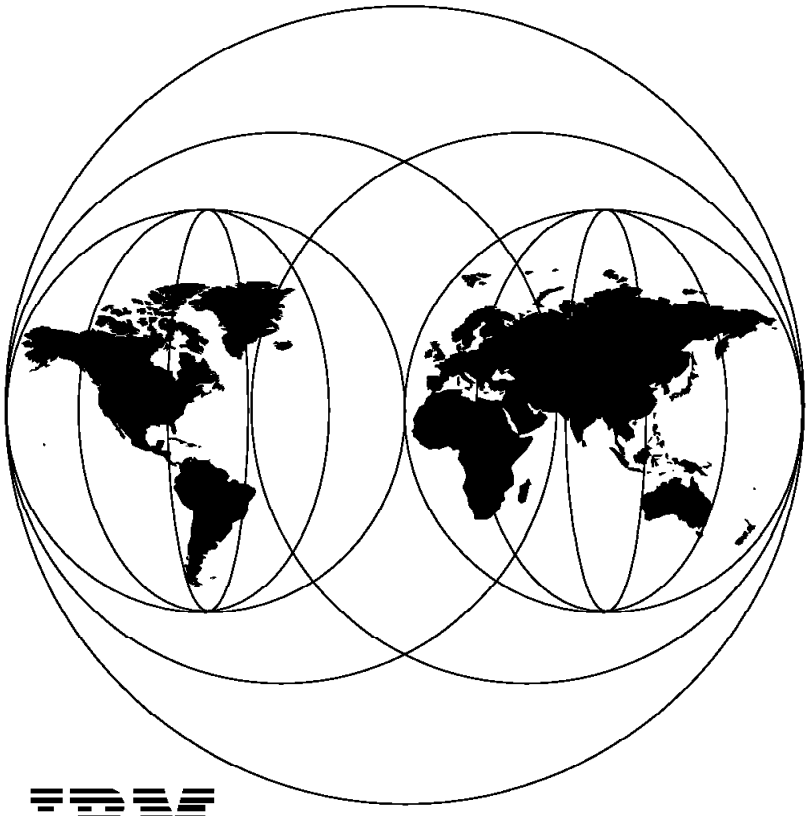
SG24-2154-00

Speak the Right Language with Your AS/400 System

August 1997



IBM

International Technical Support Organization
Rochester Center

International Technical Support Organization

SG24-2154-00

Speak the Right Language with Your AS/400 System

August 1997

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix G, “Special Notices” on page 139.

First Edition (August 1997)

This edition applies to V4R1 of OS/400, Program Number 5769-SS1.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures vii

Tables ix

Preface xi
The Team That Wrote This Redbook xi
Comments Welcome xii

Part 1. Get Your AS/400 System Language Right 1

Chapter 1. Introduction 3
1.1 Cultural Conventions 3
1.2 The History of Character Data 4
1.3 Code Page 4
1.4 Code Page Evolution 5
1.5 Tagging Character Data 5
1.5.1 Coded Character Set Identifier 6
1.6 Language Grouping 7
1.7 National Language Version 7

Chapter 2. AS/400 System National Language Support Overview 9
2.1 Primary and Secondary National Language Version 9
2.2 Language Sensitive Settings 10
2.2.1 Cultural System Values 10
2.2.2 Cultural Job Attribute Values 11
2.3 NLV Setting for Cultural Values 12
2.4 Multilingual and Multi-System Database Support 13
2.4.1 The Importance of CCSID 13
2.4.2 The Default CCSID 14
2.4.3 Changing CCSIDs of Physical Files 15
2.4.4 CCSID Special Values 16
2.5 System CCSID Support 16

Chapter 3. Considerations for Changing Language Sensitive Settings 19
3.1 Changing Primary Language 19
3.2 Changing System Values 20
3.3 Example Used in This Redbook 21
3.3.1 Current Language Sensitive Settings 21
3.3.2 Problems with ODBC 21
3.3.3 System Evaluation 21
3.3.4 Conclusion 21

Chapter 4. Planning for Changing Language Sensitive Settings 23
4.1 Collecting System Information 23
4.2 Planning Your Target Release 27
4.3 Planning Your Software Order 27
4.4 Ordering Software 28
4.5 Release Update 29
4.5.1 Recommendations for Release Prior to V3R1 29
4.6 Evaluate Database Information 29
4.6.1 Verify the Keyboard 35

Chapter 5. Preparing for Changing Language Sensitive Settings 37
5.1 Preparing to Change CCSID of Database Files 37
5.2 Other Information 38
5.2.1 Secondary Language Currently Installed 39
5.3 Obtain Preventive Service Planning Information 39

Chapter 6. Changing Language Sensitive Settings 41
6.1 Save Entire System 41
6.2 Changing Your Primary Language 42
6.3 Make Changes to CCSIDs of Database Files 42
6.4 Verify CA/400 Settings 43
6.5 Verify System Values and Applications 43
6.6 Save Entire System 44
6.7 Notify Software Provider about Your New Primary Language 44

Part 2. Background Information 45

Chapter 7. NLS considerations for Client Access/400 47
7.1 Original and Optimized Servers 47
7.2 Basic Principles 49
7.3 Original Clients 50
7.3.1 Shared Folders Function 52
7.3.2 File Transfer Function 52
7.3.3 Remote SQL 52
7.3.4 Message Function 52
7.3.5 Virtual Printer Function 52
7.3.6 Workstation Function - 5250 Emulation 52
7.3.7 Workstation Function - Printer Emulation 53
7.3.8 Considerations for Changing the NLS Option on the PC 53
7.4 Optimized Clients 54
7.4.1 Host Server CCSID 54
7.4.2 Forced Client Conversion for *HEX AS/400 Objects 57
7.4.3 Downloading the Conversion Tables 59
7.4.4 How PC Code Page is Determined 59
7.4.5 Windows 3.1 ODBC 61
7.4.6 File Transfer in Windows 95/NT and Optimized OS/2 Clients 62
7.4.7 ODBC in Windows 95/NT and Optimized OS/2 Clients 62
7.4.8 Network Drives 62
7.4.9 Data Queue APIs 63
7.4.10 Network Print 64
7.4.11 Remote Commands and Distributed Program Call APIs 64
7.4.12 Special Instructions for Setting Up PC5250 64
7.5 Other Client/Server Related Conversions 65
7.5.1 Copy Personal Computer Document 65
7.5.2 Copy Stream Files Commands 65

Chapter 8. NLS Considerations in TCP/IP Environment 67
8.1 FTP 67
8.1.1 TYPE E (EBCDIC) Transfers between Two EBCDIC Systems 68
8.1.2 TYPE A (ASCII) Transfers between Two EBCDIC Systems 68
8.1.3 TYPE A (ASCII) Transfers between EBCDIC and ASCII Systems 68
8.1.4 TYPE I (Binary) Transfers for Save Files 69
8.1.5 CCSID Tagging for New Files when Using FTP 69
8.2 TELNET 70

8.2.1 5250 and 3270 Full-Screen Mode Considerations . . . 70
8.2.2 VTxxx Full-Screen Mode Considerations . . . 71
8.3 SMTP and POP . . . 72
8.3.1 SMTP . . . 72
8.3.2 POP . . . 73
8.4 WSG . . . 76
8.5 HTTP . . . 77
8.6 SLIP . . . 77

Chapter 9. NLS Considerations for Printing Environment . . . 79
9.1 General Recommendations . . . 79
9.2 Printing in a Multilingual Environment . . . 80
9.3 Printer Files . . . 80
9.4 Printer File Parameters . . . 81
9.5 Printer File Support for CCSIDs . . . 83
9.6 Using Alternative Character Sets and Code Pages for Printer Output . . . 84
9.6.1 Character Sets and Code Pages . . . 84
9.6.2 Character Set and Code Page Combination (CHRIDs) . . . 84
9.6.3 Naming Convention for CHRIDs . . . 84
9.6.4 Selecting CHRIDs . . . 85
9.7 Character Identifier (CHRID) Values Supported . . . 86
9.8 Font Character Sets and Font Global Identifiers (FGID) . . . 90
9.8.1 Font Character Sets . . . 90
9.8.2 Naming Convention for Font Character Sets . . . 91
9.8.3 Selecting Font Character Sets . . . 91
9.8.4 Font Global Identifiers (FGIDs) . . . 92
9.8.5 Selecting Resident Fonts . . . 93
9.8.6 Font Substitution . . . 93
9.8.7 Stand-Alone Code Pages . . . 94
9.8.8 Naming Convention for Code Pages . . . 94
9.8.9 Selecting Code Pages . . . 95
9.8.10 Substituting Code Pages . . . 95
9.8.11 Coded Fonts . . . 96
9.8.12 Naming Convention for Coded Fonts . . . 96
9.8.13 Selecting Coded Fonts . . . 96
9.8.14 Substituting Coded Fonts . . . 97
9.9 Multilingual Support for ASCII Printers . . . 97
9.10 Host Print Transform Introduction . . . 97
9.11 Host Print Transform Functional Overview . . . 97
9.12 Customizing Your ASCII Printer Driver . . . 98

