic thorn, lowercase Icelandic thorn</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Spanish ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Swedish1 Ccsversion Definitions

Table 14–16 lists the definitions for the Swedish1 ccsversion. The ccsversion number for Swedish1 is 99. The coded character set base is Latin1EBCDIC.

Table 14–16. Swedish1 Ccsversion Definitions

Elements	Characters
Spaces data class	Space, no-break space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, no-break space, left double-angle quote, right double-angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, soft hyphen, superscript 1, superscript 2, superscript 3, division sign</p>

Table 14–16. Swedish1 Ccsversion Definitions

Elements	Characters
Lowercase set	a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, lowercase Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, lowercase Icelandic thorn, q, r, s, German sharp s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y diaeresis, y acute, z
Uppercase set	A, A circumflex, A diaeresis, A grave, A acute, A tilde, A ring above, A with E, B, C, C cedilla, D, uppercase Icelandic eth, E, E circumflex, E diaeresis, E grave, E acute, F, G, H, I, I circumflex, I diaeresis, I grave, I acute, J, K, L, M, N, N tilde, O, O circumflex, O diaeresis, O grave, O acute, O tilde, O with oblique stroke, P, uppercase Icelandic thorn, Q, R, S, T, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Y acute, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a diaeresis-A diaeresis, a grave-A grave, a acute-A acute, a tilde-A tilde, a ring above-A ring above, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, lowercase Icelandic eth-uppercase Icelandic eth, e-E, e circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, i grave-I grave, i acute-I acute, j-J, k-K, l-L, m-M, n-N, n tilde-N tilde, o-O, o circumflex-O circumflex, o diaeresis-O diaeresis, o grave-O grave, o acute-O acute, o tilde-O tilde, o with oblique stroke-O with oblique stroke, p-P, lowercase Icelandic thorn-uppercase Icelandic thorn, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, u acute-U acute, v-V, w-W, x-X, y-Y, y acute-Y acute, y diaeresis-Y, z-Z
Uppercase to lowercase mapping	A-a, A circumflex-a circumflex, A diaeresis-a diaeresis, A grave-a grave, A acute-a acute, A tilde-a tilde, A ring above-a ring above, A with E-a with e, B-b, C-c, C cedilla-c cedilla, D-d, uppercase Icelandic eth-lowercase Icelandic eth, E-e, E circumflex-e circumflex, E diaeresis-e diaeresis, E grave-e grave, E acute-E acute, F-f, G-g, H-h, I-i, I circumflex-i circumflex, I diaeresis-i diaeresis, I grave-i grave, I acute-i acute, J-j, K-k, L-l, M-m, N-n, N tilde-n tilde, O-o, O circumflex-o circumflex, O diaeresis-o diaeresis, O grave-o grave, O acute-o acute, O tilde-o tilde, O with oblique stroke-o with oblique stroke, P-p, lowercase Icelandic thorn-uppercase Icelandic thorn, Q-q, R-r, S-s, T-t, U-u, U circumflex-u circumflex, U diaeresis-u diaeresis, U grave-u grave, U acute-u acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z

Table 14–16. Swedish1 Ccsversion Definitions

Elements	Characters
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>NUL, SOH, STX, ETX, ST, HT, SSA, APC, EPA, RI, SS2, VT, FF, CR, DLE, DC1, DC2, DC3, OSC, NEL, BS, ESA, CAN, EM, PU2, SS3, IS4, IS3, IS2, IS1, BPH, NBH, IND, LF, ETB, ESC, HTS, HTJ, VTS, PLD PLU, ENQ, ACK, BEL, DCS, PU1, SYN, STS, CCH, MW, SPA, EOT, SOS, SCI, CSI, DC4, NAK, PM, SUB, [space, no-break space], [A, A circumflex, A grave, A acute, A tilde, a, a circumflex, a grave, a acute, a tilde], [B, b], [C, C cedilla, c, c cedilla], [D, uppercase Icelandic eth, d, lowercase Icelandic eth], [E, E acute, E circumflex, E diaeresis, E grave, e, e acute, e circumflex, e diaeresis, e grave], [F, f], [G, g], [H, h], [I, I acute, I circumflex, I diaeresis, I grave, i, i acute, i circumflex, i diaeresis, i grave], [J, j], [K, k], [L, l], [M, m], [N, N tilde, n, n tilde], [O, O circumflex, O grave, O acute, O tilde, o, o circumflex, o grave, o acute, o tilde], [P, uppercase Icelandic thorn, p, lowercase Icelandic thorn], [Q, q], [R, r], [S, s, German sharp s], [T, t], [U, U circumflex, U grave, U acute, u, u circumflex, u grave, u acute], [V, W, v, w], [X, x], [Y, U diaeresis, Y acute, y, u diaeresis, y acute, y diaeresis], [Z, z][A ring above, a ring above], [A diaeresis, A with E, a diaeresis, a with e], [O diaeresis, O with oblique stroke, o diaeresis, o with oblique stroke], 0, 1, 2, 4, 3, 4, 5, 6, 7, 8, 9, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, soft hyphen, /, vertical line, comma, %, _, >, ?, grave accent, colon, #, @, apostrophe, =, ", left double angle quote, right double angle quote, plus-minus sign, degree sign, feminine ordinal indicator, masculine ordinal indicator, cedilla, euro sign, micro sign, tilde, inverted exclamation mark, inverted question mark, registered trademark sign, cent sign, pound sign, yen sign, middle dot, copyright sign, paragraph sign, pilcrow sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, not sign, broken bar, macron, diaeresis, acute accent, multiplication sign, {, }, superscript 1, \, division sign, superscript 2, superscript 3, DEL</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Swedish1 ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Swiss Ccsversion Definitions

Table 14–17 lists the definitions for the Swiss ccsversion. The ccsversion number for Swiss is 64. The coded character set base is Latin1EBCDIC.

Table 14–17. Swiss Ccsversion Definitions

Elements	Characters
Spaces data class	Space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, Icelandic thorn, q, r, s, t, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y acute, z</p> <p>Lowercase German sharp s, y diaeresis</p> <p>Soft hyphen</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>a, a acute, a circumflex, a diaeresis, a grave, a tilde, b, c, c cedilla, d, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i acute, i circumflex, i diaeresis, i grave, j, k, l, m, n, n tilde, o, o acute, o circumflex, o diaeresis, o grave, o tilde, o with oblique stroke, p, q, r, s, German sharp s, t, u, u acute, u circumflex, u diaeresis, u grave, v, w, x, y, y acute, y diaeresis, z</p> <p>A, A Acute, A circumflex, A diaeresis, A grave, A tilde, B, C, C cedilla, D, E, E acute, E circumflex, E diaeresis, E grave, F, G, H, I, I acute, I circumflex, I diaeresis, I grave, J, K, L, M, N, N tilde, O, O acute, O circumflex, O, O diaeresis, O grave, O tilde, P, Q, R, S, T, U, U acute, U circumflex, U diaeresis, U grave, V, W, X, Y, Y acute, Z</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", degree sign, euro sign, {, }, \, no-break space, cent sign, pound sign, paragraph sign, not sign, broken bar, diaeresis, acute accent, soft hyphen</p>

Table 14–17. Swiss Ccsversion Definitions

Elements	Characters
Lowercase set	a, a circumflex, a diaeresis, a grave, a acute, a tilde, a ring above, a with e, b, c, c cedilla, d, lowercase Icelandic eth, e, e circumflex, e diaeresis, e grave, e acute, f, g, h, i, i circumflex, i diaeresis, i grave, i acute, j, k, l, m, n, n tilde, o, o circumflex, o diaeresis, o grave, o acute, o tilde, o with oblique stroke, p, q, r, s, German sharp s, t, lowercase Icelandic thorn, u, u circumflex, u diaeresis, u grave, u acute, v, w, x, y, y diaeresis, y acute, z
Uppercase set	A, A circumflex, A diaeresis, A grave, A acute, A tilde, A ring above, A with E, B, C, C cedilla, D, uppercase Icelandic eth, E, E circumflex, E diaeresis, E grave, E acute, F, G, H, I, I circumflex, I diaeresis, I grave, I acute, J, K, L, M, N, N tilde O, O circumflex, O diaeresis, O grave, O acute, O tilde, O with oblique stroke, P, Q, R, S, T, uppercase Icelandic thorn, U, U circumflex, U diaeresis, U grave, U acute, V, W, X, Y, Y acute, Z
Lowercase to uppercase mapping	a-A, a circumflex-A circumflex, a diaeresis-A diaeresis, a grave-A grave, a acute-A acute, a tilde-A tilde, a ring above-A ring above, a with e-A with E, b-B, c-C, c cedilla-C cedilla, d-D, lowercase Icelandic eth- uppercase Icelandic eth, e-E, e circumflex-E circumflex, e diaeresis-E diaeresis, e grave-E grave, e acute-E acute, f-F, g-G, h-H, i-I, i circumflex-I circumflex, i diaeresis-I diaeresis, i grave-I grave, i acute-I acute, j-J, k-K, l-L, m-M, n-N, n tilde-N tilde, o-O, o circumflex-O circumflex, o diaeresis-O diaeresis, o grave-O grave, o acute-O acute, o tilde-O tilde, o with oblique stroke-O with oblique stroke, p-P, lowercase Icelandic thorn- uppercase Icelandic thorn, q-Q, r-R, s-S, t-T, u-U, u circumflex-U circumflex, u diaeresis-U diaeresis, u grave-U grave, u acute-U acute, v-V, w-W, x-X, y-Y, y acute-Y acute, y diaeresis-Y, z-Z

Table 14–17. Swiss Ccsversion Definitions

Elements	Characters
Uppercase to lowercase mapping	<p>A-a, A circumflex-a circumflex, A diaeresis-a diaeresis, A grave-a grave, A acute-a acute, A tilde-a tilde, A ring above-a ring above, A with E-a with e, B-b, C-c, C cedilla-c cedilla, D-d, uppercase Icelandic eth-lowercase Icelandic eth, E-e, E circumflex-e circumflex, E diaeresis-e diaeresis, E grave-e grave, E acute-E acute, F-f, G-g, H-h, I-i, I circumflex-i circumflex, I diaeresis-i diaeresis, I grave-i grave, I acute-i acute, J-j, K-k, L-l, M-m, N-n, N tilde-n tilde, O-o, O circumflex-o circumflex, O diaeresis-o diaeresis, O grave-o grave, O acute-o acute, O tilde-o tilde, O with oblique stroke-o with oblique stroke, P-p, Q-q, R-r, S-s, T-t, uppercase Icelandic thorn-lowercase Icelandic thorn, U-u, U circumflex-u circumflex, U diaeresis-u diaeresis, U grave-u grave, U acute-u acute, V-v, W-w, X-x, Y-y, Y acute-y acute, Z-z</p>
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>NUL, SOH, STX, ETX, ST, HT, SSA, APC, EPA, RI, SS2, VT, FF, CR, DLE, DC1, DC2, DC3, OSC, NEL, BS, ESA, CAN, EM, PU2, SS3, IS4, IS3, IS2, IS1, BPH, NBH, IND, LF, ETB, ESC, HTS, HTJ, VTS, PLD PLU, ENQ, ACK, BEL, DCS, PU1, SYN, STS, CCH, MW SPA, EOT, SOS, SCI, CSI, DC4, NAK, PM, SUB, [space, no-break space],</p>

Table 14–17. Swiss Ccsversion Definitions

Elements	Characters
Character collating (<i>Cont.</i>)	left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, [-, soft hyphen], /, vertical line, comma, %, _, >, ?, grave accent, colon, #, @, apostrophe, =, ", tilde, {, }, \, left double angle quote, right double angle quote, inverted exclamation mark, inverted question mark, degree sign, feminine ordinal indicator, masculine ordinal indicator, broken bar, not sign, macron, diaeresis, acute accent, cedilla, registered trademark sign, copyright sign, paragraph sign, pilcrow sign, micro sign, cent sign, pound sign, yen sign, euro sign, middle dot, multiplication sign, division sign, plus-minus sign, vulgar fraction one quarter, vulgar fraction one half, vulgar fraction three quarters, superscript 1, superscript 2, superscript 3, [A, a, A acute, a acute, A grave, a grave, A circumflex, a circumflex, A diaeresis, a diaeresis, A tilde, a tilde, A ring above, a ring above], [B, b], [C, c, C cedilla, c cedilla], [D, d, uppercase Icelandic eth, lowercase Icelandic eth], [E, e, E acute, e acute, E grave, e grave, E circumflex, e circumflex, E diaeresis, e diaeresis], [F, f] [G, g], [H, h], [I, i, I acute, i acute, I grave, i grave, I circumflex, i circumflex, I diaeresis, i diaeresis], [J, j], [K, k], [L, l], [M, m], [N, n, N tilde, n tilde], [O, o, O acute, o acute, O grave, o grave, O circumflex, o circumflex, O diaeresis, o diaeresis, O tilde, o tilde, O with oblique stroke, o with oblique stroke], [P, p], [Q, q], [R, r], [S, s, German sharp s], [T, t, uppercase Icelandic thorn, lowercase Icelandic thorn], [U, u, U acute, u acute, U grave, u grave, U circumflex, u circumflex, U diaeresis, u diaeresis], [V, v], [W, w], [X, x], [Y, y, Y acute, y acute, y diaeresis], [Z, z], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, DEL
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Swiss ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Turkish Ccsversion Definitions

Table 14–18 lists the definitions for the Turkish ccsversion. The ccsversion number for Turkish is 80. The coded character set base is Latin5EBCDIC.

Table 14–18. Turkish Ccsversion Definitions

Elements	Characters
Spaces data class	Space
Alphabetic data class	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, b, c, c cedilla, d, e, f, g, g breve, h, i, j, k, l, m, n, o, o diaeresis, p, q, r, s, s cedilla, t, u, u diaeresis, v, w, x, y, z</p> <p>Lowercase i without dot above, uppercase I with dot above</p>
Numeric data class	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Presentation set	<p>Lowercase and uppercase forms of the following Latin letters:</p> <p>a, b, c, c cedilla, d, e, f, g, g breve, h, i, j, k, l, m, n, o, o diaeresis, p, q, r, s, s cedilla, t, u, u diaeresis, v, w, x, y, z</p> <p>Lowercase i without dot above, uppercase I with dot above</p> <p>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</p> <p>Space, left square bracket, full stop, <, (, +, !, ampersand, right square bracket, \$, *,), semicolon, circumflex accent, -, /, vertical line, comma, %, _, colon, >, ?, grave accent, #, @, apostrophe, =, tilde, ", {, }, \, acute accent</p>
Lowercase set	a, b, c, c cedilla, d, e, f, g, g breve, h, i, i without dot above, j, k, l, m, n, o, o diaeresis, p, q, r, s, s cedilla, t, u, u diaeresis, v, w, x, y, z
Uppercase set	A, B, C, C cedilla, D, E, F, G, G breve, H, I, I with dot above, J, K, L, M, N, O, O diaeresis, P, Q, R, S, S cedilla, T, U, U diaeresis, V, W, X, Y, Z
Lowercase to uppercase mapping	a-A, b-B, c-C, c cedilla-C cedilla, d-D, e-E, f-F, g-G, g breve-G breve, h-H, i-I, i without dot above-I, j-J, k-K, l-L, m-M, n-N, o-O, o diaeresis-O diaeresis, p-P, q-Q, r-R, s-S, s cedilla-S cedilla, t-T, u-U, u diaeresis-U diaeresis, v-V, w-W, x-X, y-Y, z-Z

Table 14-18. Turkish Ccsversion Definitions

Elements	Characters
Uppercase to lowercase mapping	A-a, B-b, C-c, C cedilla-c cedilla, D-d, E-e, F-f, G-g, G breve-g breve, H-h, I-i, I with dot above-i, J-j, K-k, L-l, M-m, N-n, O-o, O diaeresis-o diaeresis, P-p, Q-q, R-r, S-s, S cedilla-s cedilla, T-t, U-u, U diaeresis-u diaeresis, V-v, W-w, X-x, Y-y, Z-z
Character collating	<p>Note that in the character collating sequence, the series of characters within square brackets ([]) are given the same rank in equivalent ordering.</p> <p>Space, !, ", #, \$, %, ampersand, apostrophe, (,), *, +, comma, -, full stop, /, 0, 1, 2,, 3, 4, 5, 6, 7, 8, 9, colon, semicolon, <, =, >, ?, @, [A, A circumflex, a, a circumflex],[B, b], [C, c], [C cedilla, c cedilla], [D, d], [E, e], [F, f], [G, g], [G breve, g breve], [H, h], [I, i without dot above], [I with dot above, I circumflex, i, i circumflex], [J, j], [K, k], [L, l], [M, m], [N, n], [O, o], [O diaeresis, o diaeresis], [P, p], [Q, q], [R, r], [S, s], [S cedilla, s cedilla], [T, t], [U, u], [U diaeresis, u diaeresis], [V, v], [W, w], [X, x], [Y, y], [Z, z], left square bracket, right square bracket, \, circumflex accent, _, {, }, vertical line, tilde</p>
Substitutions	No concept
Alternate digits	No concept

The following escapement definitions apply to the Turkish ccsversion:

Text line orientation	Horizontal
Line advance direction	Top-to-bottom (+)
Character advance direction	All characters left-to-right (+)

Section 15

Standard Conventions

The release media contains standard conventions in the file SYSTEM/CONVENTIONS. You cannot modify or delete the standard conventions supplied on the release media. However, you can add custom conventions to the SYSTEM/CONVENTIONS file.

This section presents the convention definitions for the data items associated with each of the standard conventions as well as a brief explanation of the coding. The standard conventions are presented in alphabetical order.

Refer to Section 3, "Understanding Templates in Convention Definitions," for information about the contents of conventions.

Argentina Convention Definition

ARGENTINA	%Convention name
!d! !! !n! !x! !yyyy!	%Long date
!d! !n! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[A]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"ARA"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Argentina convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, A432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –A432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

ASeriesNative Convention Definition

The ASeriesNative convention is the default convention on the release media.

```

ASERIESNATIVE           %Convention name
!w!, !n! !d!, !yyyy!   %Long date
!1w!, !1n! !d!, !yyyy! %Short date
!0o!/!0d!/!yyyy!      %Numeric date
!0t!:!0m!:!0s!.!pppp! %Long time
!0t!:!0m!:!0s!        %Numeric time
!n[-] t[:,0,3] d[.] #! %Numeric specifications
!n[-] c[$] t[:,0,3] d[.] #! %Monetary specifications
22                      %Maximum number of monetary integer digits
2                      %Number of monetary fractional digits
2                      %International monetary fractional digits
"USD "                 %International currency notation
60                      %Default lines per page (page length)
132                     %Default characters per line (page width)
END

```

For the ASeriesNative convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$43.52.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$43.52.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Australia Convention Definition

AUSTRALIA	%Convention name
!n! !de!, !yyyy!	%Long date
!dd!-!n!-!0y!	%Short date
!d!/!o!/!0y!	%Numeric date
!h!:!m!:!s!	%Long time
!h!:!m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[\$]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"\$A"	%International currency notation
64	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Australia convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Belgium Convention Definition

Belgium	%Convention name
!w! !0d! !n! !0y!	%Long date
!lw!, !0d! !n!, !0yyy!	%Short date
!d!-!o!-!0y!	%Numeric date
!0t!:!0m!:!0s!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[F]t[.:0,3]d[,]#!	%Monetary specifications
10	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"BF"	%International currency notation
70	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Belgium convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, F43,52.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –F43,52.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Brazil Convention Definition

BRAZIL	%Convention name
!d! !! !n! !x! !yyyy!	%Long date
!0d! !1n! !yyyy!	%Short date
!0d! !0o! !.yyyy!	%Numeric date
!t! !i! !0m! !k!	%Long time
!t! !0m!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[Cr\$]t[.:0,3]d[,]#!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"BRN"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Brazil convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, Cr\$432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Chile Convention Definition

CHILE	%Convention name
!d! !l! !n! !x! !yyyy!	%Long date
!d! !ln! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[\$]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"CLP"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Chile convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

China Convention Definition

CHINA	%Convention name
!w! !n! !d!, !yyyy!	%Long date
!yyyy!.!0o!.!0d!	%Short date
!0y!.!0o!.!0d!	%Numeric date
!0t!.!0m!.!0s!	%Long time
!0t!.!0m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[48"5B"]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"RMB"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the China convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Colombia Convention Definition

COLOMBIA	%Convention name
!d! !l! !n! !x! !yyyy!	%Long date
!d! !ln! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[\$]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"COL"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Colombia convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

CostaRica Convention Definition

COSTARICA	%Convention name
!d! !! !n! !x! !yyyy!	%Long date
!d! !n! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[48"B0"]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"CRC"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the CostaRica convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Croatia Convention Definition

CROATIA	%Convention name
!	%Long date
!	%Short date
!dd!.!oo!.!0y!	%Numeric date
!	%Long time
!hh!:!mm!:!ss!	%Numeric time
!p[+]n[-]t[.:3,3]d[,,#]!	%Numeric specifications
!p[+]n[-]t[.:3,3]d[,,#]c[CRD]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"CRD"	%International currency notation
72	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Croatia convention, following is a brief explanation of the coding:

- If a number or monetary amount is positive, the positive sign (+) precedes the amount; for example, +24.
- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example +432CRD.
- The thousands separator is a period (.) and the decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Cyrillic Convention Definition

CYRILLIC	%Convention name
!w!, !d! !n! !yyy!	%Long date
!1w!, !d! !1n! !yyy!	%Short date
!0d!.!0o!.!0y!	%Numeric date
!TT! !I! !MM! !K! !SS!.!PP! !R!	%Long time
!0t!.!0m!	%Numeric time
!N[-]T[, :0,3]D[.]#!	%Numeric specification
!N[-]T[, :0,3]D[.]#C[48"626543"]!	%Monetary specification
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"RUR"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Cyrillic convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol (coded as a hexadecimal) follows the monetary amount.
- The thousands separator is a comma. The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Czechoslovakia Convention Definition

CZECHOSLOVAKIA	%Convention name
!w!, !d!. !n! !yyyy!	%Long date
!d!. !o!. !yyyy!	%Short date
!yyyy!-!0o!-!0d!	%Numeric date
!0t!:!0m!:!0s!,!p!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[48"40":0,3]d[,]#!	%Numeric specifications
!n[-]c[48"D247A2"]t[48"40":0,3]d[,]#!	%Monetary specifications
14	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"CSK"	%International currency notation
30	%Default lines per page (page length)
60	%Default characters per line (page width)
END	

For the Czechoslovakia convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol (coded as a hexadecimal) precedes the monetary amount.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (.). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Denmark Convention Definition

Denmark	%Convention name
!w! !c! !d!.!n! !0yyy!	%Long date
!1w!. !0d! !1n! !0yyy!	%Short date
!0d! !0o! !0yyy!	%Numeric date
!0t!:!0m!:!0s!.!pp! !a!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[kr.]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"DKR"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Denmark convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, kr.43,52.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –kr.43,52.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Egypt Convention Definition

EGYPT	%Convention name
!w!, !n! !d!, !yyyy!	%Long date
!1w!, !1n! !d!, !yyyy!	%Short date
!0oo!/!0dd!/!yyyy!	%Numeric date
!0t!:!0m!:!0s!.!pppp!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[48"40":0,3]d[,]#!	%Numeric specifications
!n[-]c[L.E.]t[48"40":0,3]d[,]#!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"L.E."	%International currency notation
71	%Default lines per page (page length)
81	%Default characters per line (page width)
END	

For the Egypt convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, L.E.432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –L.E.432.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (.). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

EnglishCanada Convention Definition

ENGLISHCANADA	%Convention name
!w!, !n! !d!, !yyyy!	%Long date
!1w!, !1n! !d!, !yyyy!	%Short date
!0o!/!0d!/!yyyy!	%Numeric date
!0t!:!0m!:!0s!.!pppp!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[\$]t[, :0,3]d[.]#!	%Monetary specifications
22	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
3	%International monetary fractional digits
"CAD"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the EnglishCanada convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

EuropeanStandard Convention Definition

EuropeanStandard	%Convention name
!w!, !c! !de! !n!, !0yyy!	%Long date
!1w!, !d! !1n!, !0yyy!	%Short date
!0yyy!-!0o!-!0d!	%Numeric date
!0t!:!0m!:!0s!.!pp! !a!	%Long time
!0t!:!0m!	%Numeric time
!t[48"40":0,3]d[,]#n[-]!	%Numeric specifications
!c[\$]t[48"40":0,3]d[,]#n[-]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"ECU"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the EuropeanStandard convention, note the following:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example, \$432.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (,). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

FranceBureautique Convention Definition

FranceBureautique	%Convention name
!w! !de! !n! !0yyy!	%Long date
!1w! !d! !1n! !0yyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!t!!h!0m!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]t[.:0,3]d[,]#c[F]!	%Monetary specifications
13	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"FRF"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the FranceBureautique convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 435F.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

FranceListing Convention Definition

FranceListing	%Convention name
!w! !d! !n! !0yyy!	%Long date
!1w! !d! !1n! !0yyy!	%Short date
!0d!/!0o!/!0yyy!	%Numeric date
!t! !i! !m! !k! !s!,!pp! !r!	%Long time
!0t! !h! !0m!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]t[.:0,3]d[,]#c[F]!	%Monetary specifications
13	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"FRF"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the FranceListing convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 435F.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

FrenchCanada Convention Definition

FRENCHCANADA	%Convention name
!w!, !de! !n! !0yyy!	%Long date
!1w! !d! !1n! !0yyy!	%Short date
!0yyy! !0o! !0d!	%Numeric date
!0t!:!0m!:!0s!	%Long time
!t!!h!!0m!	%Numeric time
!n[-]t[48"40":0,3]d[,]#!	%Numeric specifications
!n[-]t[48"40":0,3]d[,]#c[\$]!	%Monetary specifications
13	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"CAD"	%International currency notation
66	%Default lines per page (page length)
85	%Default characters per line (page width)
END	

For the FrenchCanada convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 432\$.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (,). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Greece Convention Definition

Greece	%Convention name
!w!, !de! !n!, !yyyy!	%Long date
!1w!, !d! !1n! !Yyyy!	%Short date
!d!-!o!-!0y!	%Numeric date
!0t!:!0m!:!0s!,!pp!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[48"40":0,3]d[,]#!	%Numeric specifications
!n[-]c[Dr.]t[48"40":0,3]d[,]#!	%Monetary specifications
14	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"Dr."	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Greece convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, Dr.432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –Dr.432.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (.). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

HongKong Convention Definition

HONGKONG	%Convention name
!w!, !do! !n!, !yyyy!	%Long date
!d! !1n! !0y!	%Short date
!d!.!oo!.!0y!	%Numeric date
!h!:!0m!:!0s!:!pp! !a!	%Long time
!h!:!0m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[\$]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"\$"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the HongKong convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$24.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example –\$24.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Hungary Convention Definition

HUNGARY	%Convention name
!yyy!..!0o!..!0d!	%Long date
!0y!..!0o!..!0d!	%Short date
!yyy! !0o! !0d!	%Numeric date
!0t!..!0m!..!0s!	%Long time
!0t! !0m! !0s!	%Numeric time
!p[+]N[-]t[.:0,3]d[,,#]!	%Numeric specifications
!p[+]n[-]t[.:0,3]d[,,#]c[Ft]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"F1t"	%International currency notation
72	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Hungary convention, following is a brief explanation of the coding:

- If a number or monetary amount is positive, the positive sign (+) precedes the amount; for example, +24.
- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example –432Ft.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Italy Convention Definition

Italy	%Convention name
!dd! !n!, !yyyy!	%Long date
!0d!/!0o!/!0y!	%Short date
!0d!/!0o!/!0y!	%Numeric date
!t!:!0m!:!0s!	%Long time
!t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[L]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"LIT"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Italy convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, L432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –L432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Japan1 Convention Definition

Japan1	%Convention name
!w! !n! !d!, !yyyy!	%Long date
!1n! !d!, !yyyy!	%Short date
!o!/!d!/!0y!	%Numeric date
!0t!:!0m!:!0s!.!ppp!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[48"5B"]t[, :0,3]d[.]#!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"YEN"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Japan1 convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol (the yen symbol coded as a hexadecimal) precedes the monetary amount.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –YEN432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Japan2 Convention Definition

Japan2	%Convention name
!w! !n! !d!, !yyyy!	%Long date
!1n! !d!, !yyyy!	%Short date
!0y!/!o!/!d!	%Numeric date
!0t!:!0m!:!0s!.!ppp!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[48"5B"]t[, :0,3]d[.]#!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"YEN"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Japan2 convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol (the yen symbol coded as a hexadecimal) precedes the monetary amount.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –YEN432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Jordan Convention Definition

```

JORDAN                %Convention name
!w!, !c! !de! !! !n! !yyyy! %Long date
!lw!, !d! !n! !yyyy! %Short date
!d! !o! !yyyy! %Numeric date
!h!:!m!:!s!.!ppp! !a! %Long time
!t!:!m!:!s! %Numeric time
!n[-]d[,,#]! %Numeric specifications
!n[-]c[JD]d[,,#]! %Monetary specifications
12 %Maximum number of monetary integer digits
3 %Number of monetary fractional digits
3 %International monetary fractional digits
"JD" %International currency notation
66 %Default lines per page (page length)
132 %Default characters per line (page width)
END

```

For the Jordan convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, JD24.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example –JD432.
- A thousands separator is not used.
- The decimal symbol is a comma (.). For example, the value 5432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Kenya Convention Definition

Kenya	%Convention name
!w!, !c! !de!, !0yyy!	%Long date
!1w!, !d! !1n!, !0yyy!	%Short date
!0yyy! !0o! !0d!	%Numeric date
!0t!:!0m!:!0s!.!pp! !a!	%Long time
!t! !0m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]t[, :0,3]d[.]#c[KSH]!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"KSH"	%International currency notation
71	%Default lines per page (page length)
81	%Default characters per line (page width)
END	

For the Kenya convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 432KSH.
- The thousands separator is a comma (.). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Kuwait Convention Definition

KUWAIT	%Convention name
!w!, !de! !n! !yyyy!	%Long date
!lw!, !d! !n! !yyyy!	%Short date
!d! !o! !yyyy!	%Numeric date
!h!:!m!:!s!.!ppp! !a!	%Long time
!t!:!m!:!s!	%Numeric time
!n[-]t[, :7,3]d[.]#!	%Numeric specifications
!n[-]c[KD]t[, :7,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
3	%Number of monetary fractional digits
2	%International monetary fractional digits
"KD"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Kuwait convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, KD24.
- In instances where both a negative sign and a currency symbol are needed, the national currency symbol precedes the monetary amount and follows the negative sign; for example –KD432.
- The thousands separator is a comma (,). However, seven digits must be present to the left of the decimal place, before the thousands separator is used.
- The decimal symbol is a period (.). For example, the value 5432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths. The value 1,234,567 represents the number one million two hundred thirty-four thousand, five hundred sixty seven.