Chapter 10. Central Site Considerations . . . 101
10.1 Multilingual System Setup . . . 101
10.2 Application Installation Considerations . . . 104
10.2.1 CCSID Considerations for Application Providers . . . 105
10.3 Additional Considerations for Source Files . . . 106
10.4 Printer File Recommendations for Application Providers . . . 107
10.4.1 Consider Printing Environment of Customer . . . 109
10.4.2 Customer Printing Environment . . . 109
10.4.3 Printer Environment Impact on Language Oriented Printouts . . . 109
10.4.4 Summary . . . 110
10.5 Decimal and Thousands Separators and Negative Numbers . . . 111
10.5.1 Decimal and Thousands Separators Mismatch between Different Languages . . . 111
10.5.2 Current Logic on Editing Output and Input Fields on AS/400 System 111

10.5.3 Recommendations 112
10.6 Bidirectional Languages 113
10.6.1 Aspects of Bidirectional Language Writing Systems 113
10.6.2 Common User Access and Bidirectional Languages 119
10.6.3 Conclusions and Guidelines 120
10.7 DBCS Language Features for DBCS Environments 120
10.7.1 Language Feature Codes for DBCS 121
10.7.2 Recommended DBCS Test Environment 122

Appendix A. National Language Versions on AS/400 System 123

Appendix B. Invariant Character Set 127

Appendix C. Language Groupings 129

Appendix D. Worksheet for Planning Primary Language Change 131

Appendix E. Program and Command CHGLIBCCS 133
E.1 The CHGLIBCCS Command 135
E.1.1 How Program CHGLIBCCS Works 135

Appendix F. Code Page Example 137

Appendix G. Special Notices 139

Appendix H. Related Publications 141
H.1 International Technical Support Organization Publications 141
H.2 Redbooks on CD-ROMs 141
H.3 Other Publications 141

How to Get ITSO Redbooks 143
How IBM Employees Can Get ITSO Redbooks 143
How Customers Can Get ITSO Redbooks 144
IBM Redbook Order Form 145

List of Abbreviations 147

Index 149

ITSO Redbook Evaluation 151

Figures

1. Character Data Integrity Problem 6
2. The CCSID for Character Data 6
3. AS/400 System Support for CCSID 17
4. CCSID Conversion Process for Client Access/400 Original Clients 50
5. Sample EHNL.BAT for Language Feature 2924 - U.S. English 53
6. Host Server Job CCSID Determination Process 54
7. Sample CWBTFR.INI File for Translation of *HEX Data 58
8. Sample CWBTFR.INI File for No Translation of *HEX Data 58
9. The Client Access ODBC Setup ″Other″ Property Sheet 58
10. Windows 95/NT Client Access/400 History Log 59
11. Windows 95 Registry 60
12. Sample Output from CHCP Command 61
13. The Client Access Properties - Network Drives Property Sheet 63
14. ASCII-to-EBCDIC Mail Handling by POP Server 74
15. EBCDIC-to-ASCII Mail Handling by POP Server 75
16. How Externally Described Printer Files Are Used 81
17. Multilingual Single System Setup 101
18. Code Page 37 137

Tables

1. Variant Characters in EBCDIC Code Pages . . . 5
2. Cultural System Values . . . 10
3. Cultural Job Attribute Values . . . 11
4. Message Description CPX8416 . . . 12
5. Planning Primary Language Change . . . 22
6. Planning Software Order . . . 22
7. Planning CCSID Change with CHGLIBCCS Command* . . . 22
8. Feature Prefix for Secondary Languages . . . 28
9. Client Access for DOS, Client Access for DOS with Extended Memory and Client Access for OS/2 Functions . . . 48
10. Client Access for Windows 3.1 and Client Access Enhanced for Windows 3.1 . . . 48
11. Client Access Optimized for OS/2 . . . 48
12. Client Access for Windows 95/NT . . . 49
13. Example of ODBC with Different CCSID Values, INZPCS Parameters where ASCII Code Page = 862 - Hebrew PC Data, EBCDIC Code Page = 424 - Hebrew . . . 51
14. Example of Database Server Prestart Job CCSID Settings when Job is in Prestart Wait State . . . 55
15. Example of Database Server Prestart Job CCSID Settings when Job is Servicing a Connected User . . . 55
16. Example of Database Server Prestart Job CCSID Settings after Compatibility Tests . . . 56
17. Example of Server Conversion using Client Access for Windows 95/NT Client . . . 56
18. Languages that Need Special Font Files . . . 65
19. CCSID Code Page Tagging for AS/400 Files . . . 69
20. Supported MIME Standard Code Pages . . . 76
21. CHRID Values and Applicable Printers (CHRID Parameter) . . . 86
22. German Cultural System Values . . . 102
23. Primary Language Feature Code for DBCS . . . 121
24. Secondary Language Feature Code for DBCS (V4R1) . . . 121
25. Secondary Language Feature Code for DBCS (V3R6 and V3R7) . . . 121
26. Secondary Language Feature Code for DBCS (V3R1 and V3R2) . . . 121
27. Internationalized Software . . . 122
28. Language Specific Software . . . 122
29. National Language Versions on AS/400 System . . . 123
30. Languages without National Language Version . . . 124
31. Invariant Character Set . . . 127
32. Planning Primary Language Change . . . 131
33. Planning Software Order . . . 131
34. Planning CCSID Change with CHGLIBCCS Command . . . 131

Preface

This redbook can help you to implement the correct national language environment on your AS/400 system. Whether you plan to migrate your current primary language on your AS/400 system to one of the other available national language versions, or you simply want to verify your current system settings, you are guided through the tasks.

This redbook is divided into two parts:

Part 1 contains an introduction to national language support topics. It provides the tasks required to plan and perform a primary language change, or if you want to keep your current primary language, it can assist you in setting the correct system values.

Part 2 contains reference information with a detailed explanation of national language support for various topics. These topics include CA/400, TCP/IP applications, and printing.

Chapter 10 is especially targeted for application providers to make them aware of the considerations when implementing solutions in different countries or in a multilingual environment.

This redbook is intended to assist customers, business partners, and IBM technical and services professionals who are directly involved in setting up the AS/400 national language environment correctly.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Rochester Center.

Jens Gert Andersen is a Senior Systems Engineer in IBM Denmark and is a member of the Product Support Services unit within IBM Global Services. He has worked at IBM for 24 years and has 10 years of experience in national language support for midrange computers. He has written extensively, given presentations, and provided consulting services on the capabilities of national language support on the AS/400 system.

Modi Amir is an AS/400 Systems Engineer in Israel. He has 15 years of field experience working with S/36 and the AS/400 system. His areas of expertise include national language support and AS/400 system in a client/server environment.

Wolfgang Schmalz is an AS/400 Systems Engineer in Germany and a member of the AS/400 Technical Support unit. He has 19 years of field experience working with S/36 and the AS/400 system. His areas of expertise include printing and AFP.

Antanas Ursulis is an AS/400 Product Manager in IBM Lithuania. He has 13 years of field experience working with S/360 and S/370 systems, and for the last three years, he has worked for IBM as an AS/400 System Engineer and Product Manager.

Jaejin Ahn is an Advisory International Technical Support Specialist for the AS/400 system at the ITSO Rochester Center. Before joining the ITSO in 1996, he worked in the AS/400 business team, IBM-Korea, as a team leader of technical support and PID-Korea.

Thanks to the following people for their invaluable contributions to this project. Unless otherwise stated, all are from IBM Rochester Development Lab.

Bruce Vining

Greg Andrews

Kevin Vette

Pat Glenski

Rod Fletcher

Shigeru Shimada

Terry Luebbe

Dale Pirie
IBM Boulder

Don Gillespie
IBM Endicott

Gary Stange
IBM Endicott

Rich Bear
IBM Endicott

Markku Rahkamaa
IBM Finland

Olli Sysinoro
IBM Finland

Israel Gidali
IBM Israel

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 151 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com`
  For IBM Intranet users `http://w3.itso.ibm.com`

- Send us a note at the following address:

  redbook@vnet.ibm.com

Part 1. Get Your AS/400 System Language Right

Part 1 contains basic information about how to get your AS/400 system language right. The following chapters are included in Part 1.

- Chapter 1 contains an introduction to national language support topics for you to use throughout this redbook.
- Chapter 2 gives an overview of the AS/400 system implementation of national language support areas.
- Chapter 3 discusses considerations for changing language sensitive settings when you change the primary language.
- Chapter 4 gives information about how to plan for language sensitive settings.
- Chapter 5 lets you prepare for changing these language sensitive settings.
- Chapter 6 shows the detailed steps to change the language sensitive settings and to verify your system settings.

Chapter 1. Introduction

Since the introduction of the AS/400® system, many new features have been added to the system. Today, customers operate in a complex environment such as:

- Global marketplace for businesses
- Multilingual systems and networks
- Multiple platforms, such as AS/400 system, S/390 system, UNIX, and PCs
- New technologies, such as Internet, Intranet, Java, and Lotus Notes

New language features have been added to the initial list of supported features, and with Version 4 Release 1, even more language features are added to the list.

This chapter intends to provide introductory information on the national language support topics you need to understand before you use the following chapters of this redbook. Detailed information about all national language support aspects can be found in the *National Language Support*, SC41-5101 and *International Application Development*, SC41-5603.

1.1 Cultural Conventions

For an end user operating a computer in any country in the world, the expectation is that the computer programs respect the specific cultural conventions when processing or presenting data.

Examples of cultural conventions include:

- Date format and separator:
  - U.S.: MM/DD/YY
  - Lithuania: YY.MM.DD
  - Denmark: DD.MM.YY
- Decimal format:
  - U.S.: 123.45
  - Lithuania and Denmark: 123,45

Using the spoken and written language is most evident. When entering data into a computer, the user requires access to characters (letters in alphabet and symbols) that are naturally used in that language.

Even in one country, some cultural and linguistic conventions may vary by language within the country. For example, in Canada, one set of conventions apply for French and another for English.

Data exchange between the computer systems in different countries creates a problem to understand and represent this culturally sensitive information in the correct form.

1.2 The History of Character Data

Initially, computers worked in a single language (usually English) and limited exchange of data took place.

On a computer, data is simply a formation of binary values and only when interpreted by a device attached to the computer, the binary values form the characters.

Entering data into a computer is mainly done with keyboards. When you type on a keyboard, the keystrokes are transformed into binary values by the computer, a process we call encoding.

Different countries and languages have different keyboard layouts. There are a number of reasons for this. For instance, in France and in Belgium, the conventional layout of the alphabetic keys begins AZER, not QWER. The most evident differences exist because of the varying characters (symbols) used in each country.

Over time, countries wanted to add their own nationally-used characters for computer processing and this started evolving into many code pages.

For example, in Denmark, Ø is a capital letter in the Danish alphabet and its lower case equivalent is ø. In the U.S., both of these symbols can mistakenly be viewed as the numeric value zero. As computers started to be used in Denmark, Danish users wanted to use the entire Danish alphabet. Viewing the symbols presented at that time, it was found that the @ was not useful in Denmark and it was replaced with the Danish letter Ø.

Different types of computers use different encoding of the data such as EBCDIC, ASCII, ISO, and ANSI. When you exchange data between countries or computers, you have to convert between the different encoding and code pages.

1.3 Code Page

A code page is a collection of characters where each character is represented by a unique binary value known as a code point. Besides the characters, a code page also contains control characters used by the system or by applications.

OS/400 uses EBCDIC (Extended Binary Coded Decimal Interchange Code) encoding for storing the representation of a character. Each character is stored as an 8-bit binary (hexadecimal) value, which is a single byte. With this encoding, systems are referred to as Single-Byte Coded Character Set (SBCS) systems.

Other operating systems use different encoding. For example, DOS uses ASCII or IBM PC, and Windows use ANSI.

In a single byte, there are 256 possible values. Some of the values are reserved for control purposes, leaving 190 values to represent characters.

A code page is displayed as a matrix of rows and columns that assigns characters to special hexadecimal code points.

The code page may vary from country to country. Each code page is given a unique number that is used as an identifier.

Figure 18 on page 137 illustrates the U.S. English code page 37. Please refer to the *International Application Development*, SC41-5603, for additional illustrations of code pages supported on AS/400 business computing systems.

1.4 Code Page Evolution

Just as each language has its own particular keyboard layout, each language has evolved its own code page to cater to the special characters used in the language. There is still a set of characters (called the invariant character set) such as A-Z, a-z, and 0-9 that have a common encoding in most SBCS code pages. Refer to Appendix B, “Invariant Character Set” on page 127.

Problems can arise where the characters unique to one or more languages are concerned. Table 1 shows a portion of some of the characters in some different EBCDIC code pages.

Table 1. Variant Characters in EBCDIC Code Pages

Country	Code Page	7B	5B	7C	4A	C0	6A	D0
U.S., Canada, Portugal, Netherlands...	37	#	$	@	¢	{	|	}
MNCS (Belgium, Switzerland...)	500	#	$	@	[	{	|	}
Denmark/Norway	277	Æ	Å	Ø	#	æ	ø	å
U.K.	285	#	£	@	$	{	|	}

The table does not represent all of the differences between code pages; there are, in fact, more code pages than shown, and there are more characters in the code pages that have different code points.

Note: We recommend that you do not use these special characters 0x5B, 0x7B, and 0x7C in library names, database file names, or database field names.

As new languages are added to the list of supported languages, new code pages are added to include the characters needed for that language,

1.5 Tagging Character Data

Because customers operate with multiple platforms and often exchange data with other countries, there was a strong requirement to identify the character data in a unique way to ensure the integrity when exchanging data between languages and platforms.

Unless application programs take care of integrity, users in different countries see different characters on their terminal when they look at the same encoded data (for example, a database file).

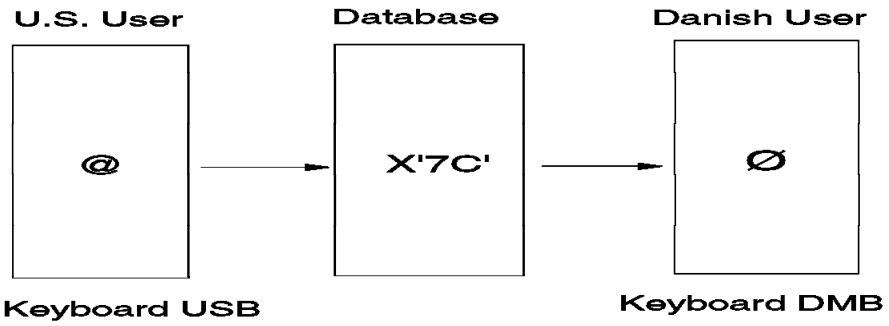


Figure 1. Character Data Integrity Problem

In Figure 1, a U.S. user enters the symbol @ into an AS/400 database file where it is encoded with the hexadecimal value 7C. A Danish user looking at the same database file sees the Danish upper case letter Ø.

1.5.1 Coded Character Set Identifier

To solve this character data integrity problem, an identifier (the coded character set identifier (CCSID)) was introduced. This identifier (the CCSID) is contained in the object description (for example, the database file object on the AS/400 system) and specifies the encoding of character data within the object.

When the character data is processed by another object with a different CCSID (for example, a job) the character data can be converted from one encoding to the other.

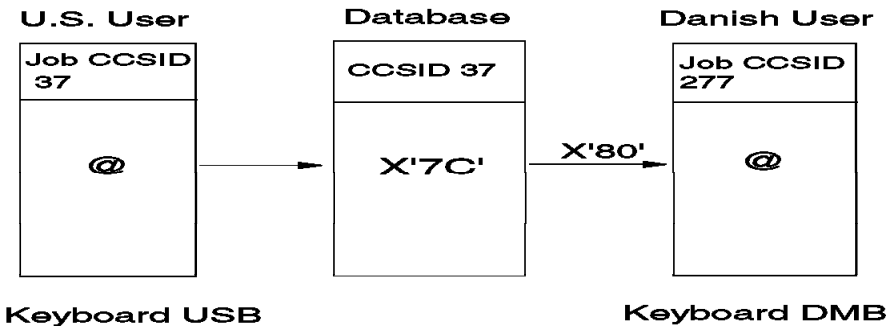


Figure 2. The CCSID for Character Data

In Figure 2, the U.S. user enters the same symbol into the database file. However, this time the database file is tagged with the CCSID 37. The Danish user job is set to CCSID 277. Now the AS/400 system converts the hexadecimal 7C to the corresponding hexadecimal value 80, which represents the symbol (@) in Denmark, and the correct symbol is shown on the Danish terminal.

1.6 Language Grouping

Some languages use the same characters and from a perspective of exchanging data, we can group those languages. Within a language group, you can look at and exchange data without losing integrity of character data. In other words, the characters appear the same to the users, even if the characters have a different encoding.

Refer to Appendix C, “Language Groupings” on page 129 for detailed information on the language grouping.

1.7 National Language Version

A national language version (NLV) is a version of the Operating System/400 (OS/400) where the cultural values are predefined for a particular language. For example, these include date and time format.

Except for a Double-Byte Coded Character Set (DBCS), there is no need for installing a particular National Language Version to get support for a language. The information you are given depends on the specific devices attached to the AS/400 system and the setting of system values and job attributes.

When you order a licensed program for an AS/400 system, you identify the national language version you want by specifying a language feature code. If you want to use more than one national language version of a licensed program on an AS/400 system, you can order additional languages. For example, if you are a customer in Germany, you may need support for both German and French on one system. You can order a national language version for German and a national language version for French. When you order more than one national language version for a system, you designate one of the versions as the primary language. All other national language versions are specified as secondary languages.

The system provides default system values for each of the primary languages. If some of the defaults do not meet the needs of your users, you can change some language-dependent system values.

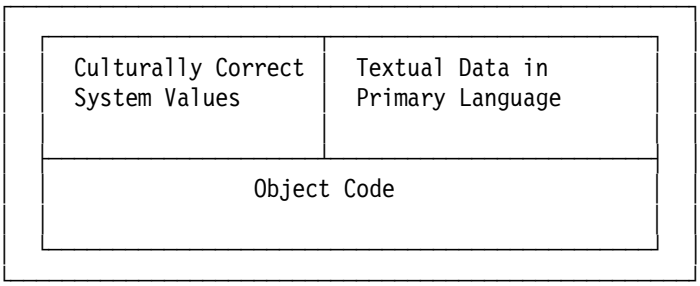
Chapter 2. AS/400 System National Language Support Overview

This chapter gives you an overview of the AS/400 system implementation of national language support (NLS).

2.1 Primary and Secondary National Language Version

Each AS/400 system has one primary national language version (NLV).

The primary national language version consists of code and textual data for each licensed program ordered. The textual data (also called machine readable information (MRI)) has been customized by IBM to meet the requirements of the language feature. It may be fully, partly, or even not translated into the language. Cultural system values are initialized for the primary language.