Lebanon Convention Definition

LEBANON	%Convention name
!w!, !n! !d!, !yyyy!	%Long date
!1w!, !d! !1n! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!	%Long time
!	%Numeric time
!n[-]t[.:2,3]d[,]#!	%Numeric specifications
!n[-]c[LL]t[.:2,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"LBP"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Lebanon convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount and follows the negative sign; for example –LL432.
- The thousands separator is a period (.).
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Malaysia Convention Definition

MALAYSIA	%Convention name
!w!, !0d! !n! !yyyy!	%Long date
!0d! !n! !yyyy!	%Short date
!dd!.!0o!.!0yyy!	%Numeric date
!h!:!0m!:!0s!.!ppp!	%Long time
!h!:!0m!	%Numeric time
!t[, :0,3]d[.]#n[-]!	%Numeric specifications
!c[MYR]t[, :0,3]d[.]#n[-]!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"\$"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Malaysia convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example, MYR356 or MYR22–.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Mexico Convention Definition

MEXICO	%Convention name
!d! !! !n! !x! !yyyy!	%Long date
!d! !n! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[\$]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"MXP"	%International currency notation
60	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Mexico convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Netherlands Convention Definition

Netherlands	%Convention name
!w!, !d! !n! !yyyy!	%Long date
!1w! !d! !n! !yyyy!	%Short date
!d!.!o!.!0y!	%Numeric date
!t!.!m! !i!	%Long time
!0t!.!0m!	%Numeric time
!t[48"40":0,3]d[,]#n[-]!	%Numeric specifications
!c[\$]t[48"40":0,3]d[,]#n[-]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"HFL"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Netherlands convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example, \$43,52.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (,). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

NewZealand Convention Definition

NEWZEALAND	%Convention name
!w!, !c! !de! !! !n!, !yyyy!	%Long date
!lw!, !d! !ln! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!0t!:!0m!:!0s!.!ppp!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[\$]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"NZD"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the NewZealand convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, \$432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –\$432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Nigeria Convention Definition

NIGERIA	%Convention name
!lw!, !d! !ln! !yyyy!	%Long date
!d!, !ln! !yyyy!	%Short date
!dd!/!oo!/!0y!	%Numeric date
!0t!:!0m!:!0s!.!pp! !a!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!p[+]n[-]t[, :3,3]d[.]#!	%Numeric specifications
!p[+]n[-]c[N]t[, :3,3]d[.]#!	%Monetary specifications
10	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"NGN"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Nigeria convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- A number or monetary amount can be preceded by a positive sign (+); for example, +42.
- The national currency symbol precedes the monetary amount and after the positive or negative sign; for example, –N432.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Norway Convention Definition

Norway	%Convention name
!w! !d! !n! !0yyy!	%Long date
!1w! !d! !1n! !0y!	%Short date
!d!.!o!.!0y!	%Numeric date
!k! !0t!.!0m!.!0s!,!pp!	%Long time
!t! !0m!	%Numeric time
!t[48"40":0,3]d[,]#n[-]!	%Numeric specifications
!c[Kr.]t[48"40":0,3]d[,]#n[-]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"NKR"	%International currency notation
72	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Norway convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example, Kr.432.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (,). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Peru Convention Definition

PERU	%Convention name
!d! !l! !n! !x! !yyyy!	%Long date
!d! !ln! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[I/.]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"PEI"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Peru convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, I/.432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –I/.432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Philippines Convention Definition

PHILIPPINES	%Convention name
!1nnn! !dd!, !yyyy!	%Long date
!0o!/!0d!/!0y!	%Short date
!0o!/!0d!/!0y!	%Numeric date
!0t!:!0m!:!0s!	%Long time
!0t!:!0m!;!0s!	%Numeric time
!t[, :0,3]d[.]#n[-]!	%Numeric specifications
!c[P]t[, :0,3]d[.]#n[-]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"PHP"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Philippines convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example P384.
- The thousands separator is a comma (.). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Poland Convention Definition

POLAND	%Convention name
!c! !d! !n! !yyyy! !z!	%Long date
!d!.!o!.!0y!	%Short date
!yyyy!-!0o!-!0d!	%Numeric date
!0t!:!0m!:!0s!,!p!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[48"40":0,3]d[,]#!	%Numeric specifications
!n[-]t[48"40":0,3]d[,]#c[48"A99A"]!	%Monetary specifications
16	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"PLZ"	%International currency notation
60	%Default lines per page (page length)
60	%Default characters per line (page width)
END	

For the Poland convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24
- The national currency symbol, 48"A99A" (coded as a hexadecimal), follows the monetary amount.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a comma (.). For example, the value 432,19 represents four hundred and thirty two and nineteen hundredths.

Romania Convention Definition

ROMANIA	%Convention name
!0d! !n! !yyyy!	%Long date
!0d!.!o!.!yyyy!	%Short date
!yyyy!-!0o!-!0d!	%Numeric date
!0t!:!0m!:!0s!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]t[.:0,3]d[,]#c[Lei]!	%Monetary specifications
14	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"Lei"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Romanian convention, following is a brief explanation of the coding:

- The negative sign (–) precedes a number or monetary amount as in –25.
- The national currency symbol, "Lei," follows the monetary amount as in 100Lei.
- The thousands separator is a period (.). For example, 1,000 is written as 1.000 and 1,000,000 is written as 1.000.000.
- The decimal symbol is a comma (,). For example, the value 432,19 represents four hundred and thirty two and nineteen hundredths.

SouthAfrica Convention Definition

SOUTHAFRICA	%Convention name
!wwwwwwww!, !dd! !nnnnnnnn!,!yyyy!	%Long date
!lw!, !d! !ln! !yyyy!	%Short date
!Od!.!Oo!.!Oy!	%Numeric date
!h!:!m!:!s! !aaaa!	%Long time
!h!:!0m!:!0s!	%Numeric time
!e[(,)]t[48"40":0,3]d[,]#!	%Numeric specifications
!e[(,)]c[R]t[48"40":0,3]d[,]#!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"ZAR"	%International currency notation
71	%Default lines per page (page length)
83	%Default characters per line (page width)
END	

For the SouthAfrica convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the amount is enclosed between parenthesis; for example, (24).
- The national currency symbol precedes the monetary amount; for example, R432 or R(432).
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (.). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Spain Convention Definition

Spain	%Convention name
!d! !! !n! !x! !yyyy!	%Long date
!d! !n! !yyyy!	%Short date
!d! !o! !yyyy!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t! !0m! !0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]t[.:0,3]d[,]#c[Pta]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"Pta"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Spain convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 52Pta.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

StandardTurkey Convention Definition

StandardTurkey	%Convention name
!w!, !d! !n!, !0yyy!	%Long date
!1w!, !d! !1n!, !0yyy!	%Short date
!d!/!o!/!0yyy!	%Numeric date
!0t!:!0m!:!0s!.!pp!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]t[.:0,3]d[,]#c[TL]!	%Monetary specifications
15	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"TRL"	%International currency notation
70	%Default lines per page (page length)
80	%Default characters per line (page width)

For the StandardTurkey convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 432TL or –432TL.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Sweden Convention Definition

Sweden	%Convention name
!w! !c! !d! !n! !0yyy!	%Long date
!1w! !d! !1n! !0yyy!	%Short date
!0yyy!-!0o!-!0d!	%Numeric date
!0t!:!0m!:!0s!,!pp!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[SEK]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"SEK"	%International currency notation
42	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Sweden convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, SEK52.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –SEK52.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Switzerland Convention Definition

Switzerland	%Convention name
!w!, !de! !n! !0yyy!	%Long date
!de! !n! !0yyy!	%Short date
!dd!.!0o!.!0yyy!	%Numeric date
!tt!:!0m!:!0s!.!ppp!	%Long time
!tt!:!0m!	%Numeric time
!n[-]t[':0,3]d[.]#!	%Numeric specifications
!c[Fr.]n[-]t[':0,3]d[.]#!	%Monetary specifications
13	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"SFR"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the Switzerland convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, Fr.52.
- In instances where both a negative sign and a currency symbol are needed, the currency symbol comes first, followed by the negative sign, followed by the amount; for example, Fr.–52.
- The thousands separator is a prime ('). The separator is used for all numbers and monetary amounts equal to or greater than 1'000. The thousands separator is used between each subsequent grouping of three digits, for example 1'000'000.
- The decimal symbol is a period (.). For example, the value 5'432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Taiwan Convention Definition

TAIWAN	%Convention name
!w!, !n! !d!, !yyyy!	%Long date
!1n! !d! !yyyy!	%Short date
!0o!/!0d!/!0y!	%Numeric date
!0t!:!0m!:!0s!.!ppp!	%Long time
!0t!:!0m!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[\$]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"NT\$"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Taiwan convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example \$57.
- In cases where both the negative sign and the currency symbol are needed, the currency symbol precedes the monetary amount and follows the negative sign.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

UAE Convention Definition

UAE	%Convention name
!z! !yyyy! !n! !l! !ed! !c! !w!	%Long date
!yyyy! !ln! !d! !lw!	%Short date
!yyyy!/!0o!/!0d!	%Numeric date
!r! !0s! !k! !0m! !i! !hh! !F!	%Long time
!0s! !0m!:!tt!	%Numeric time
!d[,,#p[+]n[-]!	%Numeric specifications
!d[,,#p[+]n[-]c[DHS]!	%Monetary specifications
12	%Maximum number of monetary integer digits
3	%Number of monetary fractional digits
2	%International monetary fractional digits
"DHS"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

Note: You must read the template from right to left.

For the UAE convention, following is a brief explanation of the coding:

- If a number or monetary amount is positive, the positive sign (+) precedes the amount; for example, +24.
- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount and precedes the positive or negative sign; for example DHS+432.
- The decimal symbol is a comma (.). A thousands separator is not used. For example, the value 5432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

UnitedKingdom1 Convention Definition

UnitedKingdom1	%Convention name
!w!, !de! !n!, !yyyy!	%Long date
!d! !1n! !0y!	%Short date
!d!.!0o!.!0y!	%Numeric date
!h!:!0m!:!0s!.!pp! !a!	%Long time
!h!:!0m! !a!	%Numeric time
!t[, :0,3]d[.]#n[-]!	%Numeric specifications
!c[\$]t[, :0,3]d[.]c[\$]#n[-]!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"\$"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the UnitedKingdom1 convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example, \$43.52.
- The thousands separator is a comma (.). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

UnitedKingdom2 Convention Definition

UnitedKingdom2	%Convention name
!w!, !de! !n! !yyyy!	%Long date
!d! !1n! !0y!	%Short date
!d!.!oo!.!0y!	%Numeric date
!h!:!0m!:!0s!.!pp! !a!	%Long time
!h!:!0m! !a!	%Numeric time
!n[-]t[, :0,3]d[.]#!	%Numeric specifications
!n[-]c[48"B1"]t[, :0,3]d[.]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"\$"	%International currency notation
70	%Default lines per page (page length)
82	%Default characters per line (page width)
END	

For the UnitedKingdom2 convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol (the pound symbol coded as a hexadecimal) precedes the monetary amount.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount.
- The thousands separator is a comma (,). The separator is used for all numbers and monetary amounts equal to or greater than 1,000. The thousands separator is used between each subsequent grouping of three digits, for example 1,000,000.
- The decimal symbol is a period (.). For example, the value 5,432.19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Venezuela Convention Definition

VENEZUELA	%Convention name
!d! !! !n! !x! !yyyy!	%Long date
!d! !n! !yyyy!	%Short date
!d!/!o!/!0y!	%Numeric date
!t! !i! !m! !k! !s! !r!	%Long time
!0t!:!0m!:!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]c[Bs]t[.:0,3]d[,]#!	%Monetary specifications
12	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"ZB"	%International currency notation
66	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Venezuela convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol precedes the monetary amount; for example, Bs432.
- In instances where both a negative sign and a currency symbol are needed, the negative sign comes first, followed by the currency symbol, followed by the amount; for example, –Bs432.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Yugoslavia Convention Definition

YUGOSLAVIA	%Convention name
!w!, !0d! !n! !yyyy!	%Long date
!1w! !0d! !1n! !0y!	%Short date
!0d!.!0o!.!0y!	%Numeric date
!0t!.!0m!.!0s!	%Long time
!0t!.!0m!.!0s!	%Numeric time
!n[-]t[.:0,3]d[,]#!	%Numeric specifications
!n[-]t[.:0,3]d[,]#c[Din]!	%Monetary specifications
10	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"YUD"	%International currency notation
72	%Default lines per page (page length)
80	%Default characters per line (page width)
END	

For the Yugoslavia convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) precedes the amount; for example, –24.
- The national currency symbol follows the monetary amount; for example, 432Din.
- The thousands separator is a period (.). The separator is used for all numbers and monetary amounts equal to or greater than 1.000. The thousands separator is used between each subsequent grouping of three digits, for example 1.000.000.
- The decimal symbol is a comma (,). For example, the value 5.432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Zimbabwe Convention Definition

Zimbabwe	%Convention name
!w! !0d! !n! !yyyy!	%Long date
!1w!, !d! !1n! !yyyy!	%Short date
!0d!/!0o!/!0y!	%Numeric date
!0t!:!0m!:!0s!	%Long time
!0t!:!0m!	%Numeric time
!t[48"40":0,3]d[,]#n[-]!	%Numeric specifications
!c[Z\$]t[48"40":0,3]d[,]#n[-]!	%Monetary specifications
9	%Maximum number of monetary integer digits
2	%Number of monetary fractional digits
2	%International monetary fractional digits
"\$"	%International currency notation
66	%Default lines per page (page length)
132	%Default characters per line (page width)
END	

For the Zimbabwe convention, following is a brief explanation of the coding:

- If a number or monetary amount is negative, the negative sign (–) follows the amount; for example, 24–.
- The national currency symbol precedes the monetary amount; for example, Z\$52.
- The thousands separator is a blank (coded as a hexadecimal). The separator is used for all numbers and monetary amounts equal to or greater than 1 000. The thousands separator is used between each subsequent grouping of three digits, for example 1 000 000.
- The decimal symbol is a comma (,). For example, the value 5 432,19 represents the number five thousand four hundred and thirty two and nineteen hundredths.

Section 16

The XlateEBCDIC Server

XlateEBCDIC is a Windows server that offers an efficient means of mapping ClearPath host data to and from UCS2NT or local workstation codepage. For example, it can map CODEPAGE1252 data to Latin1EBCDIC or CODEPAGE932 (ShiftJIS) to JapanEBCDICJBIS8 or UCS2NT to Latin5EBCDIC, just to name a few.

The heart of the server is in the algorithms and mapping tables used to convert UCS2NT characters to/from ClearPath host coded character set (CCS) based characters. If necessary, the server uses system APIs to map codepage data to/from UCS2NT as an intermediate step.

The ClearPath host coded character sets supported by the XlateEBCDIC server are

(ASeries)EBCDIC	JBIS8
ARABIC20EBCDIC	Latin1EBCDIC
ASUTL	Latin2EBCDIC
CanSupplEBCDIC	Latin5EBCDIC
EBCDICGB2312	Latin9EBCDIC
EBCDICKSC5601	LatinCyrillicEBC
EBCDICUTL	LatinGreekEBCDIC
GB2312	LetsJ
IBMSwedenEBCDIC	LetsJISX16
JapanEBCDICJBIS8	LocalEBCDIC
JapanV24JBIS8	

The client program instantiates a mapping object and then tailors the object to the chosen ClearPath host CCS by calling the Initialize method with the host CCS number as a parameter. This step selects the mapping tables and the mapping algorithms that are required to convert data in the ClearPath host character set to and from UCS2NT. Once the mapping object has been initialized, the methods for mapping characters that are exported by the mapping object are now available for use.

The XlateEBCDIC Server can be individually installed from the MCP server \INSTALLS share. It is also installed as part of the installation of the following products:

- Programmer's Workbench for ClearPath MCP
- Integration Expert
- Transaction Center
- Database Operations Center
- Print Center
- Enterprise Database OLE DB Data Provider for ClearPath MCP
- COMTI for Clearpath MCP
- ODBC Access – Corporation/Enterprise Edition for ClearPath MCP

Major Functions

The XlateEBCDIC server is a DLL or an in-process server. The COM server exports four COM objects:

- XlateEBCDIC
- XlateEBC
- XlateEBCDICCCS
- EnumXlateCCSes

These objects operate through four interfaces:

- IXlateEBCDIC
- IXlateEBC
- IXlateEBCDICCCS
- IEnumXlateCCSes

IXlateEBCDIC

The IXlateEBCDIC interface exposes the methods of an ActiveX control, XlateEBCDIC. The interface contains all of the methods required to support the mapping service. The XlateEBCDIC control is implemented in such a way that it can be embedded in a Visual Basic form. The control exports a list box, which displays a list of supported ClearPath CCSes and a method to return the numeric value of the selected CCS. This CCS number can be used to initialize the control. The list box can be either embedded within a form on the client program or displayed as a run-time dialog box. The control also exports methods that can be used to by the client program to identify a best fit between the workstation codepage and the ClearPath host CCS. Once the client has selected the host CCS number and initialized the control, the mapping methods are available for use.

IXlateEBC

The XlateEBCDIC mapping object, is an ActiveX control that enables visual development and supports OLE automation. The IXlateEBC interface was designed to support the same character mapping service that IXlateEBCDIC supports, but without OLE automation and the list box control. The XlateEBC objects can be hosted in both Multiple Threaded Apartments (MTAs) and Single Threaded Apartments (STAs), while XlateEBCDIC objects are restricted to STAs only. The parameters to the IXlateEBC methods are not restricted to just the data types that are allowed in OLE automation, and therefore are a more natural match to the needs of a Visual C++ client.

IXlateEBCDICCCS

The IXlateEBCDICCCS interface packages the capabilities of the XlateEBC objects so that a client that wishes to support multiple mapping streams does not have to manage a separate mapping object for each host CCS.

IEnumXlateCCSes

The IEnumXlateCCSes interface gives access to the ClearPath CCS names and their numeric value. This interface is intended for use by client programs that wish to create their own list box controls to allow the user to graphically select the proper mapping.

Mapping Object Initialization

The XlateEBCDIC server consists of a set of global procedures that contain and manage the mapping algorithms and a set of COM classes that export the mapping objects. An exported mapping object is not initialized until the host CCS has been specified. When this is done, the required mapping tables are placed in global memory. The host CCS number is then used to select the mapping algorithm that matches the mapping table and the character string.

The Best Fit Table

The XlateEBCDIC server has a “best fit” table that identifies a ClearPath NX/LX host CCS that most closely matches the codepage on the workstation. The Microsoft allocated number is the codepage number allocated by Microsoft to identify the Unisys character sets. The best-fit mapping is as follows:

Table 16-1. Best Fit Mapping to a Host CCS

Workstation Codepage Number	Best Fit host CCS	Microsoft Allocated Number
Codepage 437, 850, 1252	Latin1EBCDIC	Codepage 23500
Codepage 857, 1254	Latin5EBCDIC	Codepage 23026
Codepage 851, 1253	LatinGreekEBCDIC	Codepage 23423
Codepage 852, 1250	Latin2EBCDIC	Codepage 23870
Codepage 866, 1251	LatinCyrillicEBC	Codepage 23880
Codepage 932	JapanEBCDICJBIS8	Codepage 23290
Codepage 949	EBCDICKSC5601	Codepage 23933
Codepage 936	EBCDICGB2312	Codepage 23935
Codepage 950	EBCDICUTL	Codepage 23937

Error Reporting

The XlateEBCDIC server implements the ISupportErrorInfo COM interface and creates ErrorInfo objects whenever one of the exported methods receives an error.

The XlateEBCDIC server's interface methods return the following COM defined HRESULT values:

Table 16-2. HRESULT Values

Name	HRESULT value	Explanation
S_OK	(0x00000000L)	Is not considered a COM error.
S_FALSE	(0x00000001L)	Is not considered a COM error.
STG_E_PATHNOTFOUND	(0x80030003L)	<p>Returned when the control is unable to discover the path to the directory where its mapping tables are stored.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error #10: Unable to find the path to the XlateEBCDIC.DLL</p>
STG_E_FILENOTFOUND	(0x80030002L)	<p>Returned when the control is unable to find the file containing a required mapping table.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error #11: The Mapping File, <file path and name>, was not found</p>

Table 16-2. HRESULT Values

Name	HRESULT value	Explanation
STG_E_INSUFFICIENTMEMORY	(0x80030008L)	<p>Returned when the control fails to bring the mapping table into memory.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error #12: There is not enough memory available to load the Mapping File</p>
E_NOTIMPL	(0x80004001L)	<p>Returned if the type of mapping algorithms required for the mapping is unknown.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error #13: The Mapping Type is undefined</p>
E_NOTIMPL	(0x80004001L)	<p>Returned if one of the mapping methods is called before the control has been initialized.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error #14: The Mapping Object has not been initialized</p>

Table 16-2. HRESULT Values

Name	HRESULT value	Explanation
E_UNEXPECTED	(0x8000FFFFL)	<p>Returned if the IXlateEBC Initialize method is called more than once.</p> <p>The IXlateEBCDIC interface allows the Initialize method to be called more than once, the IXlateEBC interface does not.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error #15: The Mapping Object has already been initialized</p>
E_FAIL	(0x80004005L)	<p>Returned, if the Visual C++ compiler has failed to execute a switch statement correctly.</p> <p>The ErrorInfo object contains the string XlateEBCDIC error: Unknown Error</p>

Note: The COM system may return other error values as well

The mapping methods also have a result parameter, which can have the following values:

Table 16-3. Mapping Methods Results

Name	Result Value
XE_AOKV	0
XE_DEST_ERRV	1
XE_SOURCE_ERRV	2
XE_MAP_ERRV	13
XE_UNINITIALIZED_ERRV	14

If the result value is XE_AOKV, the HRESULT will be S_OK. If the result value is either XE_DEST_ERRV or XE_SOURCE_ERRV, the HRESULT will be S_FALSE. In either of these cases, the mapping was terminated prematurely because of the indicated error. The "s_bytes" and "d_bytes" parameters on the method give more information in these cases. If the result value is XE_MAP_ERRV, the HRESULT will be E_NOTIMPL, which is a COM error, and is usually caused by failing to initialize the mapping object before calling one of the mapping methods.

User Interfaces

The XlateEBCDIC server exports four COM objects:

- XlateEBCDIC,
- XlateEBC,
- XlateEBCDICCCS,
- EnumXlateCCSes

These objects operate through four interfaces:

- IXlateEBCDIC,
- IXlateEBC,
- IXlateEBCDICCCS,
- EnumXlateCCSes.

The methods in the following sections are presented using interface definition language (IDL) syntax. The syntax is very similar to C++ with hints added to describe the direction of data traffic.

IXlateEBCDIC Interface

Overview

The mapping object exported by the IXlateEBCDIC interface handles the mapping of characters of a single host CCS. The interface is designed for the visual development environment of Visual Basic. The mapping object hosts a list box control that can be embedded on a form at program design time.

The object must be configured, by calling Initialize with the host CCS number, before the object's mapping methods can be used. The character strings passed to the mapping methods do not have to be null terminated. The lengths of the strings are specified by parameters to the methods. Because the character size can vary within a single data string, the values of the data length parameters are always given in bytes.

Methods of the IXlateEBCDIC Interface

IXlateEBCDIC has the following methods:

Table 16-4. Methods of the IXlateEBCDIC Interface

Methods ...	Are...
To identify the host CCS number	GetCPNbrAndHostCCSNbr GetHostCCSNbrFromCPNbr SelectHostCCS
To initialize the mapping object	Initialize
For mapping data	XlateMsgFromHost XlateCPMsgFromHost XlateMsgToHost XlateCPMsgToHost

GetCPNbrAndHostCCSNbr

HRESULT GetCPNbrAndHostCCSNbr ([out] long * cp_nbr, [out] long * h_ccs_nbr)

This method uses system APIs to discover the codepage in use on the workstation and then does a best fit match for the ClearPath NX/LX host CCS. The best fit match was defined in Section 5.4. The method returns the local codepage in the cp_nbr parameter and the host CCS number in the h_ccs_nbr parameter.

If the codepage is not recognized, (ASeries)EBCDIC is returned as the host CCS number. The method returns HRESULT S_OK.

GetHostCCSNbrFromCPNbr

HRESULT GetHostCCSNbrFromCPNbr ([in, out] long * nbr)

This method accepts the codepage number from the caller and returns the best-fit match for the ClearPath NX/LX host CCS number. The client assigns the local codepage number to the nbr parameter and the method returns the host CCS number in the nbr parameter.

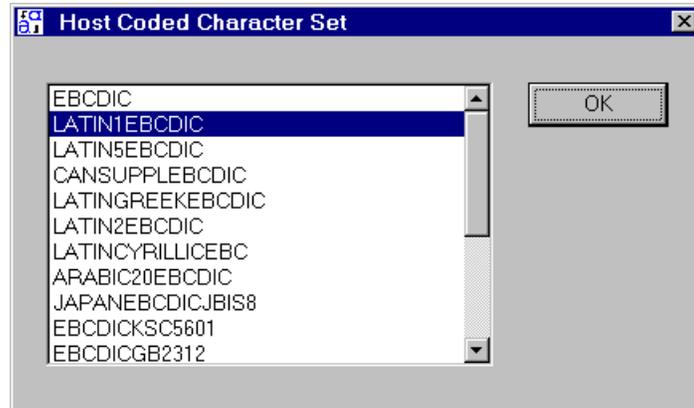
The method will accept a Microsoft codepage value for a ClearPath NX/LX host CCS and return the NX/LX host CCS value.

If the method does not recognize the codepage, (ASeries)EBCDIC is returned as the host CCS number. The method returns HRESULT S_OK.

SelectHostCCS

HRESULT SelectHostCCS ([out, retval] long * h_ccs_nbr)

This method returns the host CCS number selected by the user from either the list box embedded in the client program's form or from a dialog box displayed by the control. The best fit ClearPath NX/LX host CCS for the client's codepage is highlighted when the list is initially displayed. The selected host CCS's number is returned in the h_ccs_nbr parameter. The method returns HRESULT S_OK.



The figure above is an example of the dialog box that is displayed when the control is not embedded within a form.

Initialize

HRESULT Initialize ([in] long h_ccs_nbr)

This method configures the mapping object, selecting its mapping tables and, if necessary, bringing the tables into memory. The client assigns the host CCS number to the h_ccs_nbr parameter. The h_ccs_nbr can be one of the Microsoft codepage values that were allocated to represent ClearPath NX/LS host CCSes or a ClearPath NX/LS assigned host CCS number.

The Initialize method can be called a second (or higher) time, if the mapping object is to be re-configured for a different host CCS. If a mapping does not exist for the host CCS, the mapping for ASeries(EBCDIC) is used instead. The method returns HRESULT values S_OK, STG_E_PATHNOTFOUND, and STG_E_FILENOTFOUND.

If Initialize has not been called, the mapping methods will return an HRESULT of E_NOTIMPL and a result parameter value of XE_UNINITIALIZED_ERRV.

XlateEBCDIC error # 14: The Mapping Object has not been initialized

XlateMsgFromHost

```
HRESULT XlateMsgFromHost ([in] const BSTR * source, [in, out] long *  
s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes, [out, retval] long  
* rslt)
```

This method takes the host CCS data in source and maps it into UCS2NT encoding in dest. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV, and the HRESULT will be S_FALSE.

XlateCPMsgFromHost

```
HRESULT XlateCPMsgFromHost ([in] const BSTR * source, [in, out] long *  
s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes, [out, retval] long  
* rslt)
```

This method takes the host CCS data in source and maps it into the codepage encoding in dest. This method executes its task by first using the mapping tables to convert the host CCS data in source to UCS2NT and then calls the API WideCharToMultiByte to convert the data to the codepage encoding. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

XlateMsgToHost

```
HRESULT XlateMsgToHost ([in] const BSTR * source, [in, out] long * s_bytes,
[in, out] BSTR * dest, [in, out] long * d_bytes, [out, retval] long * rslt)
```

This method takes the UCS2NT data in source and maps it into the host CCS encoding in dest. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

XlateCPMsgToHost

```
HRESULT XlateCPMsgToHost ([in] const BSTR * source, [in, out] long *
s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes, [out, retval] long
* rslt)
```

This method takes the codepage character set data in source and maps it into the host CCS encoding in dest. This method executes its task by first calling the API MultiByteToWideChar to convert the data from the codepage encoding to UCS2NT and then uses the mapping tables to convert the UCS2NT to the host CCS encoding. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

Interface IXlateEBC

Overview

The mapping object exported by the IXlateEBC interface contains all of the methods found in IXlateEBCDIC, except for the SelectHostCCS method. The IXlateEBC interface is designed for Visual C++ client programs. The interface object must run directly in the process space of the client. There is no “marshaling” code, or overhead, to transport the data from one environment to another.

Methods

IXlateEBC has the following methods:

Methods ...	Are...
To identify the host CCS number	GetCPNbrAndHostCCSNbr GetHostCCSNbrFromCPNbr
To initialize the mapping object	Initialize
For mapping data	XlateMsgFromHost XlateCPMsgFromHost XlateMsgToHost XlateCPMsgToHost

Methods Similar to IXlateEBCDIC's

All of the methods to identify the host CCS number and to manage the debug_opt property are the same as those exported by IXlateEBCDIC and those described above. The Initialize method differs from IXlateEBCDIC's version in that calling Initialize more than once on a XlateEBC object is an error. If this is done, the method returns the HRESULT, E_UNEXPECTED, and the ErrorInfo object contains the string

XlateEBCDIC error # 15: The Mapping Object has already been initialized

The mapping methods behave exactly the same as those of IXlateEBCDIC, but they have different parameter types.