Each AS/400 system must always have a ″primary language″. This is the first language installed on the system, and is the language that is used for servicing purposes. For example, the system history log contains entries in the primary language (assuming that these entries have been translated).

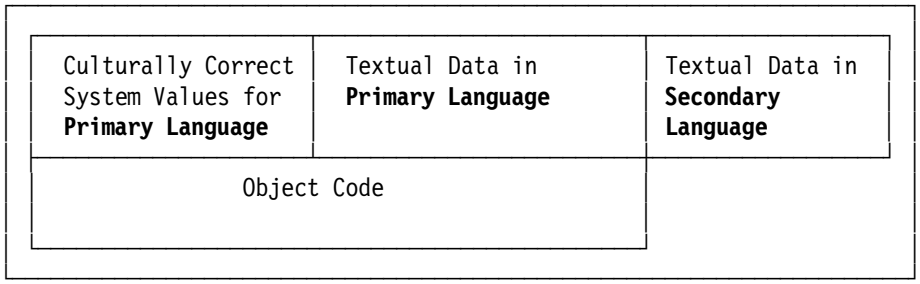
Table 29 on page 123 shows the available national language versions. If a national language version is not available for a specific language, one of the available national language versions must be used. In this case, customers have to set the culturally correct system values themselves.

Any languages other than the primary language are, by definition, secondary languages. Installing a secondary language just requires the MRI in the appropriate language. Software ordered from IBM as a secondary language cannot be used as a primary language since the program code is not contained within it.

A secondary language consists of textual data only for each licensed program. The distribution media contains textual data for all licensed programs.

When installing a secondary language, new containers for the textual data are created. For library objects, libraries are named QSYS29xx, where xx is from the last two digits of the secondary national language feature code. For objects in directories, sub-directories named MRI29xx are created and only the textual data for the licensed programs currently installed on the system are copied from the distribution media to the containers.

If new licensed programs are added after initial installation of a secondary language, the secondary language must be installed again to load the secondary language part for the new licensed programs.



Secondary languages are free of charge, but each requires approximately from 300MB to 500MB of disk space to install.

Multiple secondary languages can be installed on one AS/400 system. The maximum number of secondary languages on one AS/400 system depends on the available amount of disk space on the system and the availability of:

- A specific secondary language from the regional software distribution center
- Hardware (displays, keyboards, printers) for specific secondary languages

2.2 Language Sensitive Settings

Too often, national language support is mistakenly viewed as the task of simply translating the textual data part of a product into other national language.

The proper support is to provide facilities for the end user to obtain results that are culturally acceptable.

The AS/400 system provides these facilities through cultural system values and job attribute values.

2.2.1 Cultural System Values

The AS/400 system provides the default setting for the cultural information through system values. Table 2 shows the different system values used as defaults for cultural conventions.

Table 2 (Page 1 of 2). Cultural System Values

System Value	Description
QCCSID	Coded character set identifier
QCHRID	Character set/code page
QCNTRYID	Country identifier
QCURSYM	Currency symbol
QDATFMT	Date format
QDATSEP	Date separator
QDECFMT	Decimal format

Table 2 (Page 2 of 2). Cultural System Values	
System Value	**Description**
QKBDTYPE	Character set/code page for keyboard
QLANGID	Language identifier
QLEAPADJ	Leap year adjustment
QSRTSEQ	Sort sequence
QTIMSEP	Time separator

Use the Display System Value (DSPSYSVAL) command to display a system value and the Work with System Value (WRKSYSVAL) command to display or change system values.

Important

Do not change the cultural system values on your AS/400 system unless you are fully aware of the consequences. By changing some system values (especially QCCSID and QLANGID), you may impact your data.

2.2.2 Cultural Job Attribute Values

Job attributes are set for a job when you sign on or when a job starts.

Job attributes are set from the user profile associated with the job. The user profile often points to the default system values. The user profile may specify its own attributes for the cultural values, which allows multiple users with different needs to operate on the same AS/400 system. Table 3 specifies the cultural job attribute values.

Table 3. Cultural Job Attribute Values	
Job Attribute	**Description**
CCSID	Coded character set identifier
DFTCCSID	Default coded character set identifier
CNTRYID	Country ID
DATFMT	Date format
DATSEP	Date separator
LANGID	Language ID
SRTSEQ	Sort sequence
TIMSEP	Time separator

Once initialized, the job attributes for that job can be changed.

The default CCSID cannot be changed directly as this value is set through a dependency of CCSID and LANGID, which is explained in Section 2.4.2, “The Default CCSID” on page 14.

Important

Do not change the cultural values for job attributes or any values in user profiles on your AS/400 system unless you are fully aware of the consequences. By changing some cultural values (especially CCSID and LANGID), you may impact your data.

2.3 NLV Setting for Cultural Values

Message ID **CPX8416** in message file QCPFMSG is used during the installation to set cultural system values for the primary language.

For the primary language, message file QCPFMSG is located in library QSYS. The message description can be displayed by entering the following command:

```
DSPMSGD CPX8416
```

```
                          Display Formatted Message Text
                                                              System: xxxxxxxx
Message ID . . . . . . . . . . :   CPX8416
Message file . . . . . . . . . :   QCPFMSG
  Library  . . . . . . . . . . :     QSYS

Message . . . . :
QCHRID     697 37                  QCURSYM   $  QDATFMT    MDY QDATSEP    /
  QDECFMT      QLEAPADJ     0  QCCSID    37    QTIMSEP    : QLANGID    ENU
  QCNTRYID   US QIGCDEFNT *NONE



                                                                        Bottom
Press Enter to continue.

F3=Exit   F11=Display unformatted message text   F12=Cancel
```

This example of CPX8416 is from the U.S. English National Language Version.

Table 4. Message Description CPX8416

Identifier	Value	Description
QCHRID	697 37	Character set/code page
QCURSYM	$	Currency symbol
QDATFMT	MDY	Date format
QDATSEP	/	Date separator
QDECFMT	(blank)	Decimal format
QLEAPADJ	0	Leap year adjustment
QCCSID	37	Preferred NLV CCSID **Note:** QCCSID is initially set to 65535.
QTIMSEP	:	Time separator
QLANGID	ENU	Language ID
QCNTRYID	U.S.	Country ID

For a secondary language, message file QCPFMSG is located in library QSYS29XX. The message description can be displayed by entering the following command:

```
DSPMSGD CPX8416 MSGF(QSYS29XX/QCPFMSG)
```

If there are different language users on a single system, it is possible to use message CPX8416 to set the cultural job values during sign-on for a user. An initial program specified in the user profile can add QSYS29XX to the system library list in front of QSYS, retrieve the values from message ID CPX8416, and issue the CHGJOB command using the retrieved cultural values.

Note: Changes made to the cultural system values do not change the contents of message CPX8416 in message file QCPFMSG in QSYS, and changes made to CPX8416 do not influence system values. For customers not using their intended national language version, or languages not having a national language version, the system values and content of CPX8416 should be kept synchronized except for system value QCCSID. You may not want to change system value QCCSID until you are sure that there is no impact to your data.

Similar to message CPX8416, licensed programs use messages to define their values for cultural processing.

2.4 Multilingual and Multi-System Database Support

With the integrated DB2 for AS/400, the AS/400 system is an excellent database server. It can support both single language, multi-platform environments, and multilingual user environments.

The system needs to know about the content of the database files to perform necessary conversions requested by applications and clients.

Just as the fields are described with column heading and text, the file description contains information about the actual encoding of the database records. This is the coded character set identifier (CCSID).

2.4.1 The Importance of CCSID

Whenever a new, externally described database file is created, the AS/400 system stores information about the encoding (the CCSID) in the file description. Generally, all character fields in one database file have the same CCSID although the AS/400 allows the CCSID to be as granular as the field or column level.

All AS/400 systems are shipped with the system value QCCSID 65535, which means no conversion. The reason for selecting this value is to stay compatible with previous releases and not to cause any impact on customer data if the value was set to the recommended setting for the national language version. Refer to Section 10.2.1, “CCSID Considerations for Application Providers” on page 105 for more information.

Prior to V3R1, the default was to use the job attribute CCSID for all new created database files unless explicitly defined.

The problem with 65535 is that this CCSID simply does not work when character data processing is performed with different code pages involved. For example, DRDA (Distributed Relational Database Architecture) and Client Access/400 use

CCSIDs to determine how to convert character data to the client system. If the CCSID is 65535, no conversion is performed and the data is a garbage to the client as the AS/400 works primarily in EBCDIC and the client in ASCII.

IBM was concerned that customers who would find that all their database files were created with CCSID 65535 and now must go through a long and manual conversion of their database files to make the client support work.

For the AS/400 system to be a key player in the client/server environment, 65535 was viewed as no longer an acceptable default for new files.