XlateMsgFromHost

```
HRESULT XlateMsgFromHost ([in] const unsigned char * source, [in, out] long * s_bytes, [in, out] wchar_t * dest, [in, out] long * d_bytes, [out, retval] long * rslt)
```

This method takes the host CCS data in source and maps it into UCS2NT encoding in dest. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the

array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

XlateCPMsgFromHost

```
HRESULT XlateCPMsgFromHost ([in] const unsigned char * source, [in, out]
    long * s_bytes, [in, out] unsigned char * dest, [in, out] long * d_bytes,
    [out, retval] long * rslt)
```

This method takes the host CCS data in source and maps it into the codepage encoding in dest. This method executes its task by first using the mapping tables to convert the host CCS data in source to UCS2NT and then calls the API WideCharToMultiByte to convert the data to the codepage encoding. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

XlateMsgToHost

```
HRESULT XlateMsgToHost ([in] const wchar_t * source, [in, out] long *
    s_bytes, [in, out] unsigned char * dest, [in, out] long * d_bytes, [out,
    retval] long * rslt)
```

This method takes the UCS2NT data in source and maps it into the host CCS encoding in dest. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

XlateCPMsgToHost

```
HRESULT XlateCPMsgToHost ([in] const unsigned char * source, [in, out] long * s_bytes, [in, out] unsigned char * dest, [in, out] long * d_bytes, [out, retval] long * rslt)
```

This method takes the codepage character set data in source and maps it into the host CCS encoding in dest. This method executes its task by first calling the API MultiByteToWideChar to convert the data from the codepage encoding to UCS2NT and then uses the mapping tables to convert the UCS2NT to the host CCS encoding. The parameter s_bytes, on input, is the number of bytes (not characters) of data in the array source. On output, it is the number of source bytes left. If the value is not zero, the rslt parameter will indicate an error. The parameter d_bytes (again bytes, not characters), on input, is the size of the array dest. The mapping stops when s_bytes are consumed, d_bytes are filled, there is no longer enough room for a complete character, or source has an invalid character (such as ending with half of a double byte character). When the mapping is completed, d_bytes contains the number of bytes mapped into the array dest. If s_bytes are completely consumed without error, the HRESULT is S_OK and the rslt value is XE_AOKV. If there was an error in the source, s_bytes will be non-zero, the rslt value will be XE_SOURCE_ERRV, and the HRESULT will be S_FALSE. If there was not enough room in dest, s_bytes will be non-zero, the rslt value will be XE_DEST_ERRV and the HRESULT will be S_FALSE.

Interface IXlateEBCDICCS

Overview

The mapping object exported by the IXlateEBCDICCS interface manages an array of XlateEBC mapping objects. Instead of calling an Initialize method once with the host CCS number, the host CCS number is a parameter in each of the mapping object's methods. The host CCS number can be either the standard ClearPath NX/LX CCS value or the Microsoft codepage value for the NX/LX CCS. The host CCS number is used as an index into the array of tailored XlateEBC mapping objects. The XlateEBCDICCS object handles all of the housekeeping associated with allocating and configuring the tailored XlateEBC mapping objects.

IXlateEBCDICCS has the following methods for mapping data:

- XlateMsgFromHostCCS
- XlateCPMsgFromHostCCS
- XlateMsgToHostCCS
- XlateCPMsgToHostCCS

XlateMsgFromHostCCS

```
HRESULT XlateMsgFromHostCCS ([in] long HostCCS, [in] const BSTR * source,
[in, out] long * s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes,
[out, retval] long * rslt)
```

This method uses the HostCCS parameter to get an object reference from the array of mapping objects. If necessary, the mapping object associated with the HostCCS number is allocated and initialized. The reference to the mapping object is then used to call its XlateMsgFromHost method to do the work.

XlateCPMsgFromHostCCS

```
HRESULT XlateCPMsgFromHostCCS ([in] long HostCCS, [in] const BSTR * source,
[in, out] long * s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes,
[out, retval] long * rslt)
```

This method uses the HostCCS parameter to get an object reference from the array of mapping objects. If necessary, the mapping object associated with the HostCCS number is allocated and initialized. The reference to the mapping object is then used to call its XlateCPMsgFromHost method to do the work.

XlateMsgToHostCCS

```
HRESULT XlateMsgToHostCCS ([in] long HostCCS, [in] const BSTR * source, [in, out] long * s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes, [out, retval] long * rslt)
```

This method uses the HostCCS parameter to get an object reference from the array of mapping objects. If necessary, the mapping object associated with the HostCCS number is allocated and initialized. The reference to the mapping object is then used to call its XlateMsgToHost method to do the work.

XlateCPMsgToHostCCS

```
HRESULT XlateCPMsgToHostCCS ([in] long HostCCS, [in] const BSTR * source, [in, out] long * s_bytes, [in, out] BSTR * dest, [in, out] long * d_bytes, [out, retval] long * rslt)
```

This method uses the HostCCS parameter to get an object reference from the array of mapping objects. If necessary, the mapping object associated with the HostCCS number is allocated and initialized. The reference to the mapping object is then used to call its XlateCPMsgToHost method to do the work.

Interface IEnumXlateCCSes

Overview

The object exported by the IEnumXlateCCSes interface gives access to the names and numbers of the host CCSes supported by the XlateEBCDIC server. With this information it is easy for a program to create a dialog with a list box that allows the user to select the host CCS, and then use its number in the call on the Initialize member function of an XlateEBC or XlateEBCDIC object.

The IEnumXlateCCSes interface exposes standard enumerator methods. IEnumXlateCCSes has the following methods:

- Next
- Skip
- Reset
- Clone

Next

```
HRESULT Next([in] ULONG celt, [out, size_is(celt), length_is(*pCeltFetched)]
             CCSes * rgCCSes, [out] ULONG * pCeltFetched)
```

This method returns the structure pointed at by rgCCSes filled with CCS name and number pairs. The structure is defined in the IDL language as

```
typedef struct structCCSes
{
    BSTR    h_ccs_name;
    long    h_ccs_nbr;
} CCSes;
```

The client is responsible for allocating the space for the CCSes structure. The celt parameter is the number of CCS name and number pairs that space has been allocated for. The server updates the pCeltFetched parameter. The value returned is number of name and number pairs placed in the space pointed at by the rgCCSes parameter. The server allocates the space for the BSTR names and the client is responsible for deleting the space when it has finished with it.

If the requested number of CCS name and number pairs is less than the total in the enumeration, the object's internal enumeration index is left pointing at the next pair in the enumeration. Subsequent Next calls will start where the previous call left off. If fewer CCS name and number pairs are returned than were asked for, the HRESULT will be S_FAIL; otherwise, the method returns S_OK.

Skip

```
HRESULT Skip([in] ULONG celt)
```

This method moves the object's enumeration index, skipping over celt CCS name and number pairs. The HRESULT is always S_OK.

Reset

```
HRESULT Reset(void)
```

This method moves the object's enumeration index back to the beginning of the CCS name and number pairs. The HRESULT is always S_OK.

Clone

```
HRESULT Clone([out] IEnumXlateCCSes ** ppEnum)
```

This method creates a duplicate enumeration object with the enumeration index pointing at the same place in the CCS name and number pairs sequence as the original. The HRESULT is S_OK, unless there is a COM error in the creation of the new object.

The XlateEBCDIC Type Library

Overview

The MIDL compiler creates the XlateEBCDIC type library. The type library is contained in the XlateEBCDIC.DLL code file and is added to the registry when the XlateEBCDIC server is registered. It contains, in a programming language neutral form, definitions of all of the exported interfaces, methods, and properties. The type library also contains the type definition of the enumeration structure used in the IEnumXlateCCSes interface and the enumerated constants used by the server.

Enumeration Structure

The IEnumXlateCCSes interface exports a method that returns the CCS name and value of all of the CCSes supported by the XlateEBCDIC server. The enumeration structure is defined as follows:

```
typedef struct structCCSes
{
    BSTR    h_ccs_name;
    long    h_ccs_nbr;
} CCSes;
```

Enumerated Constants

The enumerated constants for the ClearPath NX/LX host CCS names and numbers are as follows:

Table 16-5. Enumerated Constants for CCS Names and Numbers

CCS Name	CCS Numbers
ARABIC20EBCDIC	34
ASCII	5
ASKSC	902
ASUTL	82
CANSUPPLEBCDIC	16
EBCDIC	4
EBCDICGB2312	111
EBCDICKSC5601	105
EBCDICUTL	109
GB2312	935
IBMSwedenEBCDIC	51
JAPANEBCDICJBIS8	100
JAPANV24JBIS8	114

Table 16-5. Enumerated Constants for CCS Names and Numbers

CCS Name	CCS Numbers
JBIS8	80
LATIN1EBCDIC	12
LATIN2EBCDIC	26
LATIN5EBCDIC	14
LATIN9EBCDIC	47
LATINCYRILLICEBC	29
LATINGREEKEBCDIC	19
LETSJ	104
LETSJISX16	930
LocalEBCDIC	50
UNKNOWN	0

The enumerated constants for the Microsoft codepage values for the ClearPath NX/LX host CCSes are as follows:

Table 16-6. Microsoft Codepage Values for a HostCCS

Host CCS	Value
CP_ARABIC20EBCDIC	23420
CP_CANSUPPLEEBCDIC	23037
CP_EBCDIC	2300
CP_EBCDICGB2312	23935
CP_EBCDICKSC5601	23933
CP_EBCDICUTL	23937
CP_FRENCHARABICE	23421
CP_JAPANEBCDIC	23290
CP_JAPANEBCDICJBIS8	23291
CP_LATIN1EBCDIC	23500
CP_LATIN2EBCDIC	23870
CP_LATIN5EBCDIC	23026
CP_LATINCYRILLICEBC	23880
CP_LATINGREEKEBCDIC	23423

The values for the rslt parameter are as follows:

Table 16-7. Values for the RSLT Parameter

Rslt Parameter	Value
XE_DISASTERV	-1
XE_AOKV	0
XE_DEST_ERRV	1
XE_SOURCE_ERRV	2
XE_PATH_ERRV	10
XE_FILE_ERRV	11
XE_MEM_ERRV	12
XE_MAP_ERRV	13
XE_UNINITIALIZED_ERRV	14
XE_ALREADY_INITIALIZED_ERRV	15

Appendix A

CENTRALSUPPORT Library Messages

This appendix describes the CENTRALSUPPORT library messages. The CENTRALSUPPORT library messages are of two different types:

- Result messages
These messages are returned to an application program when that application program uses the procedures in the CENTRALSUPPORT library.
- CENTRALSUPPORT initialization messages
These messages are displayed during library initialization. They are never returned to a library caller.

Result Messages

The result of a procedural call is coded as an integer value in the program. The text associated with the result value is called the *result message*. The text for each result message is the same as the text retrieved when a program calls the GET_CS_MSG procedure. Result messages can have the following three parts:

- A header message, which has a maximum of 80 characters
- A general description, which has a maximum of 80 characters
- A specific description, which has no maximum length

All parts of the result message are returned to the application program. However, you can code your program to display or use only those parts of the message that you need.

Header Messages

If a message has a header, the header is the first 80 characters of the message. Header messages identify the category of a message that is returned to an application program when that application program uses the procedures in the CENTRALSUPPORT library. Messages in the same category are usually caused by the same problem. Four header messages identify the five categories of CENTRALSUPPORT library messages. Every message is in one of the five categories. Therefore, all the messages listed in this appendix have one of the four header messages as the first 80 characters. One header message is used with two of the categories.

The following paragraphs list the header messages and explain the category of each message.

Category 0 CENTRALSUPPORT REQUEST OK (# <message number>)

The range 0 through 999 includes category 0 result messages. These messages indicate that the procedural call successfully completed.

Category 1 CENTRALSUPPORT SOFTWARE ERROR (# <message number>)

The range 1000 through 1999 includes category 1 result messages. These messages indicate that an internal error occurred in the CENTRALSUPPORT library.

Category 2 CENTRALSUPPORT INTERFACE ERROR (# <message number>)

The range 2000 through 2999 includes category 2 result messages. These messages indicate that the caller passed invalid data to the CENTRALSUPPORT procedure. The CENTRALSUPPORT procedure completed the request and attempted to return valid data.

Category 2 messages are nonfatal, informative messages. The system returns to the caller as much data as it can generate with the given error condition.

Category 3 CENTRALSUPPORT INTERFACE ERROR (# <message number>)

The range 3000 through 3999 includes category 3 result messages. These messages indicate that the caller passed an invalid parameter or invalid data to the CENTRALSUPPORT procedure. The calling procedure cannot return valid data until the invalid parameter or data is corrected.

**Category 4 CENTRALSUPPORT DATA ERROR
(# <message number>)**

The range 4000 through 4999 includes category 4 result messages. These messages indicate that the calling procedure passed data for which the called CENTRALSUPPORT procedure could not find information to return. The CENTRALSUPPORT procedure completed the request, but no data was returned.

Result Messages by Category

The following text lists result messages according to their category. Following each message is the cause of the message. After the cause, a suggested response that might help you to correct the situation is presented.

Category 0 Error Messages

These messages indicate that the call to the procedure was successfully completed.

0 REQUEST PROCESSED **Your request has been processed.**

Cause...	Suggested Response...
The procedure completed successfully. For those procedures that return a TRUE or FALSE condition, the result of the procedure is FALSE.	No action is needed.

1 REQUEST PROCESSED **Your request has been processed.**

Cause...	Suggested Response...
The procedure completed successfully. For those procedures that return a TRUE or FALSE condition, the result of the procedure is TRUE.	No action is needed.

Category 1 Error Messages

These messages indicate that internal errors occurred in the CENTRALSUPPORT library.

Program Dumps

If the CENTRALSUPPORT library returns error message 1001 or 1002, a program dump generally occurs. A maximum of 10 program dumps are allowed for each execution of the library. If a program dump does not occur, then the limit has been reached and you should reset the count of program dumps to zero.

To reset the counter, call the procedure RESET_PDUMP_COUNT. The following ALGOL code shows how to call this procedure.

```
BEGIN
  LIBRARY CENTRALSUPPORT (LIBACCESS=BYFUNCTION);
  PROCEDURE RESET_PDUMP_COUNT;
    LIBRARY CENTRALSUPPORT;
  RESET_PDUMP_COUNT;
END.
```

1000 UNABLE TO ACCESS CCSFILE OR CONVENTIONS FILE
An error occurred while accessing the SYSTEM/CCSFILE or the
SYSTEM/CONVENTIONS file.

Cause...	Suggested Response...
<p>An error occurred while the program was accessing the CCSFILE or the CONVENTIONS file.</p>	<p>If you do not know whether the coded character set, the ccsversion file, or the conventions file caused this error, you must examine both the SYSTEM/CCSFILE and the SYSTEM/CONVENTIONS files.</p> <p>Verify that the file that caused the error has the same or earlier Mark release number as the CENTRALSUPPORT library. If the level of the file is correct, you might need to file a User Communication Form (UCF).</p> <p>Because the CCSFILE and CONVENTIONS files are optional, your program can run without them. Without either file, only the ASeriesNative ccsversion is available on your system.</p> <p>You can run your program without the CCSFILE and CONVENTIONS files by doing the following steps:</p> <ol style="list-style-type: none"> 1. Remove the file or change the name of the file to another name. 2. Use the SYSOPS system command to change the system defined ccsversion or convention to ASeriesNative, depending on the file that was removed. 3. Examine your task and user settings to verify that they access ASeriesNative. The verification is necessary because other ccsversion and conventions are no longer available. 4. Perform a halt/load on your system so the CENTRALSUPPORT library initializes without the files.

1001 UNABLE TO PROCESS REQUEST: CENTRALSUPPORT FAULT
An unexpected fault occurred in CENTRALSUPPORT. Your request cannot be processed at this time.

Cause...	Suggested Response...
An unexpected fault occurred during procedure execution and a program dump of the CENTRALSUPPORT library might occur. This error might occur if array parameters passed to the procedure are not of the required length.	Verify that all the array parameters are large enough, that the size parameters are correct, and that the parameters do not exceed the size of the arrays.

1002 UNABLE TO PROCESS REQUEST: CENTRALSUPPORT SOFTWARE ERROR
A CENTRALSUPPORT software error was detected. Your request cannot be processed at this time.

Cause...	Suggested Response...
A software error was detected and a program dump of the CENTRALSUPPORT library might occur. This error occurred because of a mistake in the procedure implementation.	File a UCF.

Category 2 Error Messages

These messages identify errors that indicate the passing of invalid data to a CENTRALSUPPORT procedure; however, the CENTRALSUPPORT library was able to return some valid data to the caller. Category 2 messages are nonfatal.

2001 REQUEST PROCESSED IN SUBSTITUTE LANGUAGE
The data is not in the requested language. It is in MYSELF.LANGUAGE or the SYSTEM LANGUAGE or the first available LANGUAGE.

Cause...	Suggested Response...
The requested information is available from the CENTRALSUPPORT library, but the data was not translated to the requested language.	See your system administrator to have the requested information in the CENTRALSUPPORT library translated to your requested language. Refer to Section 6, "Creating Application Systems in an MLS Environment," for information on support for natural languages.

2002 REQUEST PROCESSED IN SUBSTITUTE CONVENTION

The data is not in the requested convention; it is in MYSELF.CONVENTION or the SYSTEM CONVENTION.

Cause...	Suggested Response...
The requested convention is not available on your system.	You can use the SYSOPS ALL system command to view the available conventions on your system. If the requested convention is not available but is one of the conventions provided on the release media, or if the convention is customized, see your system administrator to add the convention to your system.

2004 REQUEST PROCESSED: ONLY PARTIAL DATA AVAILABLE

Only partial data is being returned. There was insufficient space in the output array.

Cause...	Suggested Response...
The array passed by the caller to hold the output is too small. Some valid data was returned, but the data is incomplete.	Increase the size of the array that is passed to the procedure.

2005 REQUEST PROCESSED: INCOMPLETE DATA RETURNED

Incomplete data is being returned for a multibyte stream.

Cause...	Suggested Response...
An incomplete multibyte source character was returned, such as half of a double-byte character pair, an unbalanced single bracket, or a single shift character not followed by the data character.	Increase the size of the array that is passed to the procedure.

Category 3 Error Messages

These messages include errors that indicate the passage of invalid data to a CENTRALSUPPORT procedure. The CENTRALSUPPORT library was unable to return any valid data.

3000 UNABLE TO PROCESS REQUEST: CHECK INTERFACE TO CENTRALSUPPORT

A parameter was incorrectly specified as less than or equal to 0.

Cause...	Suggested Response...
A parameter was incorrectly specified as less than or equal to 0.	Change the value of the parameter so it is greater than 0.

3001 UNABLE TO PROCESS REQUEST: CHECK INTERFACE TO CENTRALSUPPORT

The output array size is smaller than the length of the data it is supposed to contain.

Cause...	Suggested Response...
The output array size is smaller than the length of the data it is supposed to contain.	Increase the size of the array that is passed to the procedure.

3002 UNABLE TO PROCESS REQUEST: CHECK INTERFACE TO CENTRALSUPPORT

At least one array length is invalid or the offset + length is greater than the size of the array.

Cause...	Suggested Response...
The length is not valid or the offset plus the length is greater than the size of at least one array.	Verify that the sizes of all arrays are compatible with their parameters.

3003 INVALID CHARACTER SET OR CCSVERSION NUMBER
The requested number was not found.

Cause...	Suggested Response...
The requested number for the coded character set or ccsversion is not available on your system.	Use the SYSOPS ALL system command to see the ccsversions and coded character sets that are available on your system. Compare this information to the list provided in Section 9, "Coded Character Sets and Coded Character Set Mapping Tables." Your system might contain a subset of the available ccsversions and coded character sets. If your request is not available, you can use the ASeriesNative ccsversion because the ASeriesNative ccsversion is available on all systems

3004 INVALID CHARACTER SET OR CCSVERSION NAME
The requested name was not found.

Cause...	Suggested Response...
The requested name for the coded character set or ccsversion is not available on your system.	Use the SYSOPS ALL system command to see the ccsversions and coded character sets that are available on your system. Compare this information to the list provided in Section 9, "Coded Character Sets and Coded Character Set Mapping Tables." Your system might contain a subset of the available ccsversions and coded character sets. If your request is not available, you can use the ASeriesNative ccsversion because the ASeriesNative ccsversion is available on all systems.

3005 INVALID MESSAGE NUMBER
The requested number was not found.

Cause...	Suggested Response...
The requested message number is not known to the CENTRALSUPPORT library.	Use this appendix to determine the message that is associated with a message number.

3006 UNABLE TO PROCESS REQUEST: CHECK INTERFACE TO CENTRALSUPPORT

The type code specified is out of the acceptable range.

Cause...	Suggested Response...
The type code specified is not available for this procedure.	Check the appropriate programming language reference manual for a description of the valid type codes for each procedure.

3007 UNABLE TO PROCESS REQUEST

The flag specified is out of the acceptable range.

Cause...	Suggested Response...
The specified flag is not available for this procedure.	Check the appropriate programming language reference manual for a description of the valid flags for each procedure.

3008 UNABLE TO PROCESS COMPARE REQUEST

The space for OSVS or total storage allocated in OUTPUT is not big enough for OSVs and/or PSVs.

Cause...	Suggested Response...
The output array is too small. The space allocated for OSVs, or the total space allocated for both OSVs and PSVs, is not large enough to hold the output.	<ul style="list-style-type: none"> • Increase the size of the output array. • Check the complete description of the parameters for the procedure being used.

3009 INVALID PARAMETER VALUE

The "AMT" parameter value is invalid.

Cause...	Suggested Response...
The parameter value is not an integer or it is greater than the maximum double-precision integer.	Verify the numeric input.

3011 UNABLE TO PROCESS REQUEST: BAD TEMPLATE CHARACTER
An invalid control character was detected in the template.

Cause...	Suggested Response...
An invalid control character was detected in the template. exclamation marks (!) must surround control characters.	Check the list of valid control characters. Also, make sure that the exclamation marks are in the correct place. Refer to Section 1, "Understanding the MultiLingual (MLS) System," and Section 3, "Understanding Templates in Convention Definitions," for a description of the use of control characters.

3012 INVALID CHARACTERS IN DATE
The input DATE contains illegal characters.

Cause...	Suggested Response...
The DATE value contains characters other than digits or the separator character defined for the convention.	Ensure that the date contains digits. Verify that you are using the correct separator character for the date by checking the appropriate convention definition in Section 14, "Standard Conventions."

3013 INVALID CHARACTERS IN TIME
The input TIME contains illegal characters.

Cause...	Suggested Response...
The TIME value contains characters other than digits or the separator character defined for the convention.	Ensure that the time contains digits. Verify that you are using the correct separator character for the time by checking the appropriate convention definition in Section 14, "Standard Conventions."

3014 UNABLE TO ADD CONVENTION: CONVENTION EXISTS
An attempt was made to add a new convention with the name of an existing convention.

Cause...	Suggested Response...
A new convention that has the same name as an existing convention cannot be added.	If the convention is customized, you can modify the existing convention. Also, you can choose a new convention name or delete the custom convention that you do not want.

**3015 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The maximum digits value is either missing or out of range.**

Cause...	Suggested Response...
Either no maximum value for digits exists or the maximum value is out of range.	Add a value to the Maximum Digits field of the input array.

**3016 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The fractional digits value is either missing or out of range.**

Cause...	Suggested Response...
Either no fractional digits value exists or the fractional digits value is out of range.	Add a value to the Fractional Digits field of the input array.

**3017 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The international fractional digits value is either missing or out of range.**

Cause...	Suggested Response...
Either no value for the international fractional digits exists or the international fractional digits value is out of range.	Add a value to the International Fractional Digits field of the array.

**3018 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The long date template is either missing or contains invalid information.**

Cause...	Suggested Response...
The template contains control characters that are not valid for the long date template, or the long date template is missing.	Add the long date template to the input array. If the long date template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

**3019 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The short date template is either missing or it contains invalid information.**

Cause...	Suggested Response...
The template contains control characters that are not valid for the short date template, or the short date template is missing.	Add the short date template to the input array. If the short date template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

3020 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The numeric date template is either missing or contains invalid information.

Cause...	Suggested Response...
The template contains control characters that are not valid for the numeric date template, or the numeric date template is missing.	Add the numeric date template to the input array. If the numeric date template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

3021 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The long time template is either missing or it contains invalid information.

Cause...	Suggested Response...
The template contains control characters that are not valid for the long time template, or the long time template is missing.	Add the long time template to the input array. If the long time template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

3022 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The numeric time template is either missing or contains invalid information.

Cause...	Suggested Response...
The template contains control characters that are not valid for the numeric time template, or the numeric time template is missing.	Add the numeric time template to the input array. If the numeric time template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

3023 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The monetary template is either missing or it contains invalid information.

Cause...	Suggested Response...
The template contains control characters that are not valid for the monetary template, or the monetary template is missing.	Add the monetary template to the input array. If the monetary template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

**3024 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The numeric template is either missing or it contains invalid information.**

Cause...	Suggested Response...
The template contains control characters that are not valid for the numeric template, or the numeric template is missing.	Add the numeric template to the input array. If the numeric template exists, refer to Section 14, "Standard Conventions," to verify that the characters in the template are valid.

**3027 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The lines per page value is either missing or it is out of range.**

Cause...	Suggested Response...
The lines per page value is invalid, or the value is missing.	Add the correct lines per page value to the input array. The value of the lines per page must be in the range from 12 to 255.

**3028 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The characters per line value is either missing or it is out of range.**

Cause...	Suggested Response...
The characters per line value is invalid, or the value is missing.	Add the correct characters per line value to the input array. The value of the characters per line must be in the range from 40 to 255.

**3029 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
A required symbol in either the monetary or the numeric template is missing.**

Cause...	Suggested Response...
All symbols must be defined before you can add or modify this convention.	Reformat the template to include the missing symbol.

**3030 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
An invalid template length value was encountered.**

Cause...	Suggested Response...
The length of the template is too short for a valid template. A template cannot have a length of 1.	Verify that the template is formed correctly.

3031 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
A mutually exclusive combination of control characters has been encountered in a monetary or numeric template.

Cause...	Suggested Response...
The numeric negative symbol control character was used with the enclosure symbols. Either a numeric negative symbol or an enclosure symbol can be used, but not both. If present, enclosure symbols are used in place of numeric negative symbols.	Delete either the numeric negative symbol or the enclosure symbol from the template.

3032 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
The mindigits field in a "t" control character in a monetary or numeric template is out of range.

Cause...	Suggested Response...
The template might be formed incorrectly. The syntax is t[T:m,g1,g2, . . . gn].	Examine the minimum digits, the following punctuation, and the group specification to verify that they are correct.

3033 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
A right bracket "]" is required to terminate a "t" control character symbol definition list.

Cause...	Suggested Response...
The template might be formed incorrectly. The syntax has the form t[T:m,g1,g2, . . . gn].	Add a right bracket (]).

3034 UNABLE TO PROCESS CONVENTION ADD OR MODIFY REQUEST
An expected colon ":" is missing from the "t" control character in a monetary or numeric template.

Cause...	Suggested Response...
The template might be formed incorrectly. The syntax has the form t[T:m,g1,g2, . . . gn].	Add a colon (:).

3035 BAD OR MISSING MONETARY OR NUMERIC INPUT VALUE

The input value did not contain digits or an expected symbol was missing.

Cause...	Suggested Response...
A numeric value was expected, but an invalid character was encountered. Additionally, a decimal symbol might be missing.	Verify the numeric input.

3036 REQUEST COMPLETED: CONVENTION NOT FOUND

Specified convention does not exist and cannot be retrieved, modified, or deleted.

Cause...	Suggested Response...
The convention is not available on the system.	Verify the spelling of the name of the convention. Use the SYSOPS system command for the list of conventions currently on the system. If only the ASeriesNative ccversion is available, the SYSTEM/CONVENTIONS file might not be in use. If so, put the SYSTEM/CONVENTIONS file under the usercode and on the pack of the CENTRALSUPPORT library. For the CENTRALSUPPORT library to initialize the SYSTEM/CONVENTIONS file, a halt/load must be performed.

3037 SYSTEM/CONVENTIONS FILE NOT PRESENT

A convention definition cannot be added, modified, or deleted.

Cause...	Suggested Response...
The SYSTEM/CONVENTIONS file is not on the system.	Put the SYSTEM/CONVENTIONS file under the usercode and on the pack of the CENTRALSUPPORT library. For the CENTRALSUPPORT library to initialize the SYSTEM/CONVENTIONS file, a halt/load must be performed.

3038 INVALID PARAMETER VALUE

The "PRECISION" parameter value is out of range.

Cause...	Suggested Response...
The parameter value is not in the range from 0 to 9.	Verify the numeric input.

3039 UNABLE TO COMPLETE REQUEST**A required convention name was not provided.**

Cause...	Suggested Response...
A convention name is required to perform the specified action.	Specify a convention name in the input parameter.

3040 UNABLE TO COMPLETE REQUEST**The named convention is a standard convention and cannot be modified or deleted.**

Cause...	Suggested Response...
Only custom conventions can be modified or deleted. The standard conventions provided on the release media cannot be modified or deleted.	Modify or delete only custom conventions.

3041 UNABLE TO COMPLETE REQUEST**A hexadecimal value representing a symbol in a monetary or numeric template is missing a required delimiter.**

Cause...	Suggested Response...
A quotation mark (") might be missing. The correct syntax is 48"<hex code(s)>".	Add the missing quotation mark.

3042 UNABLE TO COMPLETE REQUEST**An invalid character was encountered in a hex value representing a symbol in a monetary or numeric template.**

Cause...	Suggested Response...
The valid range for hexadecimal values is 00 through FF.	Change the hexadecimal value to a valid value.

3043 UNABLE TO COMPLETE REQUEST

The international currency notation is missing a required terminating delimiter.

Cause...	Suggested Response...
A quotation mark (") is missing. For example, correct syntax is "USD".	Add the missing quotation mark.

3044 INVALID SEPARATOR CHARACTERS IN DATE

The DATE components are separated by an invalid character.

Cause...	Suggested Response...
The character separating DATE components is not the separator defined for the specified convention.	Verify that you are using the correct separator character for the date by checking the appropriate convention definition in Section 15, "Standard Conventions."

3045 INVALID YEAR VALUE IN DATE

The YEAR component exceeds 2 digits.

Cause...	Suggested Response...
A value greater than 99 was specified for the YEAR component.	Ensure that the year value is in the acceptable range.

3046 INVALID YEAR VALUE IN DATE

A nonzero value is required for the YEAR component.

Cause...	Suggested Response...
Zero (0) has been specified for the YEAR component.	Ensure that the year value in date is greater than zero (0).

3047 INVALID MONTH VALUE IN DATE

The MONTH value is outside of the valid range. Acceptable values are in the range 1 through 12.

Cause...	Suggested Response...
The month value is outside of the acceptable range.	See the appropriate programming language reference manual to verify that the month value is in the acceptable range.

3048 INVALID DAY VALUE IN DATE

The DAY value is outside of the valid range. Acceptable values for months January through December are: 1..31, 1..28 (1..29 in a leap year), 1..31, 1..30, 1..31, 1..30, 1..31, 1..31, 1..30, 1..31, 1..30, and 1..31, respectively.

Cause...	Suggested Response...
The day value is outside of the acceptable range.	See the appropriate programming language reference manual to verify that the day value is in the acceptable range.

3049 A REQUIRED DATE VALUE IS MISSING

An input DATE is required but missing.

Cause...	Suggested Response...
A DATE value was expected, but zeros or blanks were encountered.	Ensure that a date value is present.

3050 INVALID SEPARATOR CHARACTERS IN TIME

TIME components are separated by an invalid character.

Cause...	Suggested Response...
The character separating TIME components is not the separator defined for the specified convention.	Verify that you are using the correct separator character for the time by checking the appropriate convention definition in Section 14, "Standard Conventions."

3051 INVALID 24-HOUR CLOCK VALUE IN TIME

The HOUR value is outside of the valid range for the 24-hour clock. Acceptable values are in the range 0 through 23.

Cause...	Suggested Response...
The 24-hour clock value is outside of the acceptable range.	See the appropriate programming language reference manual to verify that the hour value is in the acceptable range.

3052 INVALID 12-HOUR CLOCK VALUE IN TIME

The HOUR value is outside of the valid range for 12-hour clock. Acceptable values are in the range 1 through 12.

Cause...	Suggested Response...
The 12-hour clock value is outside of the acceptable range.	See the appropriate programming language reference manual to verify that the hour value is in the acceptable range.

3053 INVALID MINUTE VALUE IN TIME

The MINUTE value is outside the valid range. Acceptable values are in the range 0 through 59.

Cause...	Suggested Response...
The minute value is outside of the acceptable range.	See the appropriate programming language reference manual to verify that the minute value is in the acceptable range.

3054 INVALID SECOND VALUE IN TIME

The SECOND value is outside of the valid range. Acceptable values are in the range 0 through 59.

Cause...	Suggested Response...
The second value is outside of the acceptable range.	See the appropriate programming language reference manual to verify that the second value is in the acceptable range.

3055 INVALID CHARACTERS IN PARTIAL SECOND VALUE

The PARTIAL SECOND value contains invalid characters.

Cause...	Suggested Response...
The partial second value contains characters other than digits or blanks.	Ensure that the PARTIAL SECOND value contains either blanks or digits.

3056 A REQUIRED TIME VALUE IS MISSING

An input TIME is required but missing.

Cause...	Suggested Response...
A TIME value was expected, but blanks were encountered.	Ensure that a time value is present.

3057 A REQUIRED MONTH VALUE IS MISSING**The MONTH value is required but missing.**

Cause...	Suggested Response...
Because a nonzero DAY OF MONTH value is present, a nonzero MONTH value is expected.	Ensure that a nonzero month value is present in the date.

3058 INVALID DAY OF YEAR VALUE IN DATE**The DAY OF YEAR value is outside the valid range. Acceptable values are in the range 1 through 365 (1 through 366 for a leap year).**

Cause...	Suggested Response...
Zero (0) is not an acceptable value for the DAY OF YEAR value.	Ensure that a DAY OF YEAR value is in the acceptable range.

3059 A REQUIRED DATE COMPONENT IS MISSING**The DAY OF YEAR value cannot be calculated because a DATE component (year, month, or day) is missing.**

Cause...	Suggested Response...
At least one and possibly all of the required date components are missing.	Ensure that all three DATE components are specified to validate and compute the DAY OF YEAR value.

Category 4 Error Message

This message indicates that the caller passed data for which the CENTRALSUPPORT library could not find information to return. The CENTRALSUPPORT library completed the request, but returned no data.

4002 REQUEST COMPLETED: DATA NOT FOUND**The requested data was not found.**

Cause...	Suggested Response...
It is possible that the requested data is optional and, therefore, no data exists. If the data is not optional, the data is missing.	Refer to Section 9, "Coded Character Sets and Coded Character Set Mapping Tables," to find out if the requested data is optional.

4004 REQUEST COMPLETED: COMPLEX TRANSLATION REQUIRED
The translation must be performed by calling the
CCSTOCCS_TRANS_TEXT_COMPLEX procedure.

Cause...	Suggested Response...
The requested translation includes multibyte or 16-bit data, which cannot be processed by the current procedure.	The Complex mapping procedure is required.

CENTRALSUPPORT Initialization Messages

These messages are contained in the output message array CSINT_MSGS. The following text includes an explanation of the possible cause of the message. Messages that have obvious causes are not listed because no further explanation is needed. The initialization messages are used only inside the CENTRALSUPPORT library. The messages are never returned to a library caller and are most commonly displayed during initialization.

Many errors indicate a problem with I/O to a file. If you have no known problems with I/O on your system and the error is impeding initialization, you can remove the file. Removing the file is possible because the system can run in ASeriesNative mode without the file. You can then change the system default to ASeriesNative and perform a halt/load. Changing the system causes CENTRALSUPPORT to reinitialize without the file. Using this method to solve I/O problems causes internationalization features of the CENTRALSUPPORT library to be unavailable.

For most messages, the cause is provided in the text of the message. If a message does not contain an obvious cause, the cause is provided below the presentation of the message in the following text.

The system default CCSVERSION is invalid. Use AX <CCSVERSION name> to enter a new ccsversion or AX ASERIESNATIVE to use default settings.

Cause...	Suggested Response...
The name specified for the ccsversion is not available from the file SYSTEM/CCSFILE.	You can choose another ccsversion by using the ACCEPT(AX) command or you can use the ASeriesNative ccsversion.

The system default CCSVERSION specified requires the file SYSTEM/CCSFILE, but it is still missing.

Cause...	Suggested Response...
CENTRALSUPPORT is unable to locate the SYSTEM/CCSFILE.	You can place the SYSTEM/CCSFILE under the usercode and on the pack in which the CENTRALSUPPORT library resides.

Unable to initiate library. Ambiguous library name, conflicts with SYSTEM LIBRARY name.

Cause...	Suggested Response...
<p>The probable cause for this message is one of the following conditions:</p> <ul style="list-style-type: none"> • The library is initiated by a user. • The library is defined with a BYTITLE statement. • The library name is identical to the right-hand side of the SYSTEM LIBRARY declaration for the CENTRALSUPPORT library in the SYSTEM LIBRARY table. 	<p>Because a user library cannot be used as the SYSTEM LIBRARY, change the name of the library.</p>

<CONVENTION name> is an invalid system default CONVENTION. Use AX <CONVENTION name> to enter a new convention or AX ASERIESNATIVE to use default settings.

Cause...	Suggested Response...
<p>The name specified for the convention is not available from the file SYSTEM/CONVENTIONS.</p>	<p>You can choose another convention by using the ACCEPT(AX) command or you can use the ASeriesNative convention.</p>

AX: *SYSTEM/CCSFILE ON DISK NOT OPEN F.AVAILABLE = 2. ??COPY or CHANGE file and AX RETRY or AX SETASN to return to default setting.

Cause...	Suggested Response...
<p>For the current system default setting, the file SYSTEM/CCSFILE must be present; however, the file is missing or cannot be opened.</p>	<p>If you think that you have a corrupt file, you can use the ??COPY command to copy a new SYSTEM/CCSFILE file. Then type <mix number>AX RETRY. However, if you chose an incorrect system default or your file is corrupt or nonrecoverable, you can enter <mix number>AX SETASN to set the system default to the ASeriesNative ccsversion because the SYSTEM/CCSFILE file is not required.</p>

Appendix B

Examples of Using Libraries to Access Output Message Arrays

This appendix provides an example of an ALGOL library that contains an output message array. It also provides examples of C, COBOL74, COBOL85, Pascal, and RPG programs that access messages from this library.

Example ALGOL Library

Example B-1 shows an ALGOL library that contains an output message array called MESSAGES. The MESSAGES array contains two messages that appear in English.

The GET_FROM_MESSAGES procedure in this library searches for a message based on the message number received from the calling program. The GET_FROM_MESSAGES_ONE_PARAM procedure also searches for a message based on the number received. However, this procedure also receives a parameter value to be inserted into the message when it is sent to the calling program.

```
BEGIN
  OUTPUTMESSAGE ARRAY
  MESSAGES (
    ENGLISH
      <"This translators' help text will appear with all "
        "messages in this language, ENGLISH">
    (
      1 = <"This translators' help text will appear only with this "
        "message, message number 1">
        "First message",
      2 = <"This translators' help text will appear only with this "
        "message, message number 2">
        <1> " parameterized message"
    ));

  INTEGER PROCEDURE GET_FROM_MESSAGES (MSG_NUM,
                                        MSG,
                                        LEN);

  EBCDIC ARRAY

  MSG [*]    % The EBCDIC array to contain the output message.
;
```

Examples of Using Libraries to Access Output Message Arrays

```
INTEGER
  MSG_NUM      % The message number to select the message.
  ,LEN         % The length of the returned message.
;
BEGIN % procedure GET_FROM_MESSAGES
  POINTER
  PTR_MSG      % The pointer to the EBCDIC array to contain the
               % output message.
;
  PTR_MSG:= MSG[0];
  GET_FROM_MESSAGES:= MESSAGESEARCHER (MESSAGES [MSG_NUM],
                                       PTR_MSG,
                                       LEN);

  END; % procedure GET_FROM_MESSAGES

INTEGER PROCEDURE GET_FROM_MESSAGES_ONE_PARAM (MSG_NUM,
                                              MSG,
                                              LEN,
                                              PARAMS,
                                              PLEN);

EBCDIC ARRAY
  MSG          % The EBCDIC array to contain the output message.
  ,PARAMS [*] % The EBCDIC array containing the parameter(s).
;
INTEGER
  MSG_NUM      % The message number to select the message.
  ,LEN         % The length of the returned message.
  ,PLEN        % The length of the parameter in bytes.
;
BEGIN % procedure GET_FROM_MESSAGE_ONE_PARAM
  POINTER
  PTR_MSG      % The pointer to the EBCDIC array to contain the
               % output message.
;
  PTR_MSG:= MSG[0];
  GET_FROM_MESSAGES_ONE_PARAM:= MESSAGESEARCHER(MESSAGES [MSG_NUM],
                                               PTR_MSG,
                                               LEN,
                                               PARAMS[0] FOR PLEN);

  END; % procedure GET_FROM_MESSAGES_ONE_PARAM;

EXPORT GET_FROM_MESSAGES,
       GET_FROM_MESSAGES_ONE_PARAM;
FREEZE (TEMPORARY);
END.
```

Example C Include File

Example B-2 shows the include file used by a C program to access the messages in the example ALGOL library. This file defines all the entry points in the ALGOL library. Assume that this file is called EXAMPLE/MLS/C/LIBRARY/DECLARATION/H.

```
/* This include file provides the declarations needed to
import the procedures from the ALGOL library:
EXAMPLE/MLS/ALGOL/LIBRARY. */

extern "ALGOL" int GET_FROM_MESSAGES (int &msg_num,
                                     char (&msg) [],
                                     int &len);

extern "ALGOL" int GET_FROM_MESSAGES_ONE_PARAM (int &msg_num,
                                                char (&msg) [],
                                                int &len,
                                                char (&ea) [],
                                                int &plen);
```

Example C Program

Example B-3 shows the actual C program that accesses the messages from the ALGOL library. This C program uses the include file EXAMPLE/MLS/C/LIBRARY/DECLARATION/H shown in Example B-2.

```
#include <stdio.h>
#include <string.h>

/* Invoke C include file to declare the library entry points
   GET_FROM_MESSAGES and GET_FROM_MESSAGES_ONE_PARAM.          */

#include "example/mls/c/library/declaration.h" \
      (bytitle="object/example/mls/algol/library", \
       intname="EXAMPLE/LIB")

main() {

    char retn_msg[100];
    int  msg_result;
    int  msg_len;
    int  no_parm_msg = 1;
    int  one_parm_msg = 2;
    char parm1[]      = "ABC";
    int  parm1len     = strlen(parm1);

    msg_result = GET_FROM_MESSAGES (no_parm_msg, retn_msg, msg_len);

    /* The following statement will display: "First message" */
    printf("%s\n", retn_msg);

    msg_result = GET_FROM_MESSAGES_ONE_PARAM(one_parm_msg, retn_msg,
                                             msg_len, parm1, parm1len);

    /* The following statement will display:
       "ABC parameterized message"          */
    printf("%s\n", retn_msg);

}
```

Example COBOL74 Program

Example B-4 shows a COBOL74 program that accesses the messages from the example ALGOL library.

```
IDENTIFICATION DIVISION.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 MSG-NUM    PIC 9(11) BINARY.
01 MSG        PIC X(150) WITH LOWER-BOUNDS.
77 MSG-LEN    PIC 9(11) BINARY.
01 PARAM-STR  PIC X(150) WITH LOWER-BOUNDS.
77 PARAM-LEN  PIC 9(11) BINARY.
77 MSG-RSLT   PIC S9(11) BINARY.
PROCEDURE DIVISION.
MAIN.
    MOVE 1 TO MSG-NUM.
    CALL "GET_FROM_MESSAGES OF
-       "OBJECT/EXAMPLE/MLS/ALGOL/LIBRARY"
        USING MSG-NUM, MSG, MSG-LEN
        GIVING MSG-RSLT.

* This display, after calling GET_FROM_MESSAGES, should display
* "First message".
*
    DISPLAY MSG.
*
    MOVE 2 TO MSG-NUM.
    MOVE "ABC" TO PARAM-STR.
    MOVE 3 TO PARAM-LEN.
    CALL "GET_FROM_MESSAGES_ONE_PARAM OF
-       "OBJECT/EXAMPLE/MLS/ALGOL/LIBRARY"
        USING MSG-NUM, MSG, MSG-LEN, PARAM-STR, PARAM-LEN
        GIVING MSG-RSLT.

* This display, after calling GET_FROM_MESSAGES_ONE_PARAM, should
* display "ABC parameterized message".
*
    DISPLAY MSG.
*
    STOP RUN.
```

Example COBOL85 Program

Example B-5 shows a COBOL85 program that accesses the messages from the example ALGOL library with the explicit library CALL syntax. Note that a COBOL85 program can also access the messages from the example ALGOL library with the CALL syntax shown in the COBOL74 example.

```
IDENTIFICATION DIVISION.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
77 MSG-NUM          PIC S9(11)  USAGE BINARY.
01 MSG              PIC X(150).
77 MSG-LEN          PIC S9(11)  USAGE BINARY.
01 PARAM-STR        PIC X(150).
77 PARAM-LEN        PIC S9(11)  USAGE BINARY.
77 MSG-RSLT         PIC S9(11)  USAGE BINARY.

LOCAL-STORAGE SECTION.
LD  OUTMSG-PARAMS.
77 OP-MSG-NUM        PIC S9(11)  USAGE BINARY.
01 OP-MSG            PIC X(150)  WITH LOWER-BOUNDS.
77 OP-MSG-LEN        PIC S9(11)  USAGE BINARY.
01 OP-PARAM-STR      PIC X(150)  WITH LOWER-BOUNDS.
77 OP-PARAM-LEN      PIC S9(11)  USAGE BINARY.
77 OP-MSG-RSLT       PIC S9(11)  USAGE BINARY.

PROGRAM-LIBRARY SECTION.
LB  OUTPUT-MESSAGES-LIB IMPORT
*  ATTRIBUTE TITLE IS "OBJECT/EXAMPLE/MLS/ALGOL/LIBRARY".
  ATTRIBUTE TITLE IS "OBJECT/C85/CENTRAL/MLSALG".
  ENTRY PROCEDURE GET-FROM-MESSAGES
    FOR "GET_FROM_MESSAGES"
    WITH OUTMSG-PARAMS
    USING OP-MSG-NUM,
          OP-MSG,
          OP-MSG-LEN
    GIVING OP-MSG-RSLT.

  ENTRY PROCEDURE GET-FROM-MESSAGES-ONE-PARAM
    FOR "GET_FROM_MESSAGES_ONE_PARAM"
    WITH OUTMSG-PARAMS
    USING OP-MSG-NUM,
          OP-MSG,
          OP-MSG-LEN,

          OP-PARAM-STR,
          OP-PARAM-LEN
    GIVING OP-MSG-RSLT.
```

Examples of Using Libraries to Access Output Message Arrays

```
PROCEDURE DIVISION.  
  MAIN.  
    MOVE 1 TO MSG-NUM.  
    CALL GET-FROM-MESSAGES OF OUTPUT-MESSAGES-LIB  
      USING MSG-NUM,  
            MSG,  
            MSG-LEN  
      GIVING MSG-RSLT.  
  *   "First message" should be displayed.  
      DISPLAY MSG.  
  
    MOVE 2 TO MSG-NUM.  
    MOVE "ABC" TO PARAM-STR.  
    MOVE 3 TO PARAM-LEN.  
    CALL GET-FROM-MESSAGES-ONE-PARAM OF OUTPUT-MESSAGES-LIB  
      USING MSG-NUM,  
            MSG,  
            MSG-LEN,  
            PARAM-STR,  
            PARAM-LEN  
      GIVING MSG-RSLT.  
  *   "ABC parameterized message" should be displayed.  
      DISPLAY MSG.  
  
  STOP RUN.
```

Example Pascal Program

Example B-6 shows a Pascal program that accesses the messages from the example ALGOL library.

```
program p;

(* Declare the library message_lists, which contains the functions
   GET_FROM_MESSAGES and GET_FROM_MESSAGES_ONE_PARAM. *)

library message_lists
  (title = 'OBJECT/EXAMPLE/MLS/ALGOL/LIBRARY');

type str = packed array [1..100] of char;
var
  retn_msg: str;
  msg_len : integer;
  msg_result: integer;
(* Declare the external functions that reside in the message_lists
   library. *)

function get_from_messages (const msg_num: integer;
                           const msg   : str;
                           var  msg_len: integer): integer;

  message_lists;

function get_from_messages_one_param
  (const msg_num  : integer;
   const msg     : str;
   var  msg_len  : integer;
   const param_str : str;
   const param_len : integer
  ): integer;

  message_lists;

begin
msg_result:= get_from_messages (1, retn_msg, msg_len);

(* The following statement will display: 'First message' *)
display (retn_msg);

msg_result:=get_from_messages_one_param (2, retn_msg,
                                       msg_len, 'ABC',
                                       3);

(* The following statement will display:
   'ABC parameterized message' *)
display (retn_msg);
end.
```

Example RPG Program

Example B-7 shows an RPG program that accesses the messages from the example ALGOL library.

```

FOUTPUT O    80 80      REMOTE
E          MSG    80 1
E          EMSG   1 1 24
L*
L* Declare the external subroutines to be accessed via the library
L* facility.
L*
L* First, declare the library containing the output message arrays you
L* want to access. The program name of the library will be MSGLIB,
L* and linkage will be by title to OBJECT/EXAMPLE/MLS/ALGOL/LIBRARY.
L*
LIBRARY NAME MSGLIB    T OBJECT/EXAMPLE/MLS/ALGOL/LIBRARY
L*
L* Second, declare the subroutines within library MSGLIB you want to
L* call.
L*
L* MSG0, the subroutine name in this program, will link to the
L* function GET_FROM_MESSAGES in the library. Three parameters will
L* be passed to GET_FROM_MESSAGES, two integers passed by name, and
L* one EBCDIC array passed by reference with lower-bounds.
L*
LIBRARY SUBR MSG0     110 GET_FROM_MESSAGES
LIBRARY PARAMMSG_NUM  110
LIBRARY PARAMRETN_MSG *
LIBRARY PARAMMSG_LEN  110
L*
L* MSG1, the subroutine name in this program, will link to the
L* function GET_FROM_MESSAGES_ONE_PARAM in the library. Five
L* parameters will be passed to GET_FROM_MESSAGES_ONE_PARAM,
L* be passed to GET_FROM_MESSAGES, three integers passed by name, and
L* two EBCDIC arrays passed by reference with lower-bounds.
L*
LIBRARY SUBR MSG1     110 GET_FROM_MESSAGES_ONE_PARAM
LIBRARY PARAMMSG_NUM  110
LIBRARY PARAMRETN_MSG *
LIBRARY PARAMMSG_LEN  110
LIBRARY PARAMPARAM_STR *

```

Examples of Using Libraries to Access Output Message Arrays

```
LIBRARY PARAMPARAM_LEN 110
L*
C* Call MSG0 (GET_FROM_MESSAGES), set on indicator 99 if the
C* functional result is not zero, and execute the subroutine WMSG.
C*
C          EEXSRMSG0      RSLT  110
C          PARAM1
C          PARAM          MSG
C          PARAM          L      110
C          0              COMP RSLT          9999
C          EXSR WMSG
C*
C* Call MSG1 (GET_FROM_MESSAGES_ONE_PARAM), set on indicator 99 if the
C* functional result is not zero, and execute the subroutine WMSG.
C*
C          EEXSRMSG1      RSLT
C          PARAM2
C          PARAM          MSG
C          PARAM          L
C          PARAM'ABC'
C          PARAM3
C          0              COMP RSLT          9999
C          EXSR WMSG
C*
C* SETON the LR indicator to terminate the program. Without
C* a primary file the program will otherwise continuously loop
C* through calculations.
C*
C          SETON          LR
C*
C* The message returned has a null character placed at the end,
C* one character beyond the position indicated by the
C* returned message length. This subroutine replaces
C* the null character by a blank character before printing
C* the message.
C*
C* If the EEXSR result (in RSLT) is not equal to zero, an error
C* occurred while retrieving the message and indicator 99 will
C* be set. An error message will then be displayed on the
C* remote device.
C*
C          WMSG          BEGSR
C*
C          *IN99        IFNE '1'
C          ADD 1          L
C          MOVE ' '      MSG,L
C          END
```

Examples of Using Libraries to Access Output Message Arrays

```
C*
C          EXCPT

C* We need to fill MSG with blanks in case the next message is
C* shorter.
C          MOVE *BLANKS  MSG
C          ENDSR
O* The first execution of the output will print 'First message'.
O* The second execution will print 'ABC parameterized message'.
OOUTPUT  E
O          N99      MSG      80
O          99      EMSG     24
VVECTOR SEQ
ERROR RETRIEVING MESSAGE
```

Examples of Using Libraries to Access Output Message Arrays

Appendix C

Using the Complex Procedure

The CCSTOCCS_TRANS_TEXT_COMPLEX procedure is a character mapping operation that supports the complex translation requirements of mixed, multibyte and 16-bit coded character sets, as well as the standard 8-bit coded character sets. Coded character set numbers are used as input to determine the translate tables and the translation rule to be applied in the mapping.

In contrast to the standard mapping procedures, the Complex procedure maintains a State array parameter, so that mapping can be spread over multiple calls, and includes an Option parameter that controls how error recovery is handled and how mappings are terminated. For example, you can recover from situations where the source ends prematurely or the destination runs out of room unexpectedly by calling the Complex procedure again to continue the operation.

The mapping capabilities of the Complex procedure include discrete mode mapping, where the source is one complete string typically mapped in one call to one complete destination string, and stream mode mapping, where the source and destination flow continuously over multiple calls.

This appendix includes the specification of the Complex procedure, descriptions of the parameters, implementation strategies, and examples of using the Complex procedure and the Option parameter values in various mapping scenarios.

Specification

There are three versions of the Complex procedure that are exported from the CENTRALSUPPORT library: the Zero-bounded array, Star-bounded array, and Pointer. These three versions have parameters that are functionally equivalent but are declared differently to satisfy the needs of the different programming languages. The Zero-bounded and Star-bounded array versions capture and verify their parameters and then call the Pointer version to do the actual mapping of the source text into the destination array.

Zero-Bounded Array version

The Zero-bounded array version of the Complex procedure is defined to be called by COBOL85, ALGOL, and NEWP programs as follows:

```
INTEGER PROCEDURE CcsToCcs_Trans_Text_Complex (  
    Source_Ccsnum, Dest_Ccsnum,  
    Source, Source_Inx, Source_Bytes,  
    Dest, Dest_Inx, Dest_Bytes, State, Option);  
VALUE Source_Ccsnum, Dest_Ccsnum, Source_Bytes, Dest_Bytes, Option;  
EBCDIC ARRAY Source, Dest [0];  
REAL ARRAY State[0];  
INTEGER Source_Ccsnum, Dest_Ccsnum, Source_Inx, Source_Bytes,  
    Dest_Inx, Dest_Bytes, Option;
```

Refer to the *COBOL ANSI-85 Programming Reference Manual, Vol. 1*, the *ALGOL Programming Reference Manual, Vol. 1*, and the *NEWP Programming Reference Manual* for further details on declaring and calling the Complex procedure in each programming language.

Star-Bounded Array Version

The Star-bounded array version of the Complex procedure is defined to be called by C and Pascal programs as follows:

```

INTEGER PROCEDURE CcsToCcs_Trans_Text_Star (
    Source_Ccsnum, Dest_Ccsnum,
    Source, Source_Size, Source_Inx, Source_Bytes,
    Dest, Dest_Size, Dest_Inx, Dest_Bytes,
    State, State_Size, Option);
VALUE Source_Ccsnum, Dest_Ccsnum, Source_Size, Source_Bytes,
    Dest_Size, Dest_Bytes, State_Size, Option;
EBCDIC ARRAY Source, Dest [*];
REAL ARRAY State[0];
INTEGER Source_Ccsnum, Dest_Ccsnum, Source_Inx, Source_Bytes,
    Dest_Inx, Dest_Bytes, State_Size, Option;

```

Refer to the *C Programming Reference Manual, Vol. 1* and the *Pascal Programming Reference Manual, Vol. 1* for further details on declaring and calling the Complex procedure in each programming language.

Pointer Version

The Pointer version of the Complex procedure is defined to be called by ALGOL and NEWP programs as follows:

```

INTEGER PROCEDURE CcsToCcs_Trans_Text_Ptr(
    Source_Ccsnum, Dest_Ccsnum,
    PSource, Source_Inx, Source_Bytes,
    PDest, Dest_Inx, Dest_Bytes, State, Option);
VALUE Source_Ccsnum, Dest_Ccsnum, PSource, Source_Bytes,
    PDest, Dest_Bytes, Option;
POINTER PSource, PDest;
REAL ARRAY State[*];
INTEGER Source_Ccsnum, Dest_Ccsnum, Source_Inx, Source_Bytes,
    Dest_Inx, Dest_Bytes, Option;

```

Refer to the *ALGOL Programming Reference Manual, Vol. 1* and the *NEWP Programming Reference Manual* for further details on declaring and calling the Complex procedure in each programming language.

Parameters

Table C-1 lists the parameters to the Complex procedure specification. The Source_Size, Dest_Size, and State_Size parameters apply only to the Star-bounded array version. The PSource and PDest pointers apply only to the Pointer version.

Table C-1. Complex Procedure Parameters

Name	Type	Description
Source_Ccsnum	Input	Library number of the source CCS.
Dest_Ccsnum	Input	Library number of the destination CCS.
Source	Input	Contains the characters to be mapped. The maximum is 64,000 words.
Source_Size	Input	Size of the Source array.
PSource	Input	Pointer to the Source array. Equivalent to Source[Source_Inx]. Note: This pointer is not changed by the Complex procedure.
Source_Inx	Input	Specifies the byte offset (0 relative) where the mapping starts in Source. Ignored on input by the Pointer version.
	Output	Marks the offset where the mapping ended.
Source_Bytes	Input	Number of bytes to map beginning at Source_Inx. (Source_Bytes + Source_Inx) must be less than or equal to the size of the Source parameter.
Dest	Output	Contains the mapped text. Passing an array which is twice the size of the Source parameter always works.
Dest_Size	Input	Size of the Dest array.
PDest	Input	Pointer to the Dest array. Equivalent to Dest[Dest_Inx]. Note: This pointer is not changed by the Complex procedure.
Dest_Inx	Input	Specifies the byte offset (0 relative) where the mapped text is stored in Dest. Ignored on input by the Pointer version.
	Output	Marks the offset where the mapped text ended.
Dest_Bytes	Input	Maximum number of bytes beginning at Dest_Inx that can be filled with destination characters. (Dest_Bytes + Dest_Inx) must be less than or equal to the size of the Dest parameter.

Table C-1. Complex Procedure Parameters

Name	Type	Description
State	Input and Output	<p>A real array that is ten words long and owned by, but untouched by, the caller.</p> <p>Word 0: Indices to CENTRALSUPPORT tables Word 1: Previous operation status information Word 2: Partial source character Word 3: Special case optimization information Words 4-9: Reserved</p>
State_Size	Input	<p>Size of the State array. Must be equal to or greater than 10.</p>
Option	Input	<p>Controls continuation calls for multibyte mapping.</p> <p>In cases where the last character transferred into the destination array is a double-byte character, and the destination coded character set uses bracketing, you must specify whether the ending bracket is also mapped to the destination. For details, refer to "Terminating Calls with Single Bracket Invocation Characters" later in this Appendix.</p> <p>The following values control the mapping of data:</p> <p>0: CS_OPT_INITIAL_COMPLETEV Initializes State indices and previous operation status information. Represents the first call for a combination of source and destination coded character sets. The source data stream, which is not part of a prior call, is complete and the data can be mapped in one call.</p> <p>1: CS_OPT_TAILV Uses existing State indices and previous operation status information. Represents the second or higher call for a source data stream and is the last call.</p>

Table C-1. Complex Procedure Parameters

Name	Type	Description
Option <i>(Cont.)</i>	Input	<p>2: CS_OPT_INITIAL_HEADV Initializes State indices and previous operation status information. Represents the first call for a combination of source and destination coded character sets. The source data stream is incomplete or the destination is not large enough to handle all of the data in one call.</p> <p>3: CS_OPT_MIDDLEV Uses existing State indices and previous operation status information. Represents the second or higher call for a source data stream, but is not the last call.</p> <p>4: Same as 0.</p> <p>5: CS_OPT_COMPLETEV Uses existing State indices, but ignores previous operation status information. Represents a new source data stream, but is not the first call for the given combination of source and destination coded character sets. The source data stream, which is not part of a prior call, is complete and the data can be mapped in one call.</p> <p>6: Same as 2.</p> <p>7: CS_OPT_HEADV Uses existing State indices, but ignores previous operation status. Represents the beginning of a new source data stream, but is not the first call for the given combination of source and destination coded character sets. The source data stream is incomplete or the destination is not large enough to handle all of the data in one call.</p>

Interaction of the State and Option Parameters

The Complex procedure maintains a State array parameter, so that initialization overhead can be eliminated on subsequent calls. The State array parameter also keeps previous operation status information, so that multiple calls can be used to process a source data stream. The values of the Option parameter govern how the Complex procedure uses the information in the State array parameter.