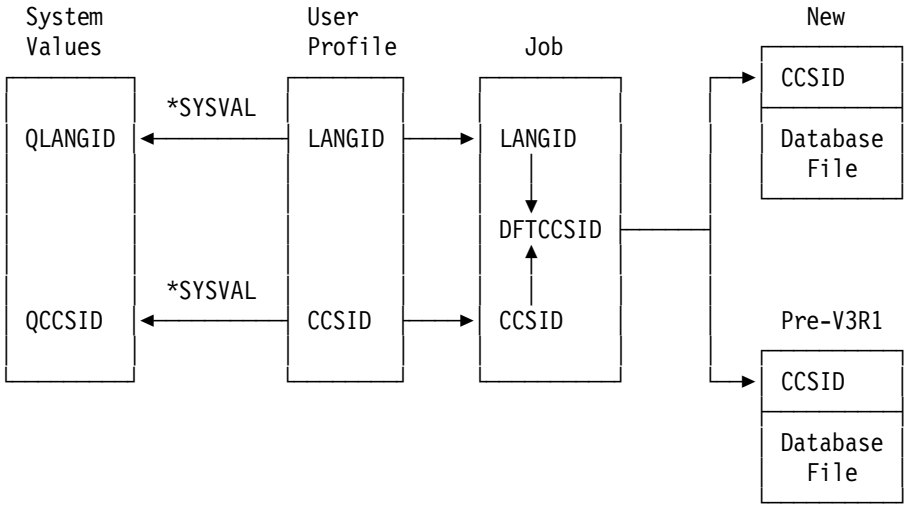
Therefore, a new job attribute was introduced with V3R1 (the default CCSID).

2.4.2 The Default CCSID

When a job is started on the AS/400 system, the job attributes are initialized.

In this section, we discuss the items that relate to CCSID issues. For detailed information on all job attributes, please refer to *Work Management*, SC41-5306.

The job attributes for the language ID and CCSID are taken from the user profile that starts the job. The user profile can define the values or point to the system values.



The default job level CCSID is determined as follows:

- If the job CCSID is 65535, the default CCSID is the CCSID corresponding to the language ID of the job.
- If the job CCSID is not 65535, the default CCSID is the same as the job CCSID.

Since the default for all user profiles is to use the system values, customers using a primary language not defining the correct QLANGID, may find that their database files are not tagged correctly.

Unintentionally, the incorrect tagging can be caused by upgrading the OS/400 release from pre-V3R1 to V3R1 or later. When OS/400 V3R1 or later release is

installed over a pre-V3R1 release, all files that are not explicitly tagged are tagged with the CCSID that corresponds to the system value language ID. The same happens when restoring pre-V3R1 database files. Although this was documented in the Memorandum to Users for V3R1, many customers did not pay attention to this important information.

For example, a customer in Israel uses the English uppercase and lowercase (feature code 2924) as their primary language, but the keyboard used is Hebrew. Looking at Table 29 on page 123, the language ID for feature 2924 is ENU and the associated CCSID is 37. Since the system value QCCSID is 65535, the default CCSID on this Israeli system is 37.

Any new externally described file (including the source file) that is not explicitly defined with a CCSID is created with the default CCSID.

For the Israeli customer, it means that the files carry the incorrect tag of CCSID 37. Looking at Table 29 on page 123, the language ID for Hebrew is HEB and the associated CCSID is 424, which is the correct CCSID tag for database files created with Hebrew keyboards.

As long as the data is used only by interactive applications on the AS/400 system, the data appears correctly. However, if a user wants to access the data with Client Access/400, the Hebrew characters do not appear correctly.

Note: Conversions within an AS/400 job can only occur when the CCSID of the job is different from the CCSID of the file and none of them are 65535.

Important

Do not change system or user profile values referring to the language ID or CCSID on your AS/400 system at this time. If your database files are incorrectly tagged, you need to plan and prepare for the change to your current, incorrectly tagged database files and the system values to be performed as one task to avoid further inconsistencies.

2.4.3 Changing CCSIDs of Physical Files

You can change the CCSID of a physical file by using the Change Physical File (CHGPF) command. Starting with V3R1, when the CCSID of a physical file is changed, its associated logical files are also changed to the new CCSID without having to delete and recreate the logical files.

There are a few restrictions that might prevent you from changing the CCSID of a physical file:

- A logical file built over the physical file has a sort sequence table and the CCSID of the sort sequence table is incompatible with the new CCSID in the physical file.
- A select/omit or join logical file, or both, that performs select/omits or joins between physical file fields that have different CCSIDs.
- A join logical file with a sort sequence table where the CCSID of the logical file′s secondary access path is different than the new CCSID for the physical file.
- CCSIDs are explicitly specified.

- A physical file has a physical file constraint. You must remove the physical file constraint before changing the CCSID of the physical file. When the CCSID is changed, the physical file constraint can be added again.

Changing the CCSID of a physical file does not change the content of the file, just the CCSID itself. The CCSID of the file should match the keyboard that entered the data into the file.

Note: When changing the CCSID of a source file, the last change date and time is updated. See Section 10.3, “Additional Considerations for Source Files” on page 106 for more information.

2.4.4 CCSID Special Values

Some of the values for the CCSID that have special meaning are:

- 65535: An object having this CCSID specified does not participate in any conversion. This is also known as *HEX.
- 65534: Informs the user of the object to look at a lower level for the actual CCSID. For example, the DSPFD shows -1, which instructs you to look at the DSPFFD to find the CCSID tagging each field. Not all fields have the same CCSID.
- 0: Informs the user of the object to look at a higher level in the hierarchy for the actual CCSID used. For example, the file field description internally specifies 0 in the CCSID when all fields within the file have the same CCSID, informing the system to look in the file description. Externally, the DSPFFD command propagates the CCSID of the file description to each field description.

2.5 System CCSID Support

The AS/400 system includes CCSID support in object types other than database files and jobs.

For detailed information about CCSIDs for other object types, refer to *National Language Support*, SC41-5101, and *International Application Development*, SC41-5603.

The complete picture of objects that support CCSIDs is shown in Figure 3 on page 17.

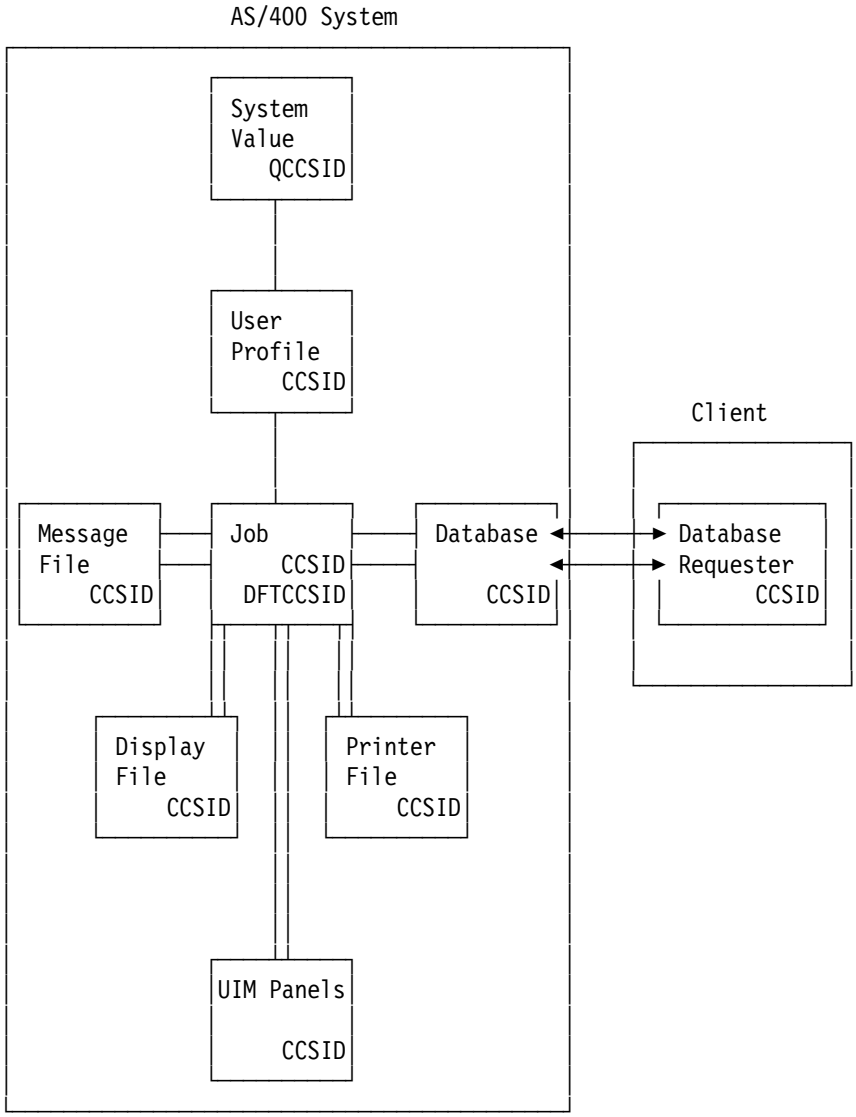


Figure 3. AS/400 System Support for CCSID

Remember the key points:

- The CCSID is just a tag in the object, an identifier to the outside world about the encoding of character data.
- The job attributes for CCSID and default CCSID are initiated by:
  - User profile values for CCSID and language ID, where the user profile may point to:
    - System values QCCSID and QLANGID
- Database files (including source files) are created with the default CCSID in the file description:
  - Unless explicitly defined
- Automatic conversion of character data is performed between two objects when:
  - The CCSIDs of the involved objects are different and none of them are 65535.

Chapter 3. Considerations for Changing Language Sensitive Settings

Before new national language versions were added to the supported list for the AS/400 system, customers had to use one of the forty available national language versions. With V4R1, AS/400 now provides support for 49 national language versions and enablement for several other languages. Some customers might still be using a primary language different from the language intended for use in that customer environment. Other customers may want to change to the new languages available with Version 4 Release 1.