State Array Parameter

The State array parameter contains two types of information:

- Static indices to the CENTRALSUPPORT tables derived from the source and destination coded character set numbers
- Previous operation status information

The static indices point to the translation rules record and to the mapping tables for the current source and destination coded character sets. The previous operation status information includes the “locked” state of the source and destination arrays and the beginning bytes of any incomplete source character. The locked state indicates whether the source and destination arrays are in the middle of double-byte characters.

The previous operation status information is used when calling the Complex procedure multiple times for a single source stream. The previous operation status information is also used to recover from CS_INCOMPLETE_CHARV errors, when the source terminates unexpectedly, or CS_INCOMPLETE_DATAV errors, when the destination runs out of room. The beginning bytes of an incomplete source character are stored in the State information and then combined with the new source data to construct a complete character.

Option Parameter

The Option parameter is used to control whether the Complex procedure initializes or reinitializes the indices to the CENTRALSUPPORT tables in the State array parameter, how the procedure uses the previous operation status information, and how to end the current operation.

The Complex procedure always terminates a mapping with the destination string ending in a complete character, which is either a single byte character, both bytes of a double-byte character, or a single state invocation character followed by a complete single or double-byte character. If the destination coded character set uses the start double-octet SDO and end double-octet EDO bracketing method to denote the beginning and ending of double-byte data, and if the last character placed in the destination array is a double-byte character, the Complex procedure does not automatically insert the ending bracket. The Option parameter indicates whether the ending bracket is added to the end of the destination array.

The following implementation strategies provide general guidelines for using the Complex procedure.

Implementation Strategies

The Complex procedure enables you to spread mapping over multiple calls for processing complex coded character sets. With this capability, you can use either discrete mode mapping, where one complete source string is typically mapped in one call to one complete destination string, or stream mode mapping, where the source and destination flow continuously over multiple calls.

When the destination cannot contain all of the source in a single call, or when it takes multiple calls to present all of the source, you must decide whether the destination can be mapped in discrete pieces that can stand alone or be mapped as a continuous stream.

Discrete Mode

In the most likely case, the source is discrete, complete in one call, and the destination is also discrete. You can use the following Option parameter values to control discrete mode mappings:

- On the first call of a given translation combination, if the source is discrete and the destination is big enough to handle the mapped characters in one call, use Option parameter value `CS_OPT_INITIAL_COMPLETEV` (0).
- On subsequent calls of a given translation combination, if the source is discrete and the destination is big enough to handle the mapped characters in one call, use Option parameter value `CS_OPT_COMPLETEV` (5).
- If the value returned on either of the above calls is `CS_INCOMPLETE_DATAV`, indicating that the destination ran out of room before all of the source could be transferred, use Option parameter value `CS_OPT_TAILV` (1) on a second call for the second destination string.

Continuous Stream Mode

Sometimes more than one call is necessary to present all of the source and the destination is to be mapped as one continuous stream. The destination array does not have to be the same on the first, middle, and last calls. You can use the following Option parameter values to control continuous stream mode mappings:

- On the first call of a given translation combination, for the first call of a stream, use Option parameter value `CS_OPT_INITIAL_HEADV` (2).
- On subsequent calls of a given translation combination, for the first call of a stream, use Option parameter value `CS_OPT_HEADV` (7).
- On the second and higher calls of a stream, use Option parameter value `CS_OPT_MIDDLEV` (3).
- On the last call of a stream, use Option parameter value `CS_OPT_TAILV` (1).

In continuous stream mode mapping, the ending bracket is not added until the last call. However, if there is not enough room in the destination array on the last call, the remaining source is truncated and the ending bracket is inserted.

Reducing Initialization Overhead

Consider the following guidelines to reduce the initialization overhead in your mapping operations:

- The most common use of the Complex procedure is the repeated use of the discrete mode case with Option parameter value CS_OPT_COMPLETEEV. The other examples provided in this appendix are uncommon scenarios for the typical user.
- When using the Complex procedure to translate multiple discrete source and destination strings for the same translation combination, only one call with Option parameter value CS_OPT_INITIAL_COMPLETEEV is necessary.

All subsequent calls can use Option parameter value CS_OPT_COMPLETEEV, which uses the existing static indices to point to the translation rules record and the translate tables needed for the translation combination.

- When translating back and forth between a pair of mixed, multibyte or 16-bit coded character sets, the use of two State array parameters is recommended to maintain the static indices information from call to call and allow for Option parameter value CS_OPT_INITIAL_COMPLETEEV to be used just once for each direction of the translation.

All subsequent calls can use Option parameter value CS_OPT_COMPLETEEV with the appropriate State array parameter.

- Using Option parameter value CS_OPT_COMPLETEEV instead of CS_OPT_INITIAL_COMPLETEEV reduces the amount of initialization overhead inside CENTRALSUPPORT. The static indices of the State array parameters can be initialized at the beginning of your program with a call to the Complex procedure using Option parameter value CS_OPT_INITIAL_COMPLETEEV and a Source_Bytes parameter value of zero (0).

Thereafter, the main loop of your program can use Option parameter value CS_OPT_COMPLETEEV with the appropriate State array parameter.

Terminating Calls with Single Bracket Invocation Characters

Terminating calls in discrete mode and continuous stream mode differs in the instance where the destination coded character set uses single bracket invocation and the last character in the destination string is double-byte. The Complex procedure ends the destination string with a complete character, but you must specify whether the ending bracket is also mapped to the destination.

- If the destination is to be used in discrete mode, where each destination array is complete in itself, then the destination receives the ending bracket. This is the case for Option parameter value CS_OPT_INITIAL_COMPLETEV and CS_OPT_COMPLETEV.
- If the call is the first of, or is in the middle of, a continuous stream, the ending bracket is not added. This is the case for Option parameter values CS_OPT_INITIAL_HEADV, CS_OPT_HEADV, and CS_OPT_MIDDLEV.
- If the call is the last call in a continuous stream, then the destination receives the ending bracket. This is the case for Option parameter value CS_OPT_TAILV.

Example Mappings

The following examples show how to use the Complex procedure and the Option parameter values to implement various mapping strategies. In each mapping scenario, the translation is from the JISASCIIJBIS7 coded character set to the JapanEBCDICJBIS8 coded character set. The complete source string is <ascsd>DDddDD<ascdo>, where "DDddDD" represents a string of multibyte data and <ascsd> and <ascdo> represent the ASCII encodings of the 8-bit starting and ending double-octet characters. The Dest array includes <sdo> and <edo>, which represent the EBCDIC 8-bit starting and ending double-octet characters.

In addition, assume that the source and destination coded character sets do not change and that the State array parameter has been initialized by a previous call that ended without error.

Mapping One Source String to One Discrete Destination String in One Call

This scenario represents the most typical case. The source string is a complete string and the destination array is large enough to contain the mapped data.

The first and only call is made with the Option parameter value equal to CS_OPT_COMPLETEV (State array initialized, ignore previous operation status, discrete destination).

The State array parameter has already been initialized and contains the indices to the CENTRALSUPPORT tables for the given source and destination coded character sets. The State array parameter also contains the operation status, which shows that the source is not in the middle of double-byte characters (s_not_locked) and that the destination is also not in the middle of double-byte characters (d_not_locked).

Call:

```

Option      CS_OPT_COMPLETEV
Source_Bytes 8
Source_Inx  0
Dest_Bytes  16 % Note: twice length of source.
Dest_Inx    0
Source      "<ascsd0>DDdddD<ascdo>"
Dest       ""

```

Return:

```

Dest      "<sdo>DDdddD<edo>"
Source_Inx 8 % Points beyond last good character.
Dest_Inx   8
State:    Indices to CENTRALSUPPORT tables
          Operation status:
          s_not_locked; d_not_locked;
          no character stored
Result:   CS_DATAOKV

```

Mapping One Complete Source String to Multiple Discrete Destination Strings

In this scenario, the caller wants to display data that might overflow one line and have to be presented as two or more discrete lines. The source string is a complete string and the caller wants discrete destinations, but the destination array might not be large enough to contain the mapped data.

The first call is made with the Option parameter value equal to CS_OPT_COMPLETEV (State array initialized, ignore previous operation status, discrete destination).

```
Call:
      Option      CS_OPT_COMPLETEV
      Source_Bytes 8
      Source_Inx   0
      Dest_Bytes   5 % Note: not twice length of source.
      Dest_Inx     0
      Source       "<ascsd0>DDddd<ascdeo>"
      Dest        ""

Return:
      Dest         "<sdo>DD<edo>"
      Source_Inx   3 % Points beyond last good character.
      Dest_Inx     4
      State:       Indices to CENTRALSUPPORT tables
                  Operation status:
                    s_locked; d_not_locked;
                    no character stored
      Result:      CS_INCOMPLETE_DATAV
```

The first call results in a CS_INCOMPLETE_DATAV error. To complete the mapping of the source string, a second and last call can be made with a new, larger destination array and the Option parameter value equal to CS_OPT_TAILV (State array initialized, use previous operation status, discrete destination). The Source_Inx parameter points to where the data mapping left off from the previous operation.

```
Call:
      Option      CS_OPT_TAILV
      Source_Bytes 5
      Source_Inx   3
      Dest_Bytes   10 % Note: twice length of source.
      Dest_Inx     0
      Source       "<ascsd0>DDddd<ascdeo>"
      Dest        ""
```

```
Return:
Dest      "<sdo>ddDD<edo>"
Source_Inx 8 % Points beyond last good character.
Dest_Inx   6
State:     Indices to CENTRALSUPPORT tables
           Operation status:
             s_not_locked; d_not_locked;
             no character stored
Result:    CS_DATAOKV
```

After the last call, the source string has been mapped into two discrete destination strings, neither of which completely fills the arrays in which they are stored. It is up to the caller to keep track of the length of data in each array.

Mapping One Source String to One or More Stream Destination Strings

In this scenario, the source string is a complete string but the destination array might not be large enough to handle the mapped data. The caller is prepared to have the source data flow into more than one destination array, and therefore uses the Option parameter value CS_OPT_HEADV, but discovers that all of the data fits in one call. A special second call is needed to add the ending bracket.

The first call is made with the Option parameter value equal to CS_OPT_HEADV (State array initialized, ignore previous operation status, stream destination).

Call:

```
Option      CS_OPT_HEADV
Source_Bytes 8
Source_Inx  0
Dest_Bytes  16 % Note: twice the length of source.
Dest_Inx    0
Source      "<ascsd>DDddd<ascdeo>"
Dest       ""
```

Return:

```
Dest        "<sdo>DDddd"
Source_Inx  8
Dest_Inx    7
State:      Indices to CENTRALSUPPORT tables
            Operation status:
            s_not_locked; d_locked;
            no character stored
Result:     CS_DATAOKV
```

The source string fits in the destination array, but the caller told the Complex procedure to stream the destination. The caller must make a second and last call with the Option parameter value equal to CS_OPT_TAILV (State array initialized, use previous operation status, discrete destination) and the Source_Bytes parameter equal to 0 to add the ending bracket. The Dest_Inx parameter points to where the data mapping left off from the previous operation.

Call:

```
Option      CS_OPT_TAILV
Source_Bytes 0 % Note: special case call.
Source_Inx  0
Dest_Bytes  9
Dest_Inx    7
Source      ""
Dest        "<sdo>DDddd"
```

Return:

Dest	"<sdo>DDddDD<edo>"
Source_Inx	0
Dest_Inx	8
State:	Indices to CENTRALSUPPORT tables
	Operation status:
	s_not_locked; d_not_locked;
	no character stored
Result:	CS_DATAOKV

After the last call, the destination array contains a well-formed discrete destination string.

Mapping One Source String to Multiple Stream Destination Strings

In this scenario, the source string is a complete string and the caller wants a single streamed destination string, but the destination array might not be large enough to contain the mapped data.

The first call is made with the Option parameter value equal to CS_OPT_HEADV (State array initialized, ignore previous operation status, stream destination).

Call:

```
Option      CS_OPT_HEADV
Source_Bytes 8
Source_Inx  0
Dest_Bytes  6 % Note: not twice the length of source.
Dest_Inx    0
Source      "<ascsd0>DDddDD<ascdeo>"
Dest       ""
```

Return:

```
Dest       "<sdo>DDdd"
Source_Inx 5 % Points beyond last good character.
Dest_Inx   5 % The last character did not fill the array.
State:     Indices to CENTRALSUPPORT tables
           Operation status:
           s_locked; d_locked;
           no character stored
Result:    CS_INCOMPLETE_DATAV
```

Unlike Figure C-3, the destination array is too small to hold all of the mapped source. To complete the mapping of the source string, a second and last call can be made with a new destination array and the Option parameter value equal to CS_OPT_TAILV (State array initialized, use previous operation status, discrete destination). The Source_Inx parameter points to where the data mapping left off from the previous operation.

Call:

```
Option      CS_OPT_TAILV
Source_Bytes 3
Source_Inx  5
Dest_Bytes  10 % Note: more than twice length of source.
Dest_Inx    0
Source      "<ascsd0>DDddDD<ascdeo>"
Dest       ""
```

```
Return:
Dest          "DD<edo>"
Source_Inx    8
Dest_Inx      3 % The destination array is not full.
State:        Indices to CENTRALSUPPORT tables
              Operation status:
                s_not_locked; d_not_locked;
                no character stored
Result:       CS_DATAOKV
```

After the last call, the source string is mapped into two destination arrays, neither of which are completely full of characters. It is up to the caller to keep track of the length of data in each array.

Mapping Multiple Stream Source Strings to One Discrete Destination String

In this scenario, the caller wants a single discrete destination built up from small strings of the source string. The source string is incomplete on the first call, but the destination array is large enough to contain the complete mapped source stream.

The first call is made with the Option parameter value equal to CS_OPT_HEADV (State array initialized, ignore previous operation status, stream destination).

Call:

```
Option      CS_OPT_HEADV
Source_Bytes 4
Source_Inx  0
Dest_Bytes  100 % Note: all the destination we'll accept.
Dest_Inx    0
Source      "<ascsd>DDd"
Dest       ""
```

Return:

```
Dest       "<sdo>DD"
Source_Inx 3 % Points beyond last good character.
Dest_Inx   3
State:     Indices to CENTRALSUPPORT tables
           Operation status:
           s_locked; d_locked;
           "d" stored
Result:    CS_INCOMPLETE_CHARV
```

To continue the mapping of the source string, a second call can be made with a new middle source string and the Option parameter value equal to CS_OPT_MIDDLEV (State array initialized, use previous operation status, stream destination). The Dest_Inx parameter points to where the data mapping left off from the previous operation.

Call:

```
Option      CS_OPT_MIDDLEV
Source_Bytes 1
Source_Inx  0
Dest_Bytes  97
Dest_Inx    3
Source      "d"
Dest       "<sdo>DD"
```

```

Return:
Dest          "<sdo>DDdd"
Source_Inx    1
Dest_Inx      5
State:        Indices to CENTRALSUPPORT tables
              Operation status:
                s_locked; d_locked;
                no character stored
Result:       CS_DATAOKV
    
```

To complete the mapping of the source string, a third and last call can be made with a new ending source string and the Option parameter value equal to CS_OPT_TAILV (State array initialized, use previous operation status, discrete destination). The Dest_Inx parameter points to where the data mapping left off from the previous operation.

```

Call:
Option        CS_OPT_TAILV
Source_Bytes   3
Source_Inx     0
Dest_Bytes     95
Dest_Inx       5
Source         "DD<ascedo>"
Dest          "<sdo>DDdd"
    
```

```

Return:
Dest          "<sdo>DDddDD<edo>"
Source_Inx    3
Dest_Inx      8
State:        Indices to CENTRALSUPPORT tables
              Operation status:
                s_not_locked; d_not_locked;
                no character stored
Result:       CS_DATAOKV
    
```

After the last call, the three source pieces have been combined into a single complete destination string.

Mapping Multiple Stream Source Strings to Multiple Discrete Destination Strings

In this scenario, the source string is made up from small pieces and each piece is built into a discrete destination array. The source string is an incomplete string, the caller wants discrete destinations, and the destination array is large enough to contain the mapped source string.

The first call is made with the Option parameter equal to CS_OPT_COMPLETEV (State array initialized, ignore previous operation status, discrete destination).

Call:

```
Option      CS_OPT_COMPLETEV
Source_Bytes 4
Source_Inx  0
Dest_Bytes  8 % Note: twice length of source.
Dest_Inx    0
Source      "<ascsd>DDd>"
Dest       ""
```

Return:

```
Dest        "<sdo>DD<edo>"
Source_Inx  3 % Points beyond last good character.
Dest_Inx    4
State:      Indices to CENTRALSUPPORT tables
            Operation status:
            s_locked; d_not_locked;
            "d" stored
Result:     CS_INCOMPLETE_CHARV
```

To complete the mapping of the source string, a second and last call can be made with a new source string, a new destination array, and the Option parameter value equal to CS_OPT_TAILV (State array initialized, use previous operation status, discrete destination).

Call:

```
Option      CS_OPT_TAILV
Source_Bytes 4
Source_Inx  0
Dest_Bytes  8 % Note: twice length of source.
Dest_Inx    0
Source      "dDD<ascdo>"
Dest       ""
```

```
Return:
  Dest      "<sdo>ddDD<edo>"
  Source_Inx 4
  Dest_Inx  6
  State:     Indices to CENTRALSUPPORT tables
             Operation status:
             s_not_locked; d_not_locked;
             no character stored
  Result:    CS_DATAOKV
```

After the last call, the source strings have been mapped into two discrete destination strings, neither of which completely fill the arrays in which they are stored. It is up to the caller to keep track of the length of data in each array.

Mapping Multiple Stream Source Strings to Multiple Stream Destination Strings

In this scenario, the caller wants to stream both the source and the destination until the end of the file. Unlike Figure C-6, the destination is spread over more than one array. The source stream is incomplete on the first call, but the destination array is large enough to contain the mapped source string.

The first call is made with the Option parameter equal to CS_OPT_HEADV (State array initialized, ignore previous operation status, stream destination).

```
Call:
      Option      CS_OPT_HEADV
      Source_Bytes 4
      Source_Inx   0
      Dest_Bytes   8 % Note: twice the length of source.
      Dest_Inx     0
      Source       "<ascsd>DDd"
      Dest         ""

Return:
      Dest         "<sdo>DD"
      Source_Inx   3 % Points beyond last good character.
      Dest_Inx     3
      State:       Indices to CENTRALSUPPORT tables
                  Operation status:
                    s_locked; d_locked;
                    "d" stored
      Result:      CS_INCOMPLETE_CHARV
```

To complete the mapping of the source string, a second and last call can be made with a new source string, a new destination array, and the Option parameter equal to CS_OPT_TAILV (State array initialized, use previous operation status, discrete destination).

```
Call:
      Option      CS_OPT_TAILV
      Source_Bytes 4
      Source_Inx   0
      Dest_Bytes   8 % Note: twice the length of source.
      Dest_Inx     0
      Source       "dDD<ascdo>"
      Dest         ""
```

```
Return:
  Dest      "ddDD<edo>"
  Source_Inx 4
  Dest_Inx  5
  State:    Indices to CENTRALSUPPORT tables
           Operation status:
             s_not_locked; d_not_locked;
             no character stored
  Result:   CS_DATAOKV
```

After the last call, the source string is mapped into two stream destination strings, neither of which completely fill the arrays in which they are stored. It is up to the caller to keep track of the length of data in each array.

Appendix D

Help Text Available for Localization

Appendix D lists the products with help text that can be translated. This appendix also applies the following information:

- The help text symbol file name for each product
- The help book or help library associated with each product
- The help utility used for processing the help text symbol file or help library

The two utilities used to process help text symbol files are as follows:

- Help Utility

To translate help text created with the Help Utility, you need the help text symbol file for the particular product that you want to translate. Note that help text symbol files are not provided with the purchase of a product—they must be purchased separately.

Once you have the help text symbol file, follow the guidelines provided under “Localizing Help Text” in Section 5 of this guide.

- SDF Plus Help utility

To translate help text created with the SDF Plus Help utility, you need either the help text symbol file or the help library for the particular product that you want to translate. Note that the help text symbol files are not provided with the purchase of a product—they must be purchased separately. Help libraries are provided with the purchase of a product.

If you are using the help text symbol file, follow the guidelines provided under “Localizing Help Text” in Section 5 of this guide. If you are using the help library, refer to “Localizing SDF Plus Screens” in Section 5 of this guide for an overview of the process. For complete details on translating help libraries, refer to the *SDF Plus Installation and Operations Guide*.

Help Text Available for Localization

Table D-1 lists the products that have help text available for translation, the name of the help book or help library associated with the product, and the help utility used to generate the help text.

Table D-1. Help Text Available for Translation

Product	Help Text Symbol File Name Help Book/Library Name	Type of Help
ALGOL Test and Debug System (TADS)	SOURCE/ALGOLTADS/UGD BOOK/ALGOLTADS/ENGLISH	Help Utility
BNA Version 2	SYMBOL/BNAV2/ENGLISH BOOK/BNAV2/ENGLISH	Help Utility
COBOL Migration Tool (CMT)	SYMBOL/CMT/HELPLIB1	SDF Plus
COBOL74 TADS	SOURCE/COBOL74TADS/UGD BOOK/COBOL74TADS/ENGLISH	Help Utility
COBOL74LOADER	SYMBOL/COBOL74LOADER/ENGLISH HELPLIB/COBOL74LOADER/ 1COBOL74LOADER	SDF Plus
Command and Edit (CANDE)	SYMBOL/CANDE/ENGLISH BOOK/CANDE/ENGLISH	Help Utility
DATAAID	SYMBOL/DATAAID/ENGLISH BOOK/DATAAID/ENGLISH	Help Utility
Documentation Library Overview	SYMBOL/DOCLIB/ENGLISH BOOK/DOCLIB/ENGLISH	Help Utility
Glossary	SYMBOL/GLOSS/ENGLISH BOOK/GLOSS/ENGLISH	Help Utility
InfoExec Advanced Data Dictionary System (ADDS)	SYMBOL/ADDS HELPLIB/ADDS/1ADDS SYMBOL/ADDS2 HELPLIB/ADDS2/1ADDS2/ENGLISH	SDF Plus
InfoExec Interactive Query Facility (IQF)	SYMBOL/IQF/ENGLISH HELPLIB/IQF/1IQF	SDF Plus
InfoExec main menus	SYMBOL/IEMANAGER/ENGLISH HELPLIB/IEMANAGER/1IEMANAGER	SDF Plus
InfoExec Operations Control Manager (OCM)	SYMBOL/OCM/ENGLISH HELPLIB/OCM/1OCM/ENGLISH SYMBOL/DSGL/ENGLISH HELPLIB/DSGL/1DSGL	SDF Plus
Interactive Configurator (IC)	SYMBOL/ICONFIG/ENGLISH HELPLIB/ICONFIG/1ICONFIGLIB	SDF Plus
Interactive Datacomm Configurator (IDC) long help	SYMBOL/IDC/ENGLISH BOOK/IDC/ENGLISH	Help Utility
IDC short help	SYMBOL/IDC/SHORTHELP/ENGLISH BOOK/IDC/SHORTHELP/ENGLISH	Help Utility

Table D-1. Help Text Available for Translation

Product	Help Text Symbol File Name Help Book/Library Name	Type of Help
Interactive Menugraph Generator (IMG) long help	SYMBOL/IMG/ENGLISH BOOK/IMG/ENGLISH	Help Utility
IMG short help	SYMBOL/IMG/SHORTHELP/ENGLISH BOOK/IMG/SHORTHELP/ENGLISH	Help Utility
Interactive Sort (ISORT)	SYMBOL/ISORT/ENGLISH HELPLIB/ISORT/1ISORT/ENGLISH	SDF Plus
KEYEDIOII	HELPLIB/KEYEDIOII/1KEYEDIOII BOOK/KEYEDIOII/ENGLISH	Help Utility
Mail system	SYMBOL/MAIL/ENGLISH BOOK/MAIL/ENGLISH	Help Utility
Menu-Assisted Resource Control (MARC) long help	SYMBOL/MARC/ENGLISH BOOK/MARC/ENGLISH	Help Utility
MARC short help	SYMBOL/MARC/SHORTHELP/ENGLISH BOOK/MARC/SHORTHELP/ENGLISH	Help Utility
MARC Open Systems Interconnection (OSI)	SYMBOL/NETADMIN/ENGLISH BOOK/NETADMIN/ENGLISH	Help Utility
Message Translation Utility (MSGTRANS)	SYMBOL/MSGTRANS/ENGLISH BOOK/MSGTRANS/ENGLISH	Help Utility
ODBC Access Drivers	TEXTLIB/INFOACCESS/IAUTILITY	SDF Plus
Remote Database Backup	SYMBOL/RDBUTILITY/ENGLISH HELPLIB/RDB/1RDBUTILITY	SDF Plus
Remote Job Entry/Bisynchronous Communication (RJE/BSC)	SYMBOL/RJEBSC/ENGLISH BOOK/RJEBSC/ENGLISH	Help Utility
Screen Design Facility (SDF)	SYMBOL/SCREENDESIGN/ENGLISH BOOK/SCREENDESIGN/ENGLISH	Help Utility
Screen Design Facility Plus (SDF Plus) Forms Editor	SYMBOL/SDFPLUS/FEHELP/ENGLISH HELPLIB/SDFPLUS/1FEHELP	SDF Plus
SDF Plus Forms Processor	SYMBOL/SDFPLUS/FPHELP/ENGLISH HELPLIB/SDFPLUS/1FPHELP	SDF Plus
SDF Plus Migrate Tool	SYMBOL/SDFPLUS/FMHELP/ENGLISH HELPLIB/SDFPLUS/1FMHELP	SDF Plus
SDF Plus Workstation	SYMBOL/SDFPLUS/WSFPHELP/ENGLISH HELPLIB/SDFPLUS/1SDFKWS	SDF Plus
Simple Installation (SI)	SYMBOL/SIMPLEINSTALL/ENGLISH BOOK/SIMPLEINSTALL/ENGLISH	Help Utility

Help Text Available for Localization

Table D-1. Help Text Available for Translation

Product	Help Text Symbol File Name Help Book/Library Name	Type of Help
Transaction Server	SYMBOL/COMS/ENGLISH BOOK/COMS/ENGLISH	Help Utility

Appendix E

Messages Available for Localization

This appendix provides information for translating messages in those products that can have their messages translated by the Message Translation Utility (MSGTRANS).

To translate messages for a particular product, you need to know the name of the object file that contains the translatable messages. Table E-1 lists the object files for each product that contains messages that can be translated with MSGTRANS. Once you have identified the file or files that you need, refer to the *MSGTRANS Operations Guide* for complete instructions on translating messages.

Table E-1. Messages Available for Localization with MSGTRANS

Product	Object File That Contains Messages
ALGOL compiler	*SYSTEM/ALGOL
ALGOL TADS	*SYSTEM/TADSSUPPORT
Binder	*SYSTEM/BINDER
C compiler	*SYSTEM/SLICESUPPORT (translate the messages in the STATMSG, LOCALES, and C_ERRNO_TEXT arrays). *SYSTEM/CC (translate the messages in the SHELL_ERRORS, CG_ERRORS, CC_ERRORS, and OPT_ERRORS arrays). Note: Translate the STATMSG array only for programs that set the STATISTICS compiler control option.
C TADS	*SYSTEM/SLICESUPPORT (Translate the messages in the LOCALES and C_ERRNO_TEXT arrays). *SYSTEM/CCTADS (Translate the messages in the SHELL_ERRORS, CG_ERRORS, CC_ERRORS, TADS_ERRORS, and OPT_ERRORS arrays).
CENTRALSUPPORT library	*SYSTEM/CENTRALSUPPORT

Table E-1. Messages Available for Localization with MSGTRANS

Product	Object File That Contains Messages
Client Access Services	*SYSTEM/NXSERVICES/ADMINCENTER *SYSTEM/NXSERVICES/CONTROL *SYSTEM/NXSERVICES/DEFAULTCONFIG (text commands) *SYSTEM/NXSERVICES/FILESERVER *SYSTEM/NXSERVICES/NAMESERVER *SYSTEM/NXSERVICES/PRINTTRACE *SYSTEM/NXSERVICES/SERVER
COBOL74 compiler	*SYSTEM/COBOL74
COBOL74 TADS	*SYSTEM/C74TADSSUPPORT
COBOL74LOADER Utility	*SYSTEM/DATABASESUPPORT (Translate only the messages in the CLU_ERROR_MESSAGES output message array).
COBOL85 compiler	*SYSTEM/SLICESUPPORT (translate only the messages in the STATMSG array). *SYSTEM/COBOL85 (translate the messages in the SHELL_ERRORS, CG_ERRORS, COBOL_ERRORS, and OPT_ERRORS arrays). Note: Translate the STATMSG array only for programs that set the STATISTICS compiler control option.
COBOL85 TADS	*SYSTEM/COBOL85TADS (Translate the messages in the SHELL_ERRORS, CG_ERRORS, COBOL_ERRORS, TADS_ERRORS, and OPT_ERRORS arrays).
Command and Edit (CANDE)	*SYSTEM/CANDE
Data Comm ALGOL compiler	*SYSTEM/DCALGOL
Database Certification	*SYSTEM/DBCERTIFICATION
Database Monitor program	*SYSTEM/DMMONITOR
Database Operations Center	*SYSTEM/DBCENTER/SERVER
DCDLPDUMPANALYZER utility	DCDLPDUMPANALYZER
Disk Cache	*SYSTEM/DISKCACHESUPPORT
DMALGOL compiler	*SYSTEM/DMALGOL
DMINTERPRETER library	*SYSTEM/BUILDINQ
DUMPALL utility	*SYSTEM/DUMPALL

Table E-1. Messages Available for Localization with MSGTRANS

Product	Object File That Contains Messages
Enterprise Database Server for ClearPath MCP	*SYSTEM/ACCESSROUTINES *SYSTEM/DASDL *SYSTEM/DMUTILITY *SYSTEM/DMDUMPPDIR *SYSTEM/DMDUMPPDIR/LIBRARY *SYSTEM/DMRECOVERY *SYSTEM/BUILDREORG *SYSTEM/PRINTAUDIT *SYSTEM/COPYAUDIT *SYSTEM/LOADDUMP *SYSTEM/DMINTERFACE *SYSTEM/DMCONTROL *SYSTEM/DMDATARECOVERY
Enterprise Database Server Inquiry	*SYSTEM/BUILDINQ
Enterprise Database Server Transaction Processing System (TPS)	*SYSTEM/TFL *SYSTEM/TRINTERFACE
FAULTLOGGER utility	*SYSTEM/FAULTLOGSUPPORT
FILEDATA utility	*SYSTEM/FILEDATA
GENERALSUPPORT library	*SYSTEM/GENERALSUPPORT
Generic Security Services—API	*SYSTEM/GSSAPI/SUPPORT
HLCN	*SYSTEM/HLCN/SUPPORT *SYSTEM/HLCN/TLISUPPORT
HLCNTS	*SYSTEM/HLCN/TERMINALSUPPORT
Host Services (file transfer, logical I/O, ODT protocol, remote tasking, Native File Transfer, station transfer, and Telnet Services)	*SYSTEM/DSSSUPPORT (Translate all messages except those in the output message arrays FTAM_MESSAGES and TCP_MESSAGES).
HWERRORSUPPORT library	*SYSTEM/HWERRORSUPPORT
InfoExec Advanced Data Dictionary System (ADDS)	*SYSTEM/ADDS/MANAGER *SYSTEM/ADDS/INTERFACE *SYSTEM/ADDS/UTILITIES *SYSTEM/ADDS/DMSIILoader *SYSTEM/ADDS/REPORTS *SYSTEM/ADDS/MIGRATE *SYSTEM/ADDS/DASDLGENERATOR *SYSTEM/DATABASESUPPORT *SYSTEM/IEMANAGER
InfoExec Operations Control Manager (OCM)	*SYSTEM/OCM

Table E-1. Messages Available for Localization with MSGTRANS

Product	Object File That Contains Messages
Interactive Datacomm Configurator (IDC)	*SYSTEM/IDC
Interactive Sort (ISORT)	*SYSTEM/ISORT
JOBFORMATTER	*SYSTEM/JOBFORMATTER
KEYEDIOII utility	*SYSTEM/KEYEDIOII/UTILITY
LANPRT (LAN Printing Enabler)	*SYSTEM/LANPRT/SUPPORT
Local port provider (LPP)	*SYSTEM/LPPSUPPORT
LOGANALYZER utility	*SYSTEM/LOGANALYZER
Business Information Server Relational Interface for ClearPath NX and A Series	*SYSTEM/MRISUPPORT *SYSTEM/MRI/ADMN *SYSTEM/MAPPER/HLCNSUPPORT *SYSTEM/MAPPER/TCPSUPPORT
Business Information Server System for ClearPath NX and A Series	*SYSTEM/MAPPER *SYSTEM/LCMAPPER *SYSTEM/MAPPER/INSTALL *SYSTEM/MAPPER/MRIM *SYSTEM/MAPPERSUPPORT *OBJECT/MAPPER *OBJECT/MAPPER/APLOAD *OBJECT/MAPPER/CPYGIN *OBJECT/MAPPER/II *OBJECT/MAPPER/MAPPRNT *OBJECT/MAPPER/MAPSTART *OBJECT/MAPPER/MENUS *OBJECT/MAPPER/PARM *OBJECT/MAPPER/PURGE *OBJECT/MAPPER/PURGETI *OBJECT/MAPPER/RECOVER
MARCUS	*SYSTEM/MARCUS
Master control program (MCP)	*SYSTEM/ASD/MCP *SYSTEM/ASDX/MCP *SYSTEM/ASD/MCP/DIAGNOSTICS *SYSTEM/ASDX/MCP/DIAGNOSTICS
Menu-Assisted Resource Control (MARC)	*SYSTEM/MARC/COMMANDER
MESSAGESUPPORT library	*SYSTEM/MESSAGESUPPORT
Network Database OLE DB Data Provider for ClearPath OS 2200	*SYSTEM/<what>/<what>
Network Definition Language II (NDLII) compiler	*SYSTEM/NDLII

Table E-1. Messages Available for Localization with MSGTRANS

Product	Object File That Contains Messages
Network Definition Language II (NDLII) post compiler	*SYSTEM/NPC
NETWORKSUPPORT library	*SYSTEM/NETWORKSUPPORT
OBDC Access Drivers	*SYSTEM/INFOACCESS/ADMINSTRATOR *SYSTEM/INFOACCESS/ADMINSTRATOR/DIAGNOSTICS *SYSTEM/INFOACCESS/UTILITY *SYSTEM/INFOACCESS/UTILITY/DIAGNOSTICS *SYSTEM/INFOACCESS/UTILITIESUPPORT *SYSTEM/INFOACCESS/UTILITIESUPPORT/DIAGNOSTICS
OLE DB Data Provider for ClearPath MCP	*SYSTEM/<what>/<what>
ONC+ Remote Procedure Call	*SYSTEM/ONC/RPCSUPPORT
OSI-FTAM (Open Systems Interconnection File Transfer, Access, and Management)	*SYSTEM/DSSSUPPORT(Translate only the messages in the output message array FTAM_MESSAGES.) *SYSTEM/FTAMSUPPORT
Programmer's Workbench for ClearPath MCP	*SYSTEM/NXEDIT/SUPPORT
Remote Data Backup	*SYSTEM/RDBUTILITY *SYSTEM/RDBSERVER *SYSTEM/RDBSUPPORT
Remote Display (RD) utility	*SYSTEM/RD
Remote Print System	*SYSTEM/PRINT/SUPPORT
RPG compiler	*SYSTEM/RPG *SYSTEM/RPGSUPPORT
RPRINTER	*SYSTEM/NWS/RPRINTER
Screen Design Facility (SDF)	*SYSTEM/SCREENDSIGN *SYSTEM/SCREENDSIGN/MANAGER *SYSTEM/SCREENDSIGN /INTERFACE *SYSTEM/SCREENDSIGN/DEFINES
Screen Design Facility Plus (SDF Plus)	*SYSTEM/SDFPLUS/ARCHIVEMANAGER
Simple Installation (SI)	*SYSTEM/SIMPLEINSTALL
SORT compiler	*SYSTEM/SORT
System Assistant	*SYSTEM/ASSISTANT *SYSTEM/ASSISTANT/STANDALONE
TCP/IP Remote Printing	*SYSTEM/TCPPRTSUPPORT

Messages Available for Localization

Table E-1. Messages Available for Localization with MSGTRANS

Product	Object File That Contains Messages
Transaction Server	*SYSTEM/COMS (Translate only the messages in the MLS_MESSAGES, MSGS, PACE_MESSAGES and DLGMGR_MESSAGES output message arrays).
TransIT Open/OLTP for ClearPath NX and A Series	*SYSTEM/DTP/BUILD/SERVER *SYSTEM/DTP/TM *SYSTEM/DTP/DA *SYSTEM/DTP/DA/PSH *SYSTEM/COMS (Translate only the messages in the XOPEN_ERROR_MESSAGES and the EXTENDED_ERROR_MESSAGES arrays).
Transmission Control Protocol/Internet Protocol (TCP/IP) Application Services for ClearPath MCP	*SYSTEM/DSSSUPPORT (Translate only the messages in the output message array TCP_MESSAGES). *SYSTEM/FTP/SUPPORT
WinRPC	*SYSTEM/WIN/EIDL *SYSTEM/WIN/RPCSUPPORT *SYSTEM/WIN/RPCSUPPORT/DIAGNOSTICS
Work Flow Language (WFL)	*SYSTEM/WFLSUPPORT

Appendix F

Understanding Railroad Diagrams

This appendix explains railroad diagrams, including the following concepts:

- Paths of a railroad diagram
- Constants and variables
- Constraints

The text describes the elements of the diagrams and provides examples.

Railroad Diagram Concepts

Railroad diagrams are diagrams that show you the standards for combining words and symbols into commands and statements. These diagrams consist of a series of paths that show the allowable structures of the command or statement.

Paths

Paths show the order in which the command or statement is constructed and are represented by horizontal and vertical lines. Many commands and statements have a number of options so the railroad diagram has a number of different paths you can take.

The following example has three paths:



The three paths in the previous example show the following three possible commands:

- REMOVE
- REMOVE SOURCE
- REMOVE OBJECT

A railroad diagram is as complex as a command or statement requires. Regardless of the level of complexity, all railroad diagrams are visual representations of commands and statements.

Understanding Railroad Diagrams

Railroad diagrams are intended to show

- Mandatory items
- User-selected items
- Order in which the items must appear
- Number of times an item can be repeated
- Necessary punctuation

Follow the railroad diagrams to understand the correct syntax for commands and statements. The diagrams serve as quick references to the commands and statements.

The following table introduces the elements of a railroad diagram:

Table F-1. Elements of a Railroad Diagram

The diagram element . . .	Indicates an item that . . .
Constant	Must be entered in full or as a specific abbreviation
Variable	Represents data
Constraint	Controls progression through the diagram path

Constants and Variables

A constant is an item that must be entered as it appears in the diagram, either in full or as an allowable abbreviation. If part of a constant appears in boldface, you can abbreviate the constant by

- Entering only the boldfaced letters
- Entering the boldfaced letters plus any of the remaining letters

If no part of the constant appears in boldface, the constant cannot be abbreviated.

Constants are never enclosed in angle brackets (< >) and are in uppercase letters.

A variable is an item that represents data. You can replace the variable with data that meets the requirements of the particular command or statement. When replacing a variable with data, you must follow the rules defined for the particular command or statement.

In railroad diagrams, variables are enclosed in angle brackets.

In the following example, BEGIN and END are constants, whereas <statement list> is a variable. The constant BEGIN can be abbreviated, since part of it appears in boldface.

— **BEGIN** —<statement list>— END —————|

Valid abbreviations for BEGIN are

- BE
- BEG
- BEGI

Constraints

Constraints are used in a railroad diagram to control progression through the diagram. Constraints consist of symbols and unique railroad diagram line paths. They include

- Vertical bars
- Percent signs
- Right arrows
- Required items
- User-selected items
- Loops
- Bridges

A description of each item follows.

Vertical Bar

The vertical bar symbol (|) represents the end of a railroad diagram and indicates the command or statement can be followed by another command or statement.

— SECONDWORD — (—<arithmetic expression>—) —————|

Percent Sign

The percent sign (%) represents the end of a railroad diagram and indicates the command or statement must be on a line by itself.

— STOP —————%

Right Arrow

The right arrow symbol (>) is used when the railroad diagram is too long to fit on one line and must continue on the next

- Is used when the railroad diagram is too long to fit on one line and must continue on the next
- Appears at the end of the first line, and again at the beginning of the next line

— SCALERIGHT — (—<arithmetic expression>— , —————>
><arithmetic expression>—) —————|

Required Item

A required item can be

- A constant
- A variable
- Punctuation

If the path you are following contains a required item, you must enter the item in the command or statement; the required item cannot be omitted.

A required item appears on a horizontal line as a single entry or with other items. Required items can also exist on horizontal lines within alternate paths, or nested (lower-level) diagrams.

In the following example, the word EVENT is a required constant and <identifier> is a required variable:

— EVENT —<identifier>—————|

User-Selected Item

A user-selected item can be

- A constant
- A variable
- Punctuation

User-selected items appear one below the other in a vertical list. You can choose any one of the items from the list. If the list also contains an empty path (solid line) above the other items, none of the choices are required.

In the following railroad diagram, either the plus sign (+) or the minus sign (–) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.

— [+] —<arithmetic expression>—————|
— [–] —

Loop

A loop represents an item or a group of items that you can repeat. A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, connected on both sides by vertical lines. The top line is a right-to-left path that contains information about repeating the loop.

Some loops include a return character. A return character is a character—often a comma (,) or semicolon (;)—that is required before each repetition of a loop. If no return character is included, the items must be separated by one or more spaces.

— [← ;] —<field value>—————|

Bridge

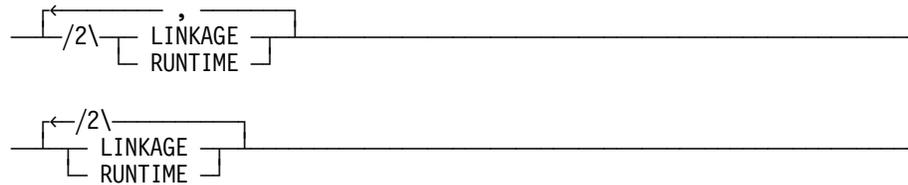
A loop can also include a bridge. A bridge is an integer enclosed in sloping lines (/ \) that

- Shows the maximum number of times the loop can be repeated
- Indicates the number of times you can cross that point in the diagram

The bridge can precede both the contents of the loop and the return character (if any) on the upper line of the loop.

Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

In the first bridge example, you can enter LINKAGE or RUNTIME no more than two times. In the second bridge example, you can enter LINKAGE or RUNTIME no more than three times.



In some bridges an asterisk (*) follows the number. The asterisk means that you must cross that point in the diagram at least once. The maximum number of times that you can cross that point is indicated by the number in the bridge.



In the previous bridge example, you must enter LINKAGE at least once but no more than twice, and you can enter RUNTIME any number of times.

Following the Paths of a Railroad Diagram

The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path; others have several alternate paths that provide choices in the commands or statements.

The following railroad diagram indicates only one path that requires the constant LINKAGE and the variable <linkage mnemonic>:



Alternate paths are provided by

- Loops
- User-selected items
- A combination of loops and user-selected items

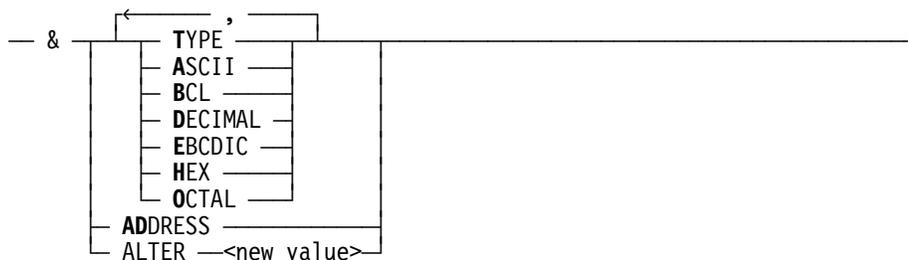
More complex railroad diagrams can consist of many alternate paths, or nested (lower-level) diagrams, that show a further level of detail.

For example, the following railroad diagram consists of a top path and two alternate paths. The top path includes

- An ampersand (&)
- Constants that are user-selected items

These constants are within a loop that can be repeated any number of times until all options have been selected.

The first alternative path requires the ampersand and the required constant ADDRESS. The second alternative path requires the ampersand followed by the required constant ALTER and the required variable <new value>.



Railroad Diagram Examples with Sample Input

The following examples show five railroad diagrams and possible command and statement constructions based on the paths of these diagrams.

Example 1

<lock statement>

— LOCK — (— <file identifier> —) —————|

Sample Input	Explanation
LOCK (FILE4)	LOCK is a constant and cannot be altered. Because no part of the word appears in boldface, the entire word must be entered. The parentheses are required punctuation, and FILE4 is a sample file identifier.

Example 2

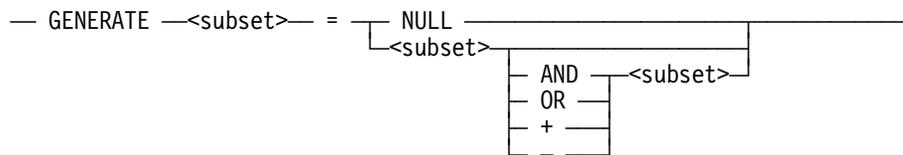
<open statement>

— OPEN — [INQUIRY] [UPDATE] — <database name> —————|

Sample Input	Explanation
OPEN DATABASE1	The constant OPEN is followed by the variable DATABASE1, which is a database name. The railroad diagram shows two user-selected items, INQUIRY and UPDATE. However, because an empty path (solid line) is included, these entries are not required.
OPEN INQUIRY DATABASE1	The constant OPEN is followed by the user-selected constant INQUIRY and the variable DATABASE1.
OPEN UPDATE DATABASE1	The constant OPEN is followed by the user-selected constant UPDATE and the variable DATABASE1.

Example 3

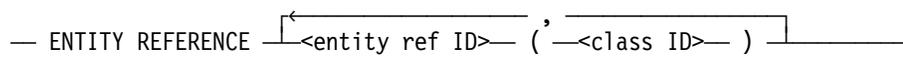
<generate statement>



Sample Input	Explanation
GENERATE Z = NULL	The GENERATE constant is followed by the variable Z, an equal sign (=), and the user-selected constant NULL.
GENERATE Z = X	The GENERATE constant is followed by the variable Z, an equal sign, and the user-selected variable X.
GENERATE Z = X AND B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the AND command (from the list of user-selected items in the nested path), and a third variable, B.
GENERATE Z = X + B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the plus sign (from the list of user-selected items in the nested path), and a third variable, B.

Example 4

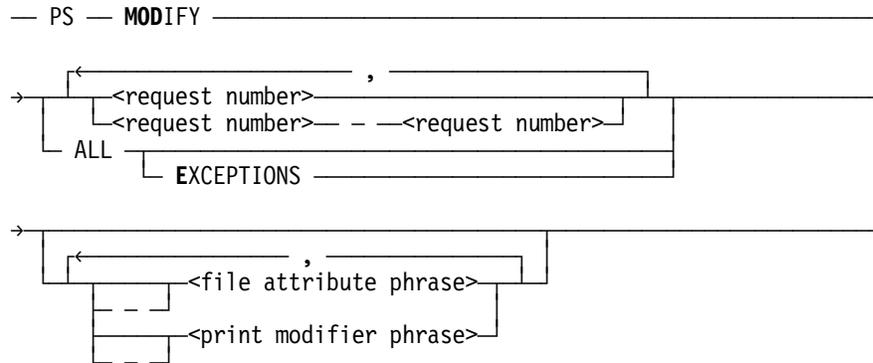
<entity reference declaration>



Sample Input	Explanation
ENTITY REFERENCE ADVISOR1 (INSTRUCTOR)	The required item ENTITY REFERENCE is followed by the variable ADVISOR1 and the variable INSTRUCTOR. The parentheses are required.
ENTITY REFERENCE ADVISOR1 (INSTRUCTOR), ADVISOR2 (ASST_INSTRUCTOR)	Because the diagram contains a loop, the pair of variables can be repeated any number of times.

Example 5

<PS MODIFY command>



Sample Input	Explanation
PS MODIFY 11159	The constants PS and MODIFY are followed by the variable 11159, which is a request number.
PS MODIFY 11159,11160,11163	Because the diagram contains a loop, the variable 11159 can be followed by a comma, the variable 11160, another comma, and the final variable 11163.
PS MOD 11159-11161 DESTINATION = "LP7"	The constants PS and MODIFY are followed by the user-selected variables 11159-11161, which are request numbers, and the user-selected variable DESTINATION = "LP7", which is a file attribute phrase. Note that the constant MODIFY has been abbreviated to its minimum allowable form.
PS MOD ALL EXCEPTIONS	The constants PS and MODIFY are followed by the user-selected constants ALL and EXCEPTIONS.

Appendix G

Related Product Information

The following documents provide information that is directly related to the primary subject of this guide.

MCP/AS ALGOL Programming Reference Manual, Volume 1: Basic Implementation (8600 0098)

This manual describes the basic features of the Extended ALGOL programming language. This manual is written for programmers who are familiar with programming concepts.

MCP/AS C Programming Reference Manual, Volume 1: Basic Implementation (8600 2268)

This manual describes the C programming language. It includes descriptions of syntax, status messages, the preprocessor, compiling system, binding system, and run-time library. Extensions such as compiler control options and the library facility are also documented. This manual is written for systems and applications programmers.

MCP/AS Data Communications Protocols Installation and Implementation Guide (8600 0486)

This guide describes the purpose of protocols and procedures for installing protocols. It also provides reference material useful in interpreting dumps associated with data communications data link processors (DCDLPs), enhanced data communications data link processors (EDCDLPs), and data communications adapters (DCAs). This guide is written for system analysts and operations specialists who work with data communications.

MCP/AS Help Utility Operations Guide (8600 0510)

This guide describes how to embed Help Utility commands in text to create a helpbook source file and how to process this source file with the Help Utility. This guide is written for inexperienced and experienced computer users, and writers or translators of online help text.

MCP/AS Interactive Datacomm Configurator (IDC) Operations Guide (8600 1880)

This guide explains how to use IDC, a menu-driven utility used to define and modify data communications networks. It provides information on configuring a data communications network using the IDC menu system and basic constructs, and provides reference information about the commands and attributes. This guide is written for individuals who have a basic knowledge of data communications concepts, but who might not know the physical characteristics of hardware devices in the network.

MCP/AS Interactive Menugraph Generator (IMG) Operations Guide (8600 0411)

This guide explains how to use and run IMG, a software tool for the design and modification of Menu-Assisted Resource Control (MARC), Interactive Datacomm Configurator (IDC), BNA Version 2, and IMG screens and forms. This guide is written for users who want to customize these screens for their system.

MCP/AS Menu-Assisted Resource Control (MARC) Operations Guide (8600 0403)

This guide provides an overview of MARC, a description of the menu structure, and information on how to use help text, commands, security features, and Transaction Server windows from MARC. The guide also explains how to run programs from MARC, how to customize MARC to meet user needs, and how to use MARC in a multinational environment. This guide is written for a wide audience, ranging from experienced system administrators to end users with no previous knowledge of MARC.

MCP/AS Message Translation (MSGTRANS) Utility Operations Guide (8600 0106)

This guide describes how to use the Message Translation Utility (MSGTRANS) to translate compiled program messages from any natural language to any other natural language. It provides complete instructions for running and using the screen and batch interfaces of MSGTRANS. This guide is written for programmers and translators who create and translate program messages in a MultiLingual System (MLS) environment.

MCP/AS Network Definition Language II (NDLII) Programming Reference Manual (8807 7011)

This manual documents the high-level programming and definition language used to describe a data communications network. It gives a brief overview and functional description of the data comm system and provides a complete description of the syntax and semantics of all language components and compiler options of NDLII. This manual is written for experienced data communications programmers.

MCP/AS Screen Design Facility Plus (SDF Plus) Installation and Operations Guide (8600 0262)

This guide explains how to use SDF Plus to create and maintain a user interface. It gives specific instructions for installing SDF Plus, using the SDF Plus forms, and installing and running a user interface created with SDF Plus.

***MCP/AS Screen Design Facility Plus (SDF Plus) Technical Overview
(8600 0072)***

This overview provides the conceptual information needed to use SDF Plus effectively to create user interfaces.

***MCP/AS TCP/IP Distributed Systems Services (DSS) Operations Guide
(8807 6385)***

This guide explains how to use most of the TCP/IP services provided by the DSS products. These products include FTP, Telnet, and Domain Name Services (DNS). The guide also explains how to use the TCP/IP DSS debugging tools. This guide is written for system planners, system programmers, application programmers, and computer operators who use TCP/IP DSSs.

For information on the Simple Mail Transfer Protocol (SMTP) used by the Mail System, see the *Mail System Operations Guide* and the *Mail System Administration Guide*.

***Unisys e-@ction ClearPath Enterprise Servers ClearPath MCP Release 6.0
Migration Guide (8999 7415)***

This guide provides a detailed overview of some of the new features and enhancements available with the ClearPath systems. Specifically, it describes the new features and enhancements that are installed in the MCP environment. It also describes changes in software behavior between the last software release and the current release. The document includes information on software migration, feature matrixes, special installation instructions for specific products, deimplementations and deimplementation warning messages, and known problems. This document is written for people who want to know how upgrading to the new release will affect their existing software.

***Unisys e-@ction ClearPath Enterprise Servers File Attributes Programming
Reference Manual (8600 0064)***

This manual contains information about each file attribute and each direct I/O buffer attribute. The manual is written for programmers and operations personnel who need to understand the functionality of a given attribute. The *I/O Subsystem Programming Guide* is a companion manual.

***Unisys e-@ction ClearPath Enterprise Servers Security Administration
Guide (8600 0973)***

This guide describes system-level security features and suggests how to use them. It provides administrators with the information necessary to set and implement effective security policy. This guide is written for system administrators, security administrators, and those responsible for establishing and implementing security policy.

***Unisys e-@ction ClearPath Enterprise Servers System Commands
Operations Reference Manual (8600 0395)***

This manual gives a complete description of the system commands used to control system resources and work flow. This manual is written for systems operators and administrators.

Unisys e-@ction Transaction Server for ClearPath MCP Programming Guide (8600 0650)

This guide explains how to write online, interactive, and batch application programs that run under Transaction Server. This guide is written for experienced applications programmers with knowledge of data communication subsystems.

Index

A

a

- time control character, 3-4, 3-10
- accessing
 - CENTRALSUPPORT library
 - messages, 6-20
 - convention definitions, using CNV_INFO procedure, 6-22
 - libraries for SDF Plus products, 5-6
 - output message arrays, 6-19, B-3
- Adapter Control
 - editing associated mapping table, 7-12
 - editing control characters in, 7-6
- adapting PROMs for code extension, 7-5
- ADDCNV—Add Convention screen
 - adding custom conventions, 4-19
- ADDS, form library files for, 5-27
- ALGOL
 - example library containing message arrays, B-1
 - MESSAGESEARCHER statement, 6-19
 - MLSACCEPT statement, 6-19
 - MLSDISPLAY statement, 6-19
 - MLSTRANSLATE statement, 6-19
 - OUTPUTMESSAGE ARRAY
 - declarations, 6-17
 - processing input and output messages, 6-19
- ALGOL translate table, 6-6
- alphabetic control characters
 - in date and time templates, 3-4
- ALT_MONTH
 - date element, 3-11, 3-13
- Appendix D
 - Help Text Available for Localization, D-1

- application programs
 - accessing the internationalization features from, 6-1
 - CENTRALSUPPORT library procedures (See CENTRALSUPPORT library procedures)
 - creating messages for, 6-16
 - defined, 1-4
 - not using the internationalization features, 6-1
 - processing input and output messages, 6-19
 - supporting natural languages, 6-16
- application systems (See application programs)
- Arabic20 ccsversion
 - definitions, 14-3
- Arabic20EBCDIC
 - code table (8CS11.01), 10-2
 - coded character set, 14-3
- Arabic20ISO
 - code table (8CS11.00), 10-3
- Argentina convention definitions, 15-2
- ASCII
 - data transfers, performing, 7-4
- ASCII code table (8CS1.00), 10-12
- ASCII double-octet characters, C-10
- ASCII mapping table
 - editing, 7-12
- ASERIESEBCDIC
 - code table (8CS1.01), 10-17
 - coded character set, 14-5
- ASeriesNative ccsversion
 - definitions, 14-5
 - number, 14-1
 - system default, 4-2

- ASeriesNative convention
 - default values
 - date control characters, 3-13
 - monetary and numeric control characters, 3-22
 - time control characters, 3-15
 - template definitions, 15-3
- ASTASCIIGB2312 code table (816CS00.00), 13-5, 13-11
- ASTASCIIKSC5601 code table (816CS00.00), 13-6
- ASTASCIUUTL code table (816CS00.00), 13-7
- asterisks (*) in help text
 - to identify keywords, 5-11
 - to identify standard defines, 5-14
- ASTUTL code table (16CS00.00), 13-8
- ASUTL code table (16CS00.00), 13-9
- Australia convention definitions, 15-4
- AUTONEXTKEYWORD command, in products using SDF Plus, 5-13

B

- Belgium convention definitions, 15-5
- blanks
 - between control characters, 3-2
 - date and time input arrays, 3-3
 - in literal text, 3-2
 - leading, 3-7
 - null template, 3-2
 - numeric and monetary templates, 3-17
 - output values, 3-5
- BOOK FILETITLE command, using in help text, 5-9
- brackets
 - use in conventions, 3-16
 - using angle brackets to identify keywords, 5-11
 - using square brackets in conventions, 3-22
- Brazil convention definitions, 15-6
- Brazilian ccsversion, 14-1, 14-6
- business conventions, 1-5
 - compiler support, 2-5
 - designing programs for, 6-2
 - SDF Plus support, 6-33
 - support for, 6-20

C

- c
 - currency control character, 3-19
 - date control character, 3-4, 3-10
 - monetary control character, 3-23
- C locales, 8-5
- C programming language
 - accessing message arrays, 6-19, B-3
- c[]
 - monetary control character, 3-19, 3-23
- CAD, 1-11
- CanadaEBCDIC ccsversion, 14-1, 14-9
- CanadaGP ccsversion, 14-1, 14-12
- CANDE (*See* Command and Edit)
- CANSuppEBCDIC
 - coded character set, 14-12
- CANSuppLEBCDIC code table (8CS51.01), 10-26
- CANSuppIIISO code table (8CS51.00), 10-27
- CCSINFO procedure, 6-5
- CCSTOCCS_TRANS_TABLE procedure, 6-6
- ccsversion definitions
 - Arabic 20, 14-3
 - ASeriesNative, 2-2, 14-5
 - Brazilian, 14-6
 - CanadaEBCDIC, 14-9
 - CanadaGP, 14-12
 - Czechoslovakia, 14-16
 - France, 14-19
 - Hungarian, 14-22
 - LatinGreek, 14-25
 - Norway, 14-30
 - Polish, 14-33
 - Romanian, 14-36
 - Russian, 14-39
 - Spanish, 14-43
 - Swedish1, 14-46
 - Swiss, 14-49
 - Turkish, 14-53