Often, it is mistakenly viewed as a requirement to install the national language version for that language to get support in OS/400. This is not the case because OS/400 uses cultural job attributes to perform cultural functions. The defaults for these attributes are taken from the system values.

Customers might experience problems with CCSID tagging of database files if the language sensitive settings are not correctly specified. For example, when using Client Access/400 functions such as ODBC and file transfer with the optimized clients, you might see unexpected results.

With the flexibility of the AS/400 system, there is a significant number of different methods for setting up the system. It is not possible to cover every possible combination in this redbook. The intention is to cover the most frequently used methods and provide ideas for further investigation.

Sometimes, local solutions are developed to support an intermediate solution for the particular language or country. These local solutions may have modified an existing keyboard, display, and printer drivers in PCs, and might not be compatible with the new officially supported solutions.

If you need assistance for evaluating and changing your specific environment, you can contact IBM Global Services.

3.1 Changing Primary Language

Changing a primary language can have advantages:

- The cultural system values are correctly set.
- System cross reference files are in the CCSID of the primary language.
- Customized settings of cultural sensitive messages (for example, for editing).
- Translated version of textual data might be available.

You can use a secondary language media to change the primary language of an AS/400 system except when changing from a Single-Byte Coded Character Set (SBCS) to a Double-Byte Coded Character Set (DBCS) or from DBCS to SBCS. For SBCS to/from DBCS changes you need to have the installation media for your new primary language as well to perform the change.

Changing the primary language replaces the current textual data for the primary language with the textual data from the secondary language media used for changing your language. The cultural system values (except QCCSID) are set to the values defined by the new primary language.

Before you change your current primary language, you need to find out where you might need to make changes to current information at the same time you change the primary language.

- Evaluate your current system.
- Does tagging of database files need to change?
- Prepare the changes you need to make.

The key point in the evaluation process is to make sure that your database files are tagged with the correct CCSID. Usually, the CCSID must be the CCSID corresponding to the keyboard used for entering the data.

When you want to change the primary language, start with Chapter 4, “Planning for Changing Language Sensitive Settings” on page 23.

3.2 Changing System Values

While you may want to keep your current primary language, you need to get the correct settings of the cultural system values. Some of the national language versions are generic for a marketplace and the system values have to be specified correctly to ensure the correct tagging of database files. Some customers may be using a national language version different than the one actually designated for their language.

Recommendation

We recommend using this procedure if your target language is within the same language grouping as your current primary language.

If you are using characters not contained in the code page for your primary language in the system cross reference files, these characters do not appear correctly.

System cross reference files CCSIDs do not change with this procedure; this CCSID is only changed when you change the primary language.

When you want to ensure the correct settings for your system, use the steps given in the sections shown in the following list to evaluate your system and change system values. At the same, you might have to change the tagging of some of your database files. Go to the following sections to perform these steps:

- Section 4.1, “Collecting System Information” on page 23
- Section 4.2, “Planning Your Target Release” on page 27
- Section 4.6, “Evaluate Database Information” on page 29
- Section 4.6.1, “Verify the Keyboard” on page 35
- Section 5.1, “Preparing to Change CCSID of Database Files” on page 37
- Section 6.1, “Save Entire System” on page 41
- Section 6.3, “Make Changes to CCSIDs of Database Files” on page 42
- Change system values QLANGID, QKBDTYPE, and QCHRID to the values you have in the **target** column of your Planning Primary Language Change table.
- Section 6.5, “Verify System Values and Applications” on page 43

3.3 Example Used in This Redbook

In the following chapters of this redbook, we have used an example of a customer installation.

3.3.1 Current Language Sensitive Settings

The system has the following characteristics:

- System installed in Israel with U.S. English as primary language
- No secondary languages installed
- System values were not changed:
  - QLANGID: ENU
  - QCCSID: 65535
  - QKBDTYPE: USB
  - QCHRID: 697 37

3.3.2 Problems with ODBC

The customer has not experienced any problems with the applications running on the system and all data appears to be correct.

However, the customer starts using Client Access/400 ODBC and the data presented on the PC appears to be corrupted.

3.3.3 System Evaluation

After evaluating this system, the worksheet copied from Appendix D, “Worksheet for Planning Primary Language Change” on page 131 looks similar to the following page.

3.3.4 Conclusion

This customer’s problem with ODBC was caused by improperly setting QLANGID when V3R1 was installed over the previous release.

This customer should order the national language version for Hebrew with secondary language feature code 5761 and use it to change the primary language from English to Hebrew.

At the same time you change the primary language from English to Hebrew, all database files currently tagged with 37 must change to 424.

Table 5. Planning Primary Language Change

Item	Description	Current	Target
1	OS/400 release	V3R1M0	V3R1M0
2	Primary language	2924	2961
3	Secondary languages	None	None
4	System value QLANGID	ENU	HEB
5	System value QCCSID	65535	
6	System value QKBDTYPE	USB	NCB
7	System value QCHRID	697 37	941 424
8	Assumed CCSID for database files	37	424
9	Evaluated database CCSIDs	37 424	

Table 6. Planning Software Order

Task	Description	Action	Feature
1	Do you need to upgrade your current release to a newer release before you change the current primary language?	None	
2	Are there any secondary languages you need on your target release?	None	_ _
3	Identify the secondary language you need to order for changing the primary language on your system.	**ORDER**	5761

*Table 7. Planning CCSID Change with CHGLIBCCS Command**

Library	From CCSID	To CCSID
*ALL	37	424

Note: Please refer to the Appendix E, “Program and Command CHGLIBCCS” on page 133 for the detailed information.

Chapter 4. Planning for Changing Language Sensitive Settings

This chapter guides you through the planning tasks you need to complete before changing the current primary language to a new primary language.

These are the following steps that you must perform:

- Collect information about your current system.
- Plan and order appropriate software features.
- Evaluate your database files.

4.1 Collecting System Information

Make a copy of Appendix D, “Worksheet for Planning Primary Language Change” on page 131. In this section, fill in the current and target columns in the Planning Primary Language Change table.

Perform the following steps to collect information about the current national language status of your system:

1. Use the GO LICPGM command and select option 10 on the Work with Licensed Programs menu to display installed licensed programs.

   If your display is similar to this, continue with the following instructions. Otherwise, go to step 2.

```
                    Display Installed Licensed Programs
                                                          System:   SYSTEM01
Licensed                                                    Installed
Program      Description                                     Release
5763SS1      OS/400 - Library QGPL                           V3R1M0  1
5763SS1      OS/400 - Library QUSRSYS                        V3R1M0
5763SS1      Operating System/400                            V3R1M0
5763SS1      OS/400 - Extended Base Support                  V3R1M0
5763SS1      OS/400 - Online Information                     V3R1M0
5763SS1      OS/400 - S/36 and S/38 Migration                V3R1M0
5763SS1      OS/400 - System/36 Environment                  V3R1M0
5763SS1      OS/400 - System/38 Environment                  V3R1M0
5763SS1      OS/400 - Example Tools Library                  V3R1M0
5763SS1      OS/400 - AFP Compatibility Fonts                V3R1M0
5763SS1      OS/400 - *PRV CL Compiler Support               V3R1M0
5763SS1      OS/400 - 9406 Problem Analysis                  V3R1M0
5763SS1      OS/400 - S/36 Migration Assistant               V3R1M0
5763SS1      OS/400 - Host Servers                           V3R1M0
5763SS1      OS/400 - Openness Includes                      V3R1M0
                                                                      More...
Press Enter to continue.

F3=Exit   F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 1994.
```

   Review the list of installed licensed programs. If a line is shown in reverse image, the option or product is not correctly installed. You must resolve the errors before you continue.

   Write the installed release **1** for item 1 (OS/400 release) on your worksheet in the **current** column.

   Continue with step 4.

2. Your display should look similar to the following display:

```
                    Display Installed Licensed Programs
                                                          System:   SYSTEM01
Licensed  Installed
Program   Status       Description
5763XC1   *COMPATIBLE  Client Access/400 - Windows 3.1 SBCS
5763XC1   *COMPATIBLE  Client Access/400 - Windows 3.1 RUMBA SBCS
5763XC1   *COMPATIBLE  Client Access/400 - Windows 3.1 PC5250
5763XC1   *COMPATIBLE  Client Access/400 - GraphicOps for Windows
5763XC1   *COMPATIBLE  Client Access/400 - Ultimedia Facilities
5763XC1   *COMPATIBLE  Client Access/400 - Graphical Access
5763XC1   *COMPATIBLE  Client Access/400 - SysObject Access
5763XD1   *COMPATIBLE  Client Access for Windows 95/NT
5763XF1   *COMPATIBLE  Client Access/400 for OS/2
5763XF1   *COMPATIBLE  Client Access/400 - OS/2 SBCS
5763XF1   *COMPATIBLE  Client Access/400 - OS/2 RUMBA SBCS
5763XF1   *COMPATIBLE  Client Access/400 - OS/2 Communications Mgr
5763XF1   *COMPATIBLE  Client Access/400 - GraphicOps for OS/2
5763XF1   *COMPATIBLE  Client Access/400 - Ultimedia Facilities
                                                                    More...
Press Enter to continue.

F3=Exit   F11=Display release   F12=Cancel
```

Review the list of installed licensed programs. If the Installed Status shows *ERROR or *BACKLEVEL, you must resolve these before you continue.