- ccsversion
 - available on host system, 6-3
 - CENTRALSUPPORT library, 2-3
 - default, 4-2
 - defined, 1-5
 - halt/load requirement, 4-9
 - identifying in application program, 6-3
 - in current release (table), 14-1
 - information in, 1-11
 - obtaining information about, in an
 - application program, 6-5
 - processing data according to, 6-9
 - SDF Plus support, 6-33
 - setting the system default
 - MARC screens methods, 4-15
 - SYSTEMOPTION command
 - method, 4-10
- CCSVSN_NAMES_NUMS procedure, 6-3
- CED, 1-11
- CENTRALSTATUS procedure, 6-3, 6-21
- Centralsupport architecture
 - defined, 2-1
 - SYSTEM/CONVENTIONS file, 2-2
- CENTRALSUPPORT library
 - defined, 2-3
 - function of, 2-3
 - initializing, A-23
- CENTRALSUPPORT library procedures
 - accessing, 6-20
 - adding custom conventions, 6-31
 - coded character set, 6-3
 - comparing text strings, 6-13
 - conventions, 6-20
 - deleting custom conventions, 6-31
 - formatting
 - dates, 6-24
 - times, 6-26
 - header messages, A-2
 - identifying
 - ccsversion on host, 6-3
 - coded character set on host, 6-3
 - natural languages available, 6-16
 - system defaults for MLS options, 6-3
 - mapping
 - coded character sets, 6-6
 - Complex procedure, 6-7
 - data from one coded character set to another, 6-6
 - modifying a convention, 6-31
 - obtaining information about
 - ccsversion, 6-5
 - coded character set, 6-5
 - positioning characters, 6-15
 - processing
 - control characters, 6-6
 - data, 6-6
 - data based on ccsversion, 6-9
 - graphic characters, 6-6
 - processing text strings, 6-6
 - result messages, A-1
 - sorting text strings, 6-13
 - character
 - limitation for code extension, 7-12
 - special, in help text, 5-13
 - used with code extension, 7-5
 - character advancement direction (CAD), 1-11
 - character escapement direction (CED), 1-11
 - character sets
 - coded (See coded character sets)
 - defined, 1-5
 - characters per line, 1-17
 - characters, 8-bit
 - obtaining decimal numbers for, 9-15
 - obtaining hexadecimal numbers for, 9-15
 - Chile convention definitions, 15-7
 - China convention definitions, 15-8
 - Client Access Services
 - configuring for internationalization, 7-22
 - translating files, 7-23
 - CNV_ADD procedure, 6-31
 - CNV_CONVERTCURRENCY procedure, 6-28
 - CNV_CONVERTDATE procedure, 6-24
 - CNV_CONVERTNUMERIC procedure, 6-28
 - CNV_CONVERTTIME procedure, 6-26
 - CNV_CURRENCYEDIT procedure, 6-28, 6-29
 - CNV_CURRENCYEDITTMP procedure, 6-29
 - CNV_DELETE procedure, 6-31
 - CNV_DISPLAYMODEL procedure, 6-24, 6-26
 - CNV_FORMATDATE procedure, 6-24
 - CNV_FORMATDATETMP procedure, 6-24
 - CNV_FORMATTIME procedure, 6-26
 - CNV_FORMATTIMETMP procedure, 6-26
 - CNV_FORMSIZE procedure, 6-30
 - CNV_INFO procedure, 6-22
 - CNV_MODIFY procedure, 6-31
 - CNV_MSGS array
 - translating date elements, 3-13
 - translating time elements, 3-15

- CNV_NAMES procedure, 6-21
- CNV_SYMBOLS procedure, 6-22
- CNV_SYSTEMDATETIME procedure, 6-25, 6-27
- CNV_SYSTEMDATETIMETMP procedure, 6-25, 6-27
- CNV_TEMPLATE procedure, 6-23
- CNV_VALIDATENAME procedure, 6-21
- COBOL74
 - accessing message arrays, 6-19
 - example of, B-5
- COBOL85
 - example of, B-6
- code extension
 - adapting PROMs for, 7-5
 - altering characters requesting, 7-21
 - character limitation, 7-12
 - configuring terminals for, 7-5
 - control characters, altering, 7-6
 - control characters, using with, 7-5
 - describing, 7-7
 - firmware requirements, 7-5
 - hexadecimal values to avoid, 7-12
 - introducing, 7-4
 - limitations, 7-5
 - receiving data, role of
 - EDCDLP or DCHA, 7-9
 - terminal or printer, 7-11
 - requesting, 7-21
 - sending data from a device to the mainframe, 7-7
 - sending data from the mainframe to a device, 7-10
 - setting the station variable, 7-21
 - settings for, 7-21
 - special characters used for, 7-5
 - transmitting data, role of
 - EDCDLP or DCHA, 7-10
 - terminal or printer, 7-8
 - using
 - one coded character set mapping table, 7-12
 - to transfer data, 7-4
 - two mapping tables, 7-14
 - with double-octet characters, 7-22
 - with Kanji, 7-22
- CODE_EXT_MODE station variable
 - setting for code extension, 7-21
- coded character set mapping table
 - code extension use, 7-12
 - localizing ODT with, 4-7
 - many-to-one mapping, 9-5
- coded character set mapping tables, 1-5
 - (See also mapping table)
- CENTRALSUPPORT library procedures, 6-6
 - mapping by ccsversion, 6-10
 - returning a mapping table, 6-6
 - returning a mapping table for a ccsversion, 6-11
- double-byte, 9-10
- eight-bit, 9-6
- hexadecimal values, 1-9
- identifying, 9-5
- INTMODE and EXTMODE values, 1-9
- mixed, multibyte, 9-11
- pairing of, 9-5
- coded character set tables
 - Arabic20EBCDIC (8CS11.01), 10-2
 - Arabic20ISO (8CS11.00), 10-3
 - ASCII (8CS1.00), 10-12
 - ASERIESEBCDIC (8CS1.01), 10-17
 - ASTASCIIGB2312 (816CS00.00), 13-5, 13-11
 - ASTASCIIKSC5601 (816CS00.00), 13-6
 - ASTASCIIUTL (816CS00.00), 13-7
 - ASTUTL (16CS00.00), 13-8
 - ASUTL (16CS00.00), 13-9
 - CANSupplISO (8CS51.00), 10-26
 - CANSupplISO(8CS51.00), 10-26
 - CODEPAGE1250 (8CS58.03), 11-65
 - CODEPAGE1251 (8CS57.03), 11-74
 - CODEPAGE1252 (8CS60.03), 11-83
 - CODEPAGE1253 (8CS00.00), 11-92, 11-101
 - CODEPAGE1254 (8CS00.00), 11-101
 - CODEPAGE1256 (8CS00.00), 11-110
 - CODEPAGE437 (8CS49.00), 11-3
 - CODEPAGE850 (8CS49.00), 11-11
 - CODEPAGE851 (8CS43.00), 11-20
 - CODEPAGE852 (8CS27.03), 11-29
 - CODEPAGE857 (8CS00.00), 11-38

- CODEPAGE864 (8CS00.00), 11-48
- CODEPAGE932 (816CS00.00), 13-10
- CODEPAGE949 (816CS00.00), 13-12
- CODEPAGE950 (816CS00.00), 13-13
- EBCDIC (8CS1.01), 10-17
- EBCDICGB2312 (816CS00.00), 13-14
- EBCDICKSC5601 (816CS00.00), 13-15
- EBCDICUTL (816CS00.00), 13-16
- EUCJp (816CS00.00), 13-17
- FrenchArabicEBCDIC (8CS46.01), 12-2
- FrenchArabicISO (8CS46.00), 12-2
- HungarianBTOS (8CS00.00), 12-11
- IBM297 (8CS62.01), 12-22
- IBMSwedenEBCDIC (8CS00.00), 12-30
- JapanEBCDICJBIS8 (816CS00.00), 13-19
- JapanV24 (8CS00.00), 12-47
- JapanV24JBIS8 (816CS00.00), 13-21
- JBIS7 (16CS00.00), 13-23
- JBIS8 (16CS00.00), 13-24
- JISASCIJBIS7 (816CS00.00), 13-25
- Latin1EBCDIC (8CS26.01), 12-66
- Latin2EBCDIC (8CS27.01), 12-76
- Latin2ISO (8CS27.00), 12-77
- Latin5EBCDIC (8CS29.01), 12-86, 12-96
- Latin5ISO (8CS29.00), 12-86, 12-96
- LatinCyrillicEBCDIC (8CS55.01), 12-106
- LatinCyrillicISO (8CS55.00), 12-107
- LatinGreekBTOS (8CS41.00), 12-116
- LatinGreekEBCDIC (8CS30.01), 12-125
- LatinGreekISO (8CS30.00), 12-125
- LetsJ (816CS00.00), 13-27
- LocalEBCDIC (8500.00), 12-135
- MACROMAN (8CS61.03), 12-144
- NorwayBTOS (8CS50.00), 12-153
- UCS2 (16CS00.00), 13-29
- UCS2NT (16CS00.00), 13-30
- coded character sets
 - available on host system, 6-3
 - choosing, 1-8
 - defined, 1-5
 - effect on code extension, 7-12
 - identifying, 9-2
 - identifying in an application program, 6-3
 - information in, 1-8
 - mapping data, 6-6
 - multiple sets, effect on code extension, 7-14
 - obtaining information about, in an application program, 6-5
 - reading double-byte and mixed, multibyte tables, 13-3
 - SDF Plus support, 6-32
 - table of sets in release, 9-2
 - types of characters, 1-8
- CODEPAGE1250 code table (8CS58.03), 11-65
- CODEPAGE1251 code table (8CS57.03), 11-74
- CODEPAGE1252 code table (8CS60.03), 11-83
- CODEPAGE1253 code table (8CS00.00), 11-92, 11-101
- CODEPAGE1254 code table (8CS00.00), 11-101
- CODEPAGE1256 code table (8CS00.00), 11-110
- CODEPAGE437 code table (8CS49.00), 11-3
- CODEPAGE850 code table (8CS49.00), 11-11
- CODEPAGE851 code table (8CS43.00), 11-20
- CODEPAGE852 code table (8CS27.03), 11-29
- CODEPAGE857 code table (8CS00.00), 11-38
- CODEPAGE864 code table (8CS00.00), 11-48
- CODEPAGE932 code table (816CS00.00), 13-10
- CODEPAGE949 code table (816CS00.00), 13-12
- CODEPAGE950 code table (816CS00.00), 13-13
- collating, 6-14
 - role of ccsversion, 4-28
 - strings, 8-3, 8-25
 - text comparisons, 6-13
- Colombia convention definitions, 15-9
- Command and Edit (CANDE)
 - command
 - CHANGE, 5-29, 8-15
 - CONVENTION, 4-27
 - LANGUAGE, 4-23
 - compiling *SYMBOL/SOURCENDLII file, 7-16
 - convention for, 4-22
 - restrictions, 4-24
 - session, 2-4, 4-21, 4-22
 - starting
 - application from, 5-6
 - applications from, 4-22
 - MSGTRANS from, 8-8
 - SDF Plus from, 5-7

- commands
 - CANDE
 - CHANGE, 5-29
 - CONVENTION, 4-27
 - LANGUAGE, 4-23
 - SECURITY, 5-29
 - CONVENTION
 - setting conventions, 4-27
 - help text
 - AUTONEXTKEYWORD, 5-13
 - KEYWORD, 5-11, 5-13
 - SECTION, 5-11
 - LANGUAGE
 - setting natural languages, 4-23
 - MARC
 - MENU, 5-4
 - required for help books
 - using Help Utility, 5-9
 - using SDF Plus Help utility, 5-9
 - system
 - AX, 4-6
 - ID, 7-17
 - MP, 5-29
 - OL, 4-7
 - SYSTEMOPTIONS, 4-7, 4-10
- COMPARE_TEXT_USING_ORDER_INFO
 - procedure, 6-13
- comparing text strings, 6-13
- compiler (See programming languages)
- Complex procedure, 6-7
 - example mapping scenarios, C-10
 - implementation strategies, C-8
 - continuous stream mode, C-8
 - discrete mode, C-8
 - guidelines, C-9
 - terminating calls, C-10
 - Option parameter, 6-7, C-7
 - parameters, C-4
 - specification for different programming languages, C-2
 - State array parameter, C-7
 - using, C-1
- COMS (See Transaction Server)
- configuring
 - system initialization, 4-6
 - terminals for code extension, 7-5
- control characters
 - alphabetic, 3-4
 - altering for code extension, 7-6
 - date elements, 3-13
 - defined, 1-8, 1-18
 - for date and time templates, 3-3, 3-9
 - for monetary and numeric templates, 3-16, 3-22
 - general characteristics in convention templates, 3-2
 - numeric, 3-5
 - processing in an MLS environment, 6-6
 - table of numeric and monetary control characters, 3-18
 - time elements, 3-15
 - translating predetermined text, 3-24
 - using with code extension, 7-5
- CONVDF—View/Add/Modify/Delete a Convention screen, 4-18
- convention, 1-4 (See also convention definitions; convention definition elements)
 - adding using MARC screens, 4-19
 - business and cultural, 6-20
 - CENTRALSUPPORT library
 - procedures, 6-20
 - changes to
 - at system level, 4-18
 - using MARC screens, 4-16
 - corporate standards, 15-1
 - creating, 3-1
 - creating, 6-20
 - custom (See custom convention definitions)
 - date display model for, 6-24, 6-26
 - deciphering, 3-1
 - deleting, 6-31
 - elements, 1-6
 - for keywords, 5-13
 - installing updates to the SYSTEM/CONVENTIONS file, 4-3
 - listing using the CNV_NAMES procedure, 6-21
 - modifying with MARC screens, 4-19
 - rules for names of, 1-16
 - selection priorities, 2-5

- setting
 - session level, 4-25
 - system default with SYSTEMOPTION command, 4-10
 - system level, 4-9
- templates, defined, 3-1
- time display model for, 6-24, 6-26
- translating predetermined text, 3-24
- use of control characters in, 1-18
- viewing
 - at session level, 4-25
 - with MARC screens, 4-18
- CONVENTION command
 - setting conventions, 4-27
- convention definition elements, 1-15
 - default page length, 1-17
 - default page width, 1-17
 - fractional monetary fractional digits, number of, 1-17
 - international currency notation, 1-17
 - international monetary fractional digits, number of, 1-17
 - long date templates, 1-16
 - long time template, 1-16
 - monetary symbols, 6-22
 - monetary templates, 1-16
 - number of maximum integer monetary digits, 1-17
 - numeric date templates, 1-16
 - numeric symbols, 6-22
 - numeric templates, 1-16
 - numeric time templates, 1-16
 - short date templates, 1-16
- convention definitions, 1-4 (*See also* convention)
 - Argentina, 15-2
 - ASeriesNative, 15-3
 - Australia, 15-4
 - Belgium, 15-5
 - Brazil, 15-6
 - Chile, 15-7
 - China, 15-8
 - Colombia, 15-9
 - CostaRica, 15-10
 - Croatia, 15-11
 - Cyrillic, 15-12
 - Czechoslovakia, 15-13
 - Denmark, 15-14
 - Egypt, 15-15
 - EnglishCanada, 15-16
 - EuropeanStandard, 15-17
 - FranceBureautique, 15-18
 - FranceListing, 15-19
 - FrenchCanada, 15-20
 - Greece, 15-21
 - HongKong, 15-22
 - Hungary, 15-23
 - Italy, 15-24
 - Japan1, 15-25
 - Japan2, 15-26
 - Jordan, 15-27
 - Kenya, 15-28
 - Kuwait, 15-29
 - Lebanon, 15-30
 - Malaysia, 15-31
 - Mexico, 15-32
 - Netherlands, 15-33
 - NewZealand, 15-34
 - Nigeria, 15-35
 - Norway, 15-36
 - Peru, 15-37
 - Philippines, 15-38
 - Poland, 15-39
 - Romania, 15-40
 - SouthAfrica, 15-41
 - Spain, 15-42
 - StandardTurkey, 15-43
 - Sweden, 15-44
 - Switzerland, 15-45
 - Taiwan, 15-46
 - UAE, 15-47
 - UnitedKingdom1, 15-48
 - UnitedKingdom2, 15-49
 - Venezuela, 15-50
 - Yugoslavia, 15-51
 - Zimbabwe, 15-52
- convention names
 - listing using the CNV_NAMES procedure, 6-21
- CONVENTION task attribute, 4-28
- CONVENTION-SET SESSION CONVENTION menu
 - selecting conventions, 4-26

- converting
 - dates to YYMMDD format, 6-24
 - digits to real numbers, 6-28
 - monetary digits to real numbers, 6-28
 - times to HHMMSSPPPP format, 6-26
 - copyright information, in help text, 5-25
 - corporate standards, for conventions, 15-1
 - CostaRica convention definitions, 15-10
 - Croatia convention definitions, 15-11
 - CSINT_MSG output message array, A-23
 - cultural conventions, 1-5
 - compiler support, 2-5
 - designing programs for, 6-2
 - SDF Plus support, 6-33
 - support for, 6-20
 - currency
 - control character, 3-19
 - restrictions, 3-16
 - control character, 3-23
 - international currency notation, 1-17
 - international currency symbol, 6-22
 - currency symbol used as decimal symbol, 3-23
 - custom convention definitions, 2-2
 - adding, using CNV_ADD procedure, 6-31
 - creating, 3-1
 - deleting, using CNV_DELETE procedure, 6-31
 - migrating to a new SYSTEM/CONVENTIONS file, 4-3
 - modifying, using CNV_MODIFY procedure, 6-31
 - setting at session level, 4-25
 - viewing at session level, 4-25
 - Cyrillic convention definitions, 15-12
 - Czechoslovakia
 - ccsversion, 14-1, 14-16
 - convention definitions, 15-13
- D**
- d
 - date control character, 3-5, 3-10
 - decimal control character, 3-19
 - monetary and numeric control character, 3-23
 - d[]
 - monetary and numeric control character, 3-19, 3-23
 - d[c[]]
 - monetary and numeric control character, 3-23
 - data classes
 - definition of, 6-9
 - determining, 2-3
 - graphic characters in, 1-13
 - requirements, 1-14
 - types of, 1-5
 - use of, 6-9
 - VSNINFO procedure, 6-5
 - data communications host adapter, 7-5
 - data processing in an MLS environment, 6-6
 - data transferal using code extension, 7-4
 - date control characters, 3-9
 - elements, 3-13
 - examples, 3-8
 - specifying alphabetic components, 3-4
 - specifying numeric components, 3-5
 - date display model for convention, 6-24, 6-26
 - date elements
 - ALT_MONTH, 3-11, 3-13
 - DAY_OF_MONTH, 3-10, 3-13
 - DAY_OF_WEEK, 3-13
 - DAY_OF_YEAR, 3-10, 3-13
 - DOM_PREFIX, 3-10, 3-13
 - DOM_SUFFIX, 3-10, 3-13
 - DOW_PREFIX, 3-12, 3-13
 - in convention definitions
 - long date templates, 1-16
 - numeric date templates, 1-16
 - short date templates, 1-16
 - LONG_DOW, 3-12, 3-14
 - LONG_MONTH, 3-11, 3-14
 - MONTH_OF_YEAR, 3-11, 3-14
 - MONTH_PREFIX, 3-11, 3-14
 - SHORT_DOW, 3-12
 - SHORT_DOW, 3-14
 - SHORT_MONTH, 3-11, 3-14
 - YEAR, 3-12, 3-14
 - YEAR_PREFIX, 3-12, 3-14
 - YEAR_SUFFIX, 3-12, 3-14

- date templates
 - processing, 3-3
 - undefined, 3-2
 - dates
 - converting to YYMMDD format, 6-24
 - formatting according to a convention, 6-24
 - formatting according to a template, 6-24
 - formatting the system date
 - using a template, 6-25, 6-27
 - using convention definition, 6-25, 6-27
 - DAY_OF_MONTH
 - date element, 3-10, 3-13
 - DAY_OF_WEEK
 - date element, 3-13
 - DAY_OF_YEAR
 - date element, 3-10, 3-13
 - DCHA, 7-5
 - DCHA and code extension
 - using, 7-5
 - decimal control character, 3-19
 - decimal number, obtaining for an 8-bit character, 9-15
 - declaring the CODE_EXT_MODE variable, 7-21
 - default
 - ccsversion, 1-11
 - coded character set, 1-8
 - displaying, 4-11
 - effect of changing, 4-9
 - language, 4-2
 - on release media, 15-3
 - overriding at session level, 2-4
 - page length, determining, 6-30
 - page width, determining, 6-30
 - deleting custom conventions
 - using CENTRALSUPPORT library procedure, 6-31
 - Denmark convention definitions, 15-14
 - digits
 - converting to real numbers, 6-28
 - directory names
 - identifying for SDF Plus libraries, 5-26
 - identifying in help text symbol file, 5-10
 - disclaimer information, dealing with in help text, 5-25
 - discrete mode, Option values for, C-8
 - display models for dates and times, 6-24, 6-26
 - DOM_PREFIX
 - date element, 3-10, 3-13
 - DOM_SUFFIX
 - date element, 3-10, 3-13
 - double octet
 - data transfers, 7-22
 - delimiters, 10-17
 - SDF Plus support, 6-32
 - double-byte coded character set mapping tables, 9-10
 - double-byte coded character set tables, 13-1
 - double-octet characters, ASCII, C-10
 - double-octet characters, EBCDIC, C-10
 - DOW_PREFIX
 - date element, 3-12, 3-13
 - dump, A-5
- ## E
- e
 - date control character, 3-4, 3-10
 - enclosure control character, 3-20
 - monetary and numeric control character, 3-23
 - e[]
 - monetary and numeric control character, 3-20, 3-23
 - EBCDIC
 - coded character set, 14-5
 - EBCDIC double-octet characters, C-10
 - EBCDICGB2312 code table (816CS00.00), 13-14
 - EBCDICKSC5601 code table (816CS00.00), 13-15
 - EBCDICUTL code table (816CS00.00), 13-16
 - EDCDLP, 7-5
 - code extension, using, 7-5
 - edit environment, establishing languages for SDF Plus, 5-7
 - editing
 - Adapter Control, 7-6
 - date, 1-16
 - definition, 3-18
 - groups, 3-22
 - items in convention templates, 3-1
 - processing direction, 3-17
 - SDF Plus, 5-7
 - time, 1-16

- Editor
 - using to edit associated mapping table, 7-14
 - EDO character
 - using with code extension, 7-22
 - Egypt convention definitions, 15-15
 - eight-bit coded character set mapping tables, 9-6
 - elements in convention definitions, 1-15
 - accessing, 6-23
 - embedded keywords, in help text, 5-14
 - empty template, 3-2
 - enclosure control character, 3-20
 - encoding number, 1-9
 - end-of-double-octet (EDO) character
 - using with code extension, 7-22
 - end-of-Kanji (EOK) character
 - using with code extension, 7-22
 - EnglishCanada convention definitions, 15-16
 - enhanced data communications data link processor (See EDCDLP)
 - Enterprise Database Server compatibility, 2-2
 - EOK character
 - using with code extension, 7-22
 - equivalent ordering information, 6-14
 - ESC SI character
 - using with code extension, 7-5
 - ESC SO character
 - using with code extension, 7-5
 - escape shift in character
 - using with code extension, 7-5
 - escape shift out character
 - using with code extension, 7-5
 - escapement
 - character advancement direction (CAD), 1-11
 - character escapement direction (CED), 1-11
 - line advancement direction (LAD), 1-11
 - text line orientation (TLO), 1-11
 - establishing languages
 - for MARC, 5-4
 - for SDF Plus, 5-6
 - for Unisys products, 5-29
 - ET terminals
 - reconfiguring for code extension, 7-5
 - EUCJp code table (816CS00.00), 13-17
 - EuropeanStandard convention
 - definitions, 15-17
 - exclamation marks (!)
 - use with control characters, 3-2
 - EXT_ESC_CO
 - setting for code extension, understanding, 7-21
 - EXT_NONE
 - setting for code extension, understanding, 7-21
 - EXT_SO
 - setting for code extension, understanding, 7-21
 - EXTMODE file attribute
 - using with coded character sets, 1-9
- ## F
- f
 - time control character, 3-4, 3-10
 - file attributes
 - EXTMODE, 1-9
 - INTMODE, 1-9
 - firmware required for code extension, 7-5
 - form libraries
 - access hierarchy, 5-6
 - identifying the directory name, 5-26
 - identifying the library name, 5-26
 - identifying the type of library, 5-26
 - localizing, 5-5
 - localizing with SDF Plus, 5-26
 - formatting
 - dates
 - according to a convention, 6-24
 - according to a template, 6-24
 - items in convention templates, 3-1
 - monetary values
 - using a template, 6-29
 - using convention definition, 6-28, 6-29
 - system date
 - using a template, 6-25, 6-27
 - using convention definition, 6-25, 6-27
 - system time
 - using a template, 6-25, 6-27
 - using convention definition, 6-25, 6-27
 - times
 - according to a convention, 6-26
 - according to a template, 6-26

fractional digits, as fractional monetary digits, 1-17
 France ccsversion, 14-1, 14-19
 FranceBureautique convention definitions, 15-18
 FranceListing convention definitions, 15-19
 FrenchArabicEBCDIC code table (8CS46.01), 12-2
 FrenchArabicISO code table (8CS46.00), 12-2
 FrenchCanada convention definitions, 15-20
 FTP
 configuring for internationalization, 7-3
 receiving from remote destination, 7-3
 representation types, 7-3
 sending to remote destination, 7-3

G

GET_CS_MSG procedure, 6-20
 graphic characters
 defined, 1-8
 positioning, 6-15
 processing in an MLS environment, 6-6
 Greece convention definitions, 15-21

H

h
 time control character, 3-5, 3-10
 halt/load
 effect on system default ccsversion, 4-9
 effect on system default convention, 2-2
 effect on system default language, 4-9
 header messages, A-2
 help books
 naming convention for products using SDF Plus, 5-10
 required commands for products using Help Utility, 5-9
 products using SDF Plus Help utility, 5-9
 using separate short and long help text books, 5-12

help libraries
 identifying the directory name, 5-26
 identifying the library name, 5-26
 identifying the type of library, 5-26
 localizing with SDF Plus, 5-26
 help text
 available for translation, D-1
 embedded keywords in, 5-14
 required information contained in the symbol file, 5-9
 short help, 5-14
 text in, 5-14
 translating
 .BOOK FILETITLE command, 5-9
 .DIRECTORY command, 5-10
 .KEYWORD commands, 5-11
 .LANGUAGE command, 5-9
 .SECTION commands, 5-11
 asterisks (*), 5-14
 copyright information, 5-25
 disclaimer information, 5-25
 embedded keywords, 5-14
 KEYWORD commands, 5-11
 SECTION commands, 5-11
 text and embedded keywords, 5-14
 trademark information, 5-25
 help utilities, types of, 5-8
 hexadecimal values
 avoiding use of in code extension, 7-12
 coded values, 1-9
 input messages, 6-19
 obtaining for an 8-bit character, 9-15
 representing nonprintable symbols, 3-16
 HOD_12_HRCL
 time element, 3-10, 3-15
 HongKong convention definitions, 15-22
 HOUR_OF_DAY
 time element, 3-12, 3-15
 HOUR_SUFFIX
 time element, 3-10, 3-15
 Hungarian ccsversion, 14-22
 HungarianBTOS (8CS00.00), 12-11
 Hungary convention definitions, 15-23

I

i
time control character, 3-4, 3-10

IBM297 code table (8CS62.01), 12-22

IBMSwedenEBCDIC code table (8CS00.00), 12-30

IDC (See Interactive Datacomm Configurator)

IEnumXlateCCSes
interface, 16-19, 16-21
methods, 16-19

IMG (See Interactive Menugraph Generator)

INCLUDE STATION construct
using to request code extension, 7-21

input messages
accessing in C, COBOL74, Pascal, and RPG Compiler programs, 6-19
creating, 6-16
in message array, 6-17
processing in an ALGOL or a NEWP program, 6-19

INSPECT_TEXT_USING_TSET procedure, 6-9

installing
MLS environment, 4-1, 4-2
ODT environment, 4-7
SYSTEM/CENTRALSUPPORT file, 4-2, 8-14
SYSTEM/CONVENTIONS file
initially, 4-3
updates to, 4-3
translated files, 5-29

Interactive Datacomm Configurator (IDC)
assigning translating tables to stations, 7-16
code extension, 7-21
creating screens for, 2-6
editing class attribute code, 7-13

Interactive Menugraph Generator (IMG)
establishing languages, 5-4
files saved as, 5-29
help text (table), 5-24
keyword naming, 5-12
localizing screens, 5-4
menugraphs, 5-4
products that can be localized (table), 5-15
report, 5-4
user interface tool, 2-6

interfaces to the MLS environment, 2-4

international currency notation
data item in conventions, 1-17
editing, 3-19

International Organization for Standardization (ISO)
standards for convention definition names, 1-16
standards for currency codes, 1-17

internationalization
data classes, 6-9
defined, 1-7
features
in a noninternationalized environment, 4-2
not using in application programs, 6-1
truthsets, 6-9

INTMODE file attribute
using with coded character sets, 1-9

ISO (See International Organization for Standardization)

Italy convention definitions, 15-24

IXlateEBC
interface, 16-14
methods, 16-14

IXlateEBCDIC
interface, 16-9
methods, 16-10

IXlateEBCDICCCS
interface, 16-17
methods, 16-17

IXlateEBCDICCCS methods
XlateCPMsgFromHostCCS, 16-17
XlateCPMsgToHostCCS, 16-17, 16-18
XlateMsgFromHostCCS, 16-17
XlateMsgToHostCCS, 16-17, 16-18

J

j
date control character, 3-5, 3-10

Japan1 convention definitions, 15-25

Japan2 convention definitions, 15-26

JapanEBCDICJBIS8 code table (816CS00.00), 13-19

JapanV24 code table (8CS00.00), 12-47

JapanV24JBIS8 code table
(816CS00.00), 13-21
JBIS7 code table (16CS00.00), 13-23
JBIS8 code table (16CS00.00), 13-24
JISASCIIBIS7 code table
(816CS00.00), 13-25
Jordan convention definitions, 15-27

K

k
time control character, 3-4, 3-10
Kanji
code extension, understanding, 7-22
delimiters, 10-17
SDF Plus support, 6-32
Kenya convention definitions, 15-28
KEYEDIOII files, 2-2
KEYWORD commands, 5-11
conventions used with, 5-13
identifying in help text, 5-11
limitations for, 5-12
special character rules for, 5-13
using asterisks (*) with, 5-12
using the AUTONEXTKEYWORD
command with, 5-13
keywords, identifying in help text, 5-14
Kuwait convention definitions, 15-29

L

l
date control character, 3-4, 3-11
LAD, 1-11
LANGUAGE command
setting natural languages, 4-23
LANGUAGE task attribute, 4-28
languages (*See also* natural languages;
programming languages)
changes using MARC screens, 4-15
determining at session with MARC
screens, 4-23
setting at session with MARC
screens, 4-23

LANGUAGE-SET SESSION LANGUAGE
screen
setting natural languages, 4-23
Latin1EBCDIC
coded character set
Brazilian base, 14-6
CanadaEBCDIC base, 14-9
Norway base, 14-30
Spanish base, 14-43
Swedish1 base, 14-46
Swiss base, 14-49
Latin1EBCDIC code table (8CS26.01), 12-66
Latin2EBCDIC
coded character set
Czechoslovakia base, 14-16
Hungarian base, 14-22
Polish base, 14-33
Romanian base, 14-36
Latin2EBCDIC code table (8CS27.01), 12-76
Latin2ISO
code table (8CS27.00), 12-77
Latin5EBCDIC
coded character set
France base, 14-19
Turkish base, 14-53
Latin5EBCDIC code table (8CS29.01), 12-86,
12-96
Latin5ISO
code table (8CS29.00), 12-86
Latin5ISO code table (8CS29.00), 12-86,
12-96
LatinCyrillicEBC
coded character set, 14-39
LatinCyrillicEBCDIC code table
(8CS55.01), 12-106
LatinCyrillicISO code table (8CS55.00), 12-107
LatinGreek ccsversion, 14-1, 14-25
LatinGreekBTOS code table
(8CS41.00), 12-116
LatinGreekEBCDIC
code table (8CS30.01), 12-125
coded character set, 14-25
LatinGreekISO
code table (8CS30.00), 12-125
Lebanon convention definitions, 15-30