***ERROR** The product has not been installed successfully or is only partially installed. For example, the language objects for the licensed program that are the same national language version (NLV) as the operating system may not have been installed successfully. Use the Check Product Option (CHKPRDOPT) command to determine the cause of the failure.

***BACKLEVEL** The product is successfully installed but its version, release, and modification level are not compatible with the currently installed level of the operating system. You need to update the product to be compatible, or delete the product from your system.

3. Use F11 to display the release of the operating system installed.

```
                    Display Installed Licensed Programs
                                                          System:   SYSTEM01
Licensed  Installed
Program    Release   Description
5716SS1    V3R7M0  1 OS/400 - Library QGPL
5716SS1    V3R7M0    OS/400 - Library QUSRSYS
5716SS1    V3R7M0    Operating System/400
5716SS1    V3R7M0    OS/400 - Extended Base Support
5716SS1    V3R7M0    OS/400 - Online Information
5716SS1    V3R7M0    OS/400 - System/36 Environment
5716SS1    V3R7M0    OS/400 - Example Tools Library
5716SS1    V3R7M0    OS/400 - AFP Compatibility Fonts
5716SS1    V3R7M0    OS/400 - *PRV CL Compiler Support
5716SS1    V3R7M0    OS/400 - Host Servers
5716SS1    V3R7M0    OS/400 - Openness Includes
5716SS1    V3R7M0    OS/400 - Advanced 36
5716SS1    V3R7M0    OS/400 - Locale Source Library
5716AF1    V3R7M0    AFP Utilities for OS/400
                                                                    More...
Press Enter to continue.

F3=Exit   F11=Display option   F12=Cancel
```

Write the installed release **1** for item 1 (OS/400 Release) on your worksheet in the **current** column.

4. Select option 20 on the Work with Licensed Programs menu to display installed secondary languages.

```
                    Display Installed Secondary Languages
                                                          System:   SYSTEM01
Primary language . . . . . :   2924  2
Description  . . . . . . . :   English

                                                                    Installed
Language     Description                          Library           Release
 3
  (No secondary languages installed.)

                                                                        Bottom
Press Enter to continue.

F3=Exit   F12=Cancel
(C) COPYRIGHT IBM CORP. 1980, 1996.
```

Write the primary language feature number **2** for item 2 (primary language) on your worksheet in the **current** column.

Write the secondary language numbers **3** for item 3 (secondary languages) on your worksheet in both the **current** and the **target** column. If no secondary languages are installed, write NONE.

5. Find the system value for the language identifier on your current system by using the following command:

   ```
   DSPSYSVAL QLANGID
   ```

```
                            Display System Value

System value . . . . . :   QLANGID
Description  . . . . . :   Language identifier


Language identifier  . :   ENU   4   Language abbreviation




Press Enter to continue.

F3=Exit   F12=Cancel
```

Write the language identifier **4** for item 4 (system value QLANGID) on your worksheet in the **current** column.

6. Find the system value for CCSID on your current system by using the following command:

   ```
   DSPSYSVAL QCCSID
   ```

```
                              Display System Value

System value . . . . . :   QCCSID
Description  . . . . . :   Coded character set identifier


Coded character set
  identifier . . . . . :   65535   5   1-65535









Press Enter to continue.

F3=Exit   F12=Cancel
```

Write the CCSID **5** for item 5 (system value CCSID) on your worksheet in the **current** column.

7. Find the system value for the keyboard type on your current system by using the following command:

   DSPSYSVAL QKBDTYPE

```
                              Display System Value

System value . . . . . :   QKBDTYPE
Description  . . . . . :   Keyboard language character set


Character set  . . . . :   USB   6   Language/Country abbreviation









Press Enter to continue.

F3=Exit   F12=Cancel
```

   Write the keyboard character set **6** for item 6 (system value keyboard type) on your worksheet in the **current** column.

8. Find the system value for the character set and code page on your current system by using the following command:

   DSPSYSVAL QCHRID

```
                              Display System Value

System value . . . . . :   QCHRID
Description  . . . . . :   Graphic character set and code page


Character ID . . . . . :   697   7         1-32767
Code page  . . . . . . :   37    7         1-32767









Press Enter to continue.

F3=Exit   F12=Cancel
```

   Write both values **7** for item 7 (system value QCHRID) on your worksheet in the **current** column.

4.2 Planning Your Target Release

In this context, ″target release″ means the national language version you use as your first natural choice. However, your natural choice might not be available, or you deliberately select a different primary language.

For example, Hebrew was not available as a national language version until V3R1. In this section, a customer in Israel chooses Hebrew as the ″target release″ for our evaluation purpose.

Locate your target language in Table 29 on page 123 using the national language version column. Some languages do not have a national language version. If your language is not in Table 29 on page 123, you should be able to locate it in Table 30 on page 124 so you do not change your primary language. Write the following information from the table in the **target** column on your worksheet:

- Use release entry to determine item **1**, OS/400 release:
  - If your current release entry is later than the release entry, or the release entry is blank, write your current release.
  - Otherwise, write the release entry from the table.
- Feature code number in item **2** (primary language)
- Language ID in item **4** (system value QLANGID)
- Keyboard in item **6** (system value QKBDTYPE)
- CHRID in item **7** (system value QCHRID)
- CCSID in item **8** (assumed CCSID for database files)

4.3 Planning Your Software Order

In this task, you identify the software to order that is required for changing the primary language of your system.

Find your copy of Appendix D, “Worksheet for Planning Primary Language Change” on page 131 with the information you entered in Section 4.1, “Collecting System Information” on page 23 and Section 4.2, “Planning Your Target Release.” Use the information in the Planning Primary Language Change table to fill in the action and feature columns in the Planning Software Order table. Perform the following tasks:

1. Compare current and target columns for item **1** (OS/400 release) on your Planning Primary Language Change table.
   a. If the current column is equal to the target column, write NONE in task 1 in the Planning Software Order table under the **action** column and continue with Step 2.
   b. Otherwise, write ORDER in item 1 in the Planning Software Order table. Locate item 2 (primary language) in the **current** column in the Planning Primary Language Change table and write this in task 1 in the **feature** column of the Planning Software Order table.
2. Locate item 2 (primary language) in the **target** column of the Planning Primary Language Change table. Write the last two digits from the primary

language feature code (in case of 2924, 24 is the last two digits) after the _ _ in the Planning Software Order table. task 3, under **feature** column.

3. Locate item 3 (secondary languages) in the **target** column of the Planning Primary Language Change table. If the column indicates NONE, write NONE in task 2 of the Planning Software Order table in the **action** column. Otherwise, write the last two digits after the _ _ in task 2 in the Planning Software Order table in the **feature** column.

The next step is to finalize the feature codes for the secondary languages you need.

Use your entry from the Planning Primary Language Change table for item 1 (OS/400 release) in the target column to locate the first two digits in the following table.

Table 8. Feature Prefix for Secondary Languages

Release	Feature Prefix
V3R1, V3R2	57
V3R6, V3R7	56
V4R1	55

Write the two digits in the Planning Software Order table for task 2 and 3, in front of the other two digits, replacing the _ _.

4.4 Ordering Software

You can change the primary language of your system using one of the following two methods:

- By using secondary language media to replace the language objects
- While replacing a release using the manual installation process

We recommend using the first option, which is changing the primary language using a secondary language media.

Note!

If you change the primary language during a release update, you may not know if a problem is caused by the release update or by the primary language change. By keeping these procedures separate, you have better control over problem determination.

On your copy of the worksheet from Appendix D, “Worksheet for Planning Primary Language Change” on page 131, you have the information in the Planning Software Order table to order your software. If you need to upgrade your current release, you also have to upgrade the licensed programs to the new release.

Give the order information to your AS/400 system software provider and continue with the next section.

4.5 Release Update

If you need to update your current release before you change your primary language, follow the procedures you receive for the release update.

We recommend that you perform the release update one to two weeks before you change the primary language.

4.5.1 Recommendations for Release Prior to V3R1

If your current release is prior to V3R1, make sure that the language ID is set correctly, corresponding to your target release language ID before you update your current release to the target release.

With the information you entered on your copy of the Planning Primary Language Change table, you can change the following system values by using your entry in the **target** column:

```
CHGSYSVAL QLANGID
```

Follow the instructions for updating your release before you continue with the preparation for changing the primary language.

4.6 Evaluate Database Information

Collect information about CCSIDs used in your database files by performing the following steps:

1. Create a new library on your system (for example: CRTLIB ZCCSID)
2. Retrieve file attributes for all physical files in user libraries and store this information in a output file by using the DSPFD command:

```
Selection or command
===> DSPFD FILE(*ALLUSR/*ALL) TYPE(*ATR) OUTPUT(*OUTFILE) FILEATR(*PF)
       OUTFILE(ZCCSID/ALLFILES)
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assista
F23=Set initial menu
```

3. Summarize the CCSIDs used in your database files. The following example uses Query/400, but it can be substituted for a similar tool.

   Create and run a query using the WRKQRY command to summarize the used CCSIDs:

```
                              Work with Queries

 Type choices, press Enter.

   Option  . . . . . .   1             1=Create, 2=Change, 3=Copy, 4=Delete
                                       5=Display, 6=Print definition
                                       8=Run in batch, 9=Run
   Query . . . . . . .   CCSIDSUM      Name, F4 for list
     Library . . . . .     ZCCSID      Name, *LIBL, F4 for list




 F3=Exit       F4=Prompt       F5=Refresh       F12=Cancel
                                                 (C) COPYRIGHT IBM CORP. 1988
```

```
                              Define the Query

Query . . . . . :   CCSIDSUM          Option  . . . . . :   CREATE
  Library . . . . :     ZCCSID        CCSID . . . . . :   65535

Type options, press Enter.  Press F21 to select all.
  1=Select

Opt    Query Definition Option
 1     Specify file selections
       Define result fields
 1     Select and sequence fields
 1     Select records
 1     Select sort fields
       Select collating sequence
       Specify report column formatting
 1     Select report summary functions
 1     Define report breaks
 1     Select output type and output form
       Specify processing options

F3=Exit            F5=Report          F12=Cancel
F13=Layout         F18=Files          F21=Select all
```

```
                           Specify File Selections

Type choices, press Enter.  Press F9 to specify an additional
  file selection.

  File . . . . . .   ALLFILES       Name, F4 for list
    Library  . . . .     ZCCSID     Name, *LIBL, F4 for list
  Member . . . . . .   *FIRST        Name, *FIRST, F4 for list
  Format . . . . . .   *FIRST        Name, *FIRST, F4 for list



F3=Exit           F4=Prompt          F5=Report           F9=Add file
F12=Cancel        F13=Layout         F24=More keys
```

Locate and select the PHCSID field for the CCSID.

```
                        Select and Sequence Fields

Type sequence number (0-9999) for the names of up to 500 fields to
  appear in the report, press Enter.

Seq   Field            Text                                          Len  Dec
  10  PHCSID           -1=See DSPFFD                                   5    0
      PHRCEN           Retrieval century:  0=19xx, 1=20xx              1
      PHRDAT           Retrieval date:  year/month/day                 6
      PHRTIM           Retrieval time:  hour/minute/second             6
      PHFILE           File                                           10
      PHLIB            Library                                        10
      PHFTYP           Type of file:  P=PF, R=DDM PF                   1
      PHFILA           File attribute:  *PHY                           4
      PHMXD            Reserved                                        3
      PHFATR           File attribute:  PF or PF38                     6
      PHSYSN           System Name (Source System, if file is DDM      8
      PHASP            Auxiliary storage pool ID:  1=System ASP        3    0


                                                                    More...
F3=Exit          F5=Report         F11=Display names only    F12=Cancel
F13=Layout       F20=Renumber      F21=Select all            F24=More keys
Press Enter to confirm.
```

The list of libraries excludes libraries starting with # and Q.

Note: Character # must be the character corresponding to code point X'7B' in user's code page. If you have any user database files in library QGPL, evaluate the CCSID of those as well.

```
                              Select Records

Type comparisons, press Enter.  Specify OR to start each new group.
  Tests:  EQ, NE, LE, GE, LT, GT, RANGE, LIST, LIKE, IS, ISNOT...

AND/OR  Field            Test   Value (Field, Number, 'Characters', or ...)
        PHLIB            NLIKE  '#_________'
 AND    PHLIB            NLIKE  'Q_________'




                                                                     Bottom

Field           Text                                                 Len  Dec
PHCSID          -1=See DSPFFD                                          5    0
PHRCEN          Retrieval century:  0=19xx, 1=20xx                     1
PHRDAT          Retrieval date:  year/month/day                        6
PHRTIM          Retrieval time:  hour/minute/second                    6
                                                                    More...
F3=Exit          F5=Report          F9=Insert          F11=Display names only
F12=Cancel       F13=Layout         F20=Reorganize     F24=More keys
```

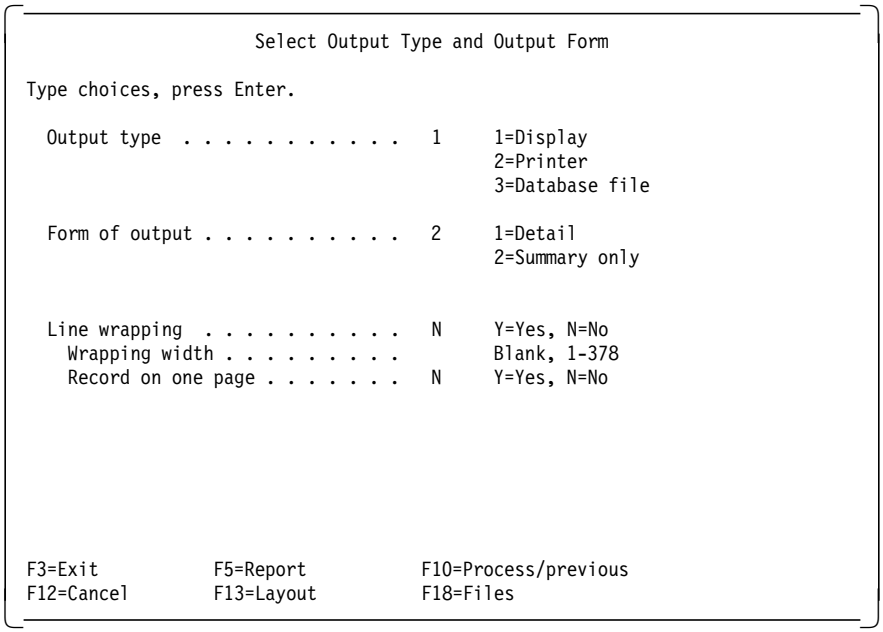
```
                          Select Sort Fields

Type sort priority (0-999) and A (Ascending) or D (Descending) for
  the names of up to 32 fields, press Enter.

Sort
Prty A/D  Field          Text                                      Len  Dec
 1        PHCSID         -1=See DSPFFD                               5    0









                                                                   Bottom
F3=Exit            F5=Report       F11=Display names only   F12=Cancel
F13=Layout         F18=Files       F20=Renumber             F24=More keys
```

Use the summary function to count the number of files for each CCSID.

```
                     Select Report Summary Functions

Type options, press Enter.
  1=Total   2=Average   3=Minimum   4=Maximum   5=Count

---Options---  Field          Text                                 Len  Dec
5              PHCSID         -1=See DSPFFD                          5    0









                                                                   Bottom
F3=Exit       F5=Report       F10=Process/previous    F11=Display names only
F12=Cancel    F13=Layout      F18=Files               F23=Long comment
```

```
                          Define Report Breaks

Type break level (1-6) for up to 9 field names, press Enter.
  (Use as many fields as needed for each break level.)

Break    Sort
Level    Prty  Field          Text                                 Len  Dec
  1        10  PHCSID         -1=See DSPFFD                          5    0









                                                                   Bottom
F3=Exit        F5=Report      F10=Process/previous    F11=Display names only
F12=Cancel     F13=Layout     F18=Files               F23=Long comment
```

Press Enter on the Format Report Break displays and select 2=Summary only on the Form of output field.



Use F3=Exit to exit, save, and run the query to summarize the current CCSID settings of your database files.

Your output should be similar to this report:

```
Line   ....+....1....+.
             Coded
             Character
             Set ID
000001              37
000002 COUNT       880
000003
000004             424
000005 COUNT         8
000006          65,535
000007 COUNT        33
```

This report shows a summary of all different CCSIDs used in user database files. There are 880 files with CCSID 37, eight files with CCSID 424, and 33 files with CCSID 65535.

The 33 files with CCSID 65535 probably do not have to change as it can be caused by the following situations:

- The file is an internally described file.
- The file contains only numeric fields or hexadecimal fields.
- The file is explicitly assigned with CCSID 65535.

Note: If your current release is prior to V3R1, all files might be tagged with 65535.

In this example there are just two more CCSIDs, 37 and 424.

Write the CCSIDs from your query report on your copy of the Planning Primary Language Change table in the **current** column of item 9 (evaluated database CCSIDs).

You now have your worksheet completed with information from your system and target environment.

If your worksheet contains only the CCSID from your current release, you can continue with Section 4.6.1, “Verify the Keyboard” on page 35.

In our example, we found two CCSIDs, 37 and 424. Our worksheet contains 37 in the current column and 424 in the target column for system value QCCSID. In this case, someone might have changed the CCSID of some database files. Another possibility is that the language ID system value or the value in the user profile might have been changed.

If your evaluation shows just the two CCSIDs (the one from the current release and the other from the target release), you can continue with Section 4.6.1, “Verify the Keyboard” on page 35.

If your worksheet shows many different CCSIDs, you have to perform further investigation. You may even have to perform your evaluation for each AS/400 library containing database files. These are the questions and possible actions you must review:

- Are there specific libraries carrying a different CCSID?
  - From the information you have collected about all database files (stored in file ALLFILES in library ZCCSID), you can create a query showing which libraries contains database files with CCSIDs summarized. You may have a report similar to this:

```
Line   ....+....1....+....2..
       Library     Coded
                   Character
                   Set ID
  ..          ..         ..
000040 F0089              37
000041       COUNT        84
000042
000043 F0089          65,535
000044       COUNT        12
  ..          ..         ..
000094 NAVTT              37
000095       COUNT       148
000096
000097 NAV89              37
000098       COUNT       121
000099
000100 NAV89          65,535
000101       COUNT         1
000102
000103 ODBCLIB           424
000104       COUNT         8
000105
000106 PJGAMX            285
000107       COUNT         8
```

    In this example, an unusual CCSID appeared, 285. This CCSID is for U.K. English keyboards. This CCSID might have appeared as a result of receiving an application package from the U.K. and installing it on the Hebrew system without taking care of the initial CCSID setting. This