- length limitations for KEYWORD commands
 - in, 5-12
- LetsJ code table (816CS00.00), 13-27
- library coded character set number, 1-8
 - error, 9-5
 - use of, 9-5
- library names, identifying for SDF Plus libraries, 5-26
- limitations for
 - code extension availability, 7-5
 - KEYWORD commands in, 5-12
 - short help text, 5-14
- line advancement direction (LAD), 1-11
- line support processor (LSP) and code extension
 - using, 7-5
- lines per page, as a data item in conventions, 1-17
- listing available conventions
 - using CNV_NAMES procedure, 6-21
- literal text
 - general characteristics in convention templates, 3-2
 - rules for numeric and monetary conventions, 3-17
 - SDF Plus localization, 6-32
- LOCAL menu
 - setting ccsversions on, 4-14
- LocalEBCDIC
 - code table (8500.00), 12-135
- localization, defined, 1-7
- localizing, 5-4 (*See also* translating)
 - form libraries, 5-5
 - help libraries, 5-6
 - MARC, 5-4
 - menus, 5-4
 - messages, 5-23
 - screens, 5-4
 - screens created through SDF Plus, 5-5
 - text libraries, 5-5
 - user interfaces, 6-32
- locating MLS components, 2-1
- logical comparison
 - example, 6-15
- LOGICALIO
 - editing associated mapping table, 7-14

- long date
 - as a data item in a convention, 1-16
- long time, as a data item in a convention, 1-16
- LONG_DOW
 - date element, 3-12, 3-14
- LONG_MONTH
 - date element, 3-11, 3-14

M

- m
 - time control character, 3-5, 3-11
- MACROMAN code table (8CS61.03), 12-144
- Malaysia convention definitions, 15-31
- many-to-one mapping of characters, 9-5
- mapping
 - tables used in code extension
 - multiple coded character sets, 7-14
- mapping table, 1-14 (*See also* coded character set mapping tables)
 - contents of, 1-14
 - procedures for, 6-6
- MARC (*See* Menu-Assisted Resource Control)
- maximum integer monetary digits, 1-17
- MCP_BOUND_LANGUAGES procedure, 6-16
- MENU command, displaying menugraph titles, 5-4
- Menu-Assisted Resource Control (MARC), 4-23
 - establishing languages, 5-4
 - MENU command, 5-4
 - session level
 - selecting conventions using screens, 4-26
 - setting conventions using the CONVENTION command, 4-27
 - session level
 - setting natural languages using screens, 4-23
 - setting natural languages using the LANGUAGE command, 4-23
 - viewing conventions using screens, 4-25
 - viewing natural languages using screens, 4-23

- system level
 - setting system defaults, 4-15
 - viewing conventions using screens, 4-17
- menugraphs, definition of, 5-4
- menus, 5-4 (*See also* screens)
- message arrays, 6-17 (*See also* output messages)
 - accessing in C, COBOL74, Pascal, and RPG Compiler programs, 6-19
 - example ALGOL library for, B-1
 - example program to access
 - C programming language, B-3
 - COBOL74, B-5
 - COBOL85, B-6
 - Pascal, B-8
 - RPG, B-9
- messages
 - CENTRALSUPPORT library result, A-1
 - during system initialization, 4-8
 - initialization of CENTRALSUPPORT, A-23
 - translating
 - for Unisys products, 5-23
 - object file names (table), E-1
- MESSAGESEARCHER statement, 6-19
- metatoken brackets (<>), using to identify keywords, 5-11
- Mexico convention definitions, 15-32
- Micro A system
 - code extension requirements, 7-5
- MIN_SUFFIX
 - time element, 3-10, 3-15
- MINUTES
 - time element, 3-11, 3-15
- mixed, multibyte coded character set
 - mapping tables, 9-11
- mixed, multibyte coded character set tables, 13-1
- MLS (*See* MultiLingual System)
- MLSACCEPT statement, 6-19
- MLSDISPLAY statement, 6-19
- MLSMSG array, establishing languages, 5-5
- MLSTRANSLATE statement, 6-19
- MODCNV—Modify Convention screen
 - modifying custom conventions, 4-19
- modifying
 - C locales, 8-5
 - custom convention definitions, 1-15
 - with CENTRALSUPPORT library procedures, 6-31
 - with MARC, 4-17
 - restrictions, 1-6
 - SDF Plus forms, 5-7
- monetary control characters, 3-22
 - examples, 3-22
 - specifics, 3-16
 - table of, 3-18
 - using, 3-18
- monetary digits
 - as data items in conventions, 1-17
 - converting to real numbers, 6-28
- monetary elements in convention definitions
 - international currency notation, 1-17
 - international monetary digits, number of, 1-17
 - number of fractional digits, 1-17
 - number of maximum integer digits, 1-17
- monetary symbols
 - listing using the CNV_SYMBOLS procedure, 6-22
- monetary templates
 - processing, 3-17
 - table of control characters, 3-18
 - understanding, 3-16
 - use of numeric value placement symbol (#), 3-17
- monetary values, formatting
 - using a template, 6-29
 - using convention definition, 6-28, 6-29
- MONTH_OF_YEAR
 - date element, 3-11, 3-14
- MONTH_PREFIX
 - date element, 3-11, 3-14
- MSGTRANS screens
 - Edit screen
 - first physical message, 8-10
 - message 0 of LOCALES array, 8-11
 - Home screen, 8-9
 - Restart screen, 8-13
 - Save screen, 8-12
 - Start Session screen, 8-8

MSGTRANS utility
 adding or modifying C locales with, 8-8
 executing, 8-8
 messages that can be translated with, E-1
 translating
 file arrays, 5-5, 8-5
 literal text in conventions, 3-10, 3-13,
 3-15, 3-24

multilingual
 definition of, 1-4

MultiLingual System (MLS), 1-3
 components of, 2-1
 defined, 1-3
 functions of, 1-3
 installing the Centralsupport
 architecture, 4-1, 4-2
 interfaces to, 2-4
 options
 identifying default values, 6-3
 options
 setting at system level, 4-9

multiple octet characters
 transferring, 7-22

N

n
 date control character, 3-4, 3-11
 monetary and numeric control
 character, 3-23
 negative sign control character, 3-20

n[]
 monetary and numeric control
 character, 3-20, 3-23

naming convention for help books using
 IMG, 5-12
 SDF Plus, 5-10

natural languages
 defined, 1-4
 designing programs for, 6-2
 SDF Plus support, 6-33

setting
 at session level, 4-22
 halt/load requirement, 4-9
 priorities, 2-5
 system default using the
 SYSTEMOPTION
 command, 4-10
 support for, 1-15
 supporting in an application program, 6-16

NDLII (See Network Definition Language II)

negative sign control character, 3-20

Netherlands convention definitions, 15-33

Network Definition Language II (NDLII), 7-21
 code extension, 7-21

network support processor (NSP) and code
 extension, 7-5

NewZealand convention definitions, 15-34

Nigeria convention definitions, 15-35

Norway
 ccsversion, 14-1, 14-30
 convention definitions, 15-36

NorwayBTOS code table (8CS50.00), 12-153

null template, 3-2

number of international fractional digits
 in conventions, 1-17

numeric control characters, 3-22
 examples, 3-22
 in date and time templates, 3-5
 partial seconds components in time
 templates, 3-7
 preceded by a zero, 3-7
 series of control character, 3-6
 series of three, 3-6
 single control character, 3-6
 specifics, 3-16
 table of, 3-18
 using, 3-18
 year components in date and time
 templates, 3-8

numeric elements in convention definitions
 numeric time, 1-16

numeric symbols
 listing using the CNV_SYMBOLS
 procedure, 6-22

numeric templates
 processing, 3-17
 table of control characters, 3-18
 understanding, 3-16
 use of numeric value placement symbol
 (#), 3-17
 numeric value placement symbol (#)
 use in conventions, 3-17
 NX/Services (See Client Access Services)

O

o
 date control character, 3-5, 3-11
 object files with translatable messages, E-1
 octet data transfers involving
 double-octet characters, 7-22
 single-octet characters, 7-5
 one-to-one mappings, 6-10
 online documentation, translating
 asterisks (*), 5-14
 embedded keywords, 5-14
 KEYWORD commands, 5-11
 SECTION commands, 5-11
 Option parameter values, 6-7
 Option parameter, Complex procedure, C-7
 ordering sequence values, 1-14
 equivalent comparison, 6-13
 in strcoll function, 8-25
 logical comparison, 6-13
 rules, 1-14
 OSV (See ordering sequence values)
 output message array
 creating, 6-17
 CSINT_MSG, A-23
 example, 6-18
 output messages, 6-16
 accessing
 CENTRALSUPPORT library, 6-20
 in C, COBOL74, Pascal, and RPG
 Compiler programs, 6-19
 creating, 6-16
 in message array, 6-17
 processing in an ALGOL or a NEWP
 program, 6-19
 OUTPUT-MARC COMMAND OUTPUT screen
 viewing conventions, 4-18
 OUTPUTMESSAGE ARRAY declaration
 using, 6-17

P

p
 monetary and numeric control
 character, 3-23
 positive sign control character, 3-20
 time control character, 3-5, 3-11
 p[]
 monetary and numeric control
 character, 3-20, 3-23
 page length
 convention element, 1-4, 1-6
 default, 1-17
 determining, 6-30
 page width
 convention element, 1-4, 1-6
 default, 1-17
 determining, 6-30
 partial seconds component
 in time template, 3-7
 PARTIAL_SECS
 time element, 3-11, 3-15
 Pascal
 accessing message arrays, 6-19
 Pascal, accessing message arrays
 example program, B-8
 Peru convention definitions, 15-37
 Philippines convention definitions, 15-38
 pointer version, Complex procedure, C-3
 Poland convention definitions, 15-39
 Polish ccsversion, 14-1, 14-33
 poll/select
 using code extensions, 7-4
 positive sign control character, 3-20
 predetermined text in conventions
 translating, 3-24
 priorities for
 selecting conventions, 2-5
 setting natural languages, 2-5
 priority sequence values, 1-14
 in strcoll function, 8-25
 logical comparison, 6-13
 rules, 1-14
 privileges of files, reestablishing, 5-29
 procedural level
 access to internationalization features, 2-4
 products
 help text symbol file, 5-9
 localizing, 5-2

- program dump, A-5
- programmable read-only memory (PROM)
 - adapting for code extension, 7-5
- programming languages
 - defined, 1-4
 - languages supporting MLS, 2-5
 - specifying localization with, 4-28
- programs (See application programs)
- PROM
 - adapting for code extension, 7-5
- protocol for transporting data, 7-4
- PSV (See priority sequence values)

R

- r
 - time control character, 3-4, 3-11
- railroad diagrams, explanation of, F-1
- reconfiguring terminals for code extension, 7-5
- related product information, G-1
- repertoire number, 1-9
- RESET_PDUMP_COUNT procedure, A-5
- result messages, A-1
- reverse video control character
 - altering for code extension, 7-6
- Romania convention definitions, 15-40
- Romanian ccsversion, 14-36
- RPG Compiler, accessing message arrays
 - explanation of, 6-19
- RPG, accessing message arrays
 - example program, B-9
- run-time environment, establishing languages
 - for SDF Plus, 5-6
- Russian
 - ccsversion, 14-1, 14-39

S

- s
 - time control character, 3-5, 3-12
- screens, 5-4
 - ADDCNV—Add Convention screen, 4-19
 - based on IMG
 - establishing languages for, 5-4
 - localizing, 5-4
 - CONVDF—View/Add/Modify/Delete a Convention screen, 4-18

- CONVENTION-SET SESSION
 - CONVENTION menu, 4-26
- LANGUAGE-SET SESSION LANGUAGE
 - screen, 4-23
- LOCAL menu
 - setting ccsversions, 4-14
- MODCNV—Modify Convention
 - screen, 4-19
- MSGTRANS
 - Edit first physical message, 8-10
 - Edit message 0 of the LOCALES array, 8-11
 - Home screen, 8-9
 - Restart screen, 8-13
 - Save screen, 8-12
 - Start Session screen, 8-8
- OUTPUT-MARC COMMAND OUTPUT
 - screen, 4-18
- SC-SESSION CONTROL menu
 - selecting conventions, 4-26
 - setting natural languages, 4-23
- setting conventions, 4-16
- setting languages, 4-15
- SSCONV-Show/Set Convention
 - menu, 4-26
- SSLANG-Show/Set Language menu, 4-23
- SYSTEMCCSVERSION—Change System Ccsversion screen, 4-15
- SYSTEMCONVENTION—Change System Convention screen, 4-16
- SYSTEMLANGUAGE—Change System Language screen, 4-15
- translating, 5-4
- SC-SESSION CONTROL menu
 - selecting conventions, 4-26
 - setting natural languages, 4-23
- SDF Plus
 - accessing localized libraries, 5-6
 - business and cultural conventions, 6-33
 - ccsversions, 6-33
 - coded character set support, 6-32
 - establishing languages, 5-6
 - identifying
 - directory names, 5-26
 - type of library, 5-26
 - localizing
 - help libraries, 5-6
 - messages, 5-24
 - screens, 5-5

- natural languages, 6-33
- setting natural languages, 4-22, 5-6
- starting from CANDE, 5-7
- translating
 - product-based screens, 5-5
 - screens, 5-5
- user interface system, 6-32
- SDO character
 - using with code extension, 7-22
- SEC_SUFFIX
 - time element, 3-11, 3-15
- SECONDS
 - time element, 3-12, 3-15
- SECTION commands, 5-11
 - identifying in help text, 5-11
- security considerations, 5-29
- session language
 - setting, 4-23
 - viewing, 4-23
- session level
 - access to internationalization features, 2-4
 - selecting conventions using MARC screens, 4-26
 - setting conventions, 4-25
 - using system commands, 4-27
 - setting natural languages, 4-22
 - using MARC screens, 4-23
 - using system commands, 4-23
 - viewing conventions using MARC screens, 4-25
- setting
 - ccsversion, 4-10, 4-15
 - conventions, 4-25, 4-27
 - natural languages, 4-22
 - priorities, 2-5
 - system-level options, 4-9
- shift in character
 - using with code extension, 7-5
- shift out character
 - using with code extension, 7-5
- short date
 - as a data item in a convention, 1-16
- short help text, rules for, 5-14
- SHORT_DOW
 - date element, 3-12, 3-14
- SHORT_MONTH
 - date element, 3-11, 3-14
- SI character
 - using with code extension, 7-5
- Simple Installation (SI) program, 5-30
- single bracket invocation characters
 - using with Complex procedure, C-10
- single-octet data transfers, 7-5
- SO character
 - using with code extension, 7-5
- SOK character
 - using with code extension, 7-22
- sorting
 - by ccsversion ordering information, 1-14
 - ccsversion change effect, 4-9
 - CENTRALSUPPORT library procedures for, 6-6
 - equivalent ordering, 6-14
 - text strings, 6-13
- source files
 - help text lists of, D-1
- SOURCENDLII file
 - editing Adapter Control, 7-6
 - editing mapping table in Adapter Control, 7-12
 - LOGICALIO, 7-14
- SouthAfrica convention definitions, 15-41
- space characters
 - definition of, 1-8
- Spain convention definitions, 15-42
- Spanish ccsversion, 14-1, 14-43
- special characters in keywords, 5-13
- square brackets []
 - convention control characters, 3-16, 3-22
- SSCONV-Show/Set Convention menu
 - selecting conventions, 4-26
- SSLANG-Show/Set Language menu
 - setting natural languages, 4-23
- standard
 - conventions, 15-1
 - EBCDIC mapping table, editing, 7-12
 - help text information
 - copyright, 5-25
 - disclaimer, 5-25
 - trademark information, 5-25
- StandardTurkey convention definitions, 15-43
- star-bounded array, Complex procedure, C-3
- start-of-double-octet (SDO) character
 - using with code extension, 7-22
- start-of-Kanji (SOK) character
 - using with code extension, 7-22
- State array parameter, Complex procedure, C-7

- station variable
 - altering for code extension, 7-21
 - setting for code extension, 7-21
 - strings
 - converting to real numbers, 6-28
 - substitution
 - CENTRALSUPPORT library
 - procedure, 6-14
 - procedure for returning information
 - about, 6-14
 - rules for, 1-14
 - SDF Plus, 6-33
 - Sweden convention definitions, 15-44
 - Swedish1 ccsversion, 14-1, 14-46
 - Swiss ccsversion, 14-1, 14-49
 - Switzerland convention definitions, 15-45
 - symbol definition lists
 - syntax and uses, 3-16
 - symbol files, documentation
 - description of, 5-9
 - SYMBOL/SOURCENDLII file
 - editing Adapter Control, 7-6
 - editing mapping table in
 - Adapter Control, 7-12
 - LOGICALIO, 7-14
 - symbols used in convention templates, 3-1
 - system
 - dates, formatting
 - with convention definition, 6-25, 6-27
 - with template, 6-25, 6-27
 - initialization messages, 4-8
 - times, formatting
 - with convention definition, 6-25, 6-27
 - with template, 6-25, 6-27
 - system defaults
 - setting with the SYSTEMOPTIONS command, 4-10
 - system level
 - access to internationalization features, 2-4
 - viewing conventions using MARC screens, 4-17
 - SYSTEM/CCSFILE file
 - architecture component, 2-1
 - coded character set mapping tables, 1-9
 - contents of, 2-2
 - installing, 4-1
 - interface to, 2-3
 - SYSTEM/CENTRALSUPPORT file
 - installing, 4-1
 - translating convention text, 3-24
 - SYSTEM/CONVENTIONS file
 - contents of, 2-2, 15-1
 - deleting custom convention definitions, 4-20
 - interface to, 2-3
 - listing conventions in
 - using the CENTRALSUPPORT library procedure, 6-21
 - retrieving symbols for conventions, 3-17
 - when updates take effect, 4-17
 - SYSTEM/MARC/COMMANDER file, 5-5
 - SYSTEM/SDFPLUS/ARCHIVEMANAGER
 - messages, 5-24
 - SYSTEM/SLICESUPPORT file, 8-5
 - SYSTEMCCSVERSION—Change System Ccsversion screen
 - setting ccsversions, 4-15
 - SYSTEMCONVENTION—Change System Convention screen
 - selecting convention definitions, 4-16
 - SYSTEMLANGUAGE—Change System Language menu
 - setting natural languages, 4-15
 - SYSTEMOPTIONS (SYSOPS) command
 - examples of use, 4-13
 - syntax, 4-10
 - using to set or display system defaults, 4-10
- T**
- t
 - monetary and numeric control character, 3-24
 - thousands separator control character, 3-20
 - time control character, 3-5, 3-12
 - t[]
 - monetary and numeric control character, 3-20, 3-24
 - T27 terminals
 - reconfiguring for code extension, 7-5

- Taiwan convention definitions, 15-46
- task level
 - access to internationalization features, 2-4, 4-28
- TEACH AUTONEXTKEYWORD command, in products using SDF Plus, 5-13
- TEACH DIRECTORY command, using in help text, 5-10
- TEACH KEYWORD commands (See KEYWORD commands)
- TEACH LANGUAGE command, using in help text, 5-9
- TEACH SECTION commands, 5-11
- templates
 - accessing from a convention, 6-23
 - creating, 3-1, 3-2
 - creating convention, 6-20
 - defined in conventions, 3-1
 - long date, 1-16
 - long time, 1-16
 - monetary, 3-16
 - monetary, 1-16
 - numeric, 3-16
 - numeric, 1-16
 - numeric date, 1-16
 - numeric time, 1-16
 - processing
 - date templates, 3-3
 - monetary templates, 3-17
 - numeric templates, 3-17
 - time templates, 3-3
 - short date, 1-16
 - symbols used in, 3-1
 - table of numeric and monetary control characters, 3-18
- terminals compatible with code extension
 - describing, 7-5
- text
 - comparing and sorting, 6-13
 - identifying in help text, 5-14
 - predetermined, 3-24
- text libraries
 - access hierarchy, 5-6
 - identifying the directory name, 5-26
 - identifying the library name, 5-26
 - identifying the type of library, 5-26
 - localizing, 5-5
 - localizing with SDF Plus, 5-26
- text library messages, 5-24
- text line orientation (TLO), 1-11
- text strings (See control characters; graphic characters)
 - comparing, 6-13
 - positioning characters in, 6-15
 - processing in an MLS environment, 6-6
 - sorting, 6-13
- thousands separator control character, 3-20
- time control characters, 3-9
 - elements, 3-15
 - examples, 3-8, 3-9
 - specifying alphabetic components, 3-4
 - specifying numeric components, 3-5
- time display model for convention, 6-24, 6-26
- time elements
 - HOD_12_HRCL, 3-10, 3-15
 - HOUR_OF_DAY, 3-12, 3-15
 - HOUR_SUFFIX, 3-10, 3-15
 - MIN_SUFFIX, 3-10, 3-15
 - MINUTES, 3-11, 3-15
 - PARTIAL_SECS, 3-11, 3-15
 - SEC_SUFFIX, 3-11, 3-15
 - SECONDS, 3-12, 3-15
 - TIME_PREFIX, 3-10, 3-15
 - TIME_SUFFIX, 3-10, 3-15
- time templates
 - convention definition element, 1-16
 - examples, 3-9
 - processing, 3-3
 - undefined, 3-2
- TIME_PREFIX
 - time element, 3-10, 3-15
- TIME_SUFFIX
 - time element, 3-10, 3-15
- times
 - converting to HHMMSSPPPP format, 6-26
 - formatting, 6-25
- TLO, 1-11
- tools for
 - creating and localizing multilingual applications, 2-5
 - translation, 5-2
 - viewing application systems in a natural language, 2-6
- trademark information, dealing with, in help text, 5-25

- TRANS_TEXT_USING_TTABLE
 - procedure, 6-10
- Transaction Server
 - applications based on, 4-22
 - SDF Plus, 4-22
 - session language, 4-21
- transferring data using code extension, 7-4
- translate table (See coded character set mapping tables)
- translating, 5-4 (See also localizing)
 - control character in conventions, 3-24
 - convention text, 3-24
 - date elements with CNV_MSGS
 - array, 3-13
 - MARC, 5-4
 - menus, 5-4
 - procedures to process data with a
 - ccsversion, 6-9
 - screens, 5-4
 - SDF Plus screens, 5-5
 - time elements with CNV_MSGS
 - array, 3-15
- translation
 - defined, 1-7
- translation table (See coded character set mapping tables)
- transliteration table (See coded character set mapping tables)
- transmitting data
 - using code extension, 7-4
 - to a terminal, 7-10
 - to the mainframe, 7-7
- truthset
 - definition of, 6-9
 - use of, 2-3
- Turkish ccsversion, 14-1, 14-53

U

- UAE convention definitions, 15-47
- UCS2 code table (16CS00.00), 13-29
- UCS2NT code table (16CS00.00), 13-30
- UIDL files, 5-5
- UIS, 6-32
- underline control character
 - altering for code extension, 7-6

- Unisys products
 - establishing languages for, 5-29
 - localizing screens, 5-4
 - based on SDF Plus, 5-5
 - MARC
 - establishing languages for, 5-4
 - obtaining a localized version, 5-4
 - SDF Plus
 - establishing languages for, 5-6
 - localizing screens, 5-5
 - translating messages for, 5-23
- Unisys standards
 - coded character set mapping tables, 9-1
 - coded character sets, 9-1
- UnitedKingdom1 convention
 - definitions, 15-48
- UnitedKingdom2 convention
 - definitions, 15-49
- upgrading terminals to use with code extension, 7-5
- user interface system (UIS), 6-32
- user interfaces
 - accessing, 5-1
 - definition of, 1-4
 - designing, 6-32
- usercode level
 - access to internationalization features, 2-4
 - overriding, 2-5
 - setting defaults, 4-21
- user-created products, localizing
 - screens, 5-4
 - screens based on IMG, 5-4
 - screens based on SDF Plus, 5-5

V

- v
 - date control character, 3-4, 3-12
- VALIDATE_NAME_RETURN_NUM
 - procedure, 6-4
- VALIDATE_NUM_RETURN_NAME
 - procedure, 6-4
- variables for code extension, 7-21
- Venezuela convention definitions, 15-50
- viewing conventions at session level, 4-25

VSNCOMPARE_TEXT procedure, 6-13
 VSNESCAPEMENT procedure, 6-15
 VSNGETORDERINGFOR_ONE_TEXT
 procedure, 6-14
 VSNINFO procedure, 6-5
 VSNINSPECT_TEXT procedure, 6-10
 VSNORDERING_INFO procedure, 6-14
 VSNTRANS_TEXT procedure, 6-11
 VSNTRANSTABLE procedure, 6-11
 VSNTRUTHSET procedure, 6-12

W

w
 date control character, 3-4, 3-12
 writer-determined keywords, translating, 5-11

X

x
 date control character, 3-4, 3-12
 XlateEBCDIC interfaces
 IEnumXlateCCSes, 16-3
 IXlateEBC, 16-2
 IXlateEBCDIC, 16-2
 IXlateEBCDICCCS, 16-2, 16-3
 XlateEBCDIC server, 16-1
 XlateEBCDIC type library, 16-21

Y

y
 date control character, 3-5, 3-12
 YEAR
 date element, 3-12, 3-14
 year control character, 3-8
 YEAR_PREFIX
 date element, 3-12, 3-14
 YEAR_SUFFIX
 date element, 3-12, 3-14
 Yugoslavia convention definitions, 15-51

Z

z
 date control character, 3-12
 zero-bounded array, Complex procedure, C-2
 Zimbabwe convention definitions, 15-52

Special Characters

!1c!
 monetary control character, 3-23
 !1n!
 date control character, 3-4, 3-11, 3-14
 !1w!
 date control character, 3-4, 3-12, 3-14
 !1x!
 date control character, 3-12
 !2n!
 date control character, 3-4, 3-11, 3-13
 !a!
 time control character, 3-4, 3-10, 3-15
 !c!
 date control character, 3-13
 date control character, 3-10
 monetary control character, 3-23
 !d!
 date control character, 3-5, 3-13
 date control character, 3-10
 monetary and numeric control
 character, 3-23
 !e!
 date control character, 3-4, 3-10, 3-13
 !f!
 time control character, 3-4, 3-10, 3-15
 !h!
 time control character, 3-5, 3-10, 3-15
 !i!
 time control character, 3-4, 3-10, 3-15
 !j!
 date control character, 3-5, 3-10, 3-13
 !k!
 time control character, 3-4, 3-10, 3-15

- !!!
 - date control character, 3-4, 3-14
 - date control character, 3-11
- !m!
 - time control character, 3-5, 3-11, 3-15
- !n!
 - date control character, 3-4, 3-11, 3-14
 - monetary and numeric control character, 3-23
- !n[]!
 - monetary and numeric control character, 3-20
- !o!
 - date control character, 3-5, 3-11, 3-14
- !p!
 - monetary and numeric control character, 3-23
 - time control character, 3-5, 3-11, 3-15
- !p[]!
 - monetary and numeric control character, 3-20
- !r!
 - time control character, 3-4, 3-11, 3-15
- !s!
 - time control character, 3-5, 3-12, 3-15
- !t!
 - monetary and numeric control character, 3-24
 - time control character, 3-5, 3-12
 - time control character, 3-15
- !t[]!
 - monetary and numeric control character, 3-20
- !v!
 - date control character, 3-4, 3-12, 3-13
- !w!
 - date control character, 3-4, 3-12, 3-14
- !x!
 - date control character, 3-4, 3-12, 3-14
- !y!
 - date control character, 3-5, 3-12, 3-14
- !z!
 - date control character, 3-12, 3-14
- #
 - monetary and numeric control character, 3-23
 - use as numeric value placement symbol, 3-17
- * (asterisk), using
 - in help text, 5-14
 - to identify keywords, 5-11
- .BOOK FILETITLE command, using in help text, 5-9
- .DIRECTORY command, using in help text, 5-10
- .KEYWORD command, using in help text, 5-11
- .LANGUAGE command, using in help text, 5-9
- .SECTION command, using in help text, 5-11
- [] (square brackets)
 - using with conventions, 3-22
- <convention name>
 - setting conventions, 4-27
- <language name>
 - setting natural languages, 4-23
- <> (angle brackets), using to identify keywords, 5-11
- 12-hour clock, 3-10, 3-15
- 16-bit data transfers
 - performing, 7-22
- 16CS00.00 coded character set, 13-8, 13-9, 13-23, 13-24, 13-29, 13-30
- 1c
 - monetary control character, 3-19, 3-23
- 1n
 - date control character, 3-4
- 1w
 - date control character, 3-4
- 1x
 - date control character, 3-12
- 24-hour clock, 3-12, 3-15
- 2n
 - date control character, 3-4
- 816CS00.00 coded character set, 13-5, 13-6, 13-7, 13-10, 13-11, 13-12, 13-13, 13-14, 13-15, 13-16, 13-17, 13-19, 13-21, 13-25, 13-27

8500.00 coded character set, 12-135
8-bit data transfers
 performing, 7-4
8CS00.00 coded character set, 11-38, 11-48,
 11-92, 11-101, 11-110, 12-11, 12-30,
 12-47
8CS1.00 coded character set, 10-12
8CS1.01 coded character set, 10-17
8CS11.00 coded character set, 10-3
8CS11.01 coded character set, 10-2
8CS26.01 coded character set, 12-66
8CS27.00 coded character set, 12-77
8CS27.01 coded character set, 12-76
8CS27.03 coded character set, 11-29
8CS29.00 coded character set, 12-86, 12-96
8CS29.01 coded character set, 12-86, 12-96
8CS30.00 coded character set, 12-125
8CS30.01 coded character set, 12-125
8CS41.00 coded character set, 12-116
8CS43.00 coded character set, 11-20
8CS46.00 coded character set, 12-2
8CS46.01 coded character set, 12-2
8CS49.00 coded character set, 11-3, 11-11
8CS50.00 coded character set, 12-153
8CS51.01 coded character set, 10-26
8CS55.00 coded character set, 12-107
8CS55.01 coded character set, 12-106
8CS57.03 coded character set, 11-74
8CS58.03 coded character set, 11-65
8CS60.03 coded character set, 11-83
8CS61.03 coded character set, 12-144
8CS62.01 coded character set, 12-22



86000288-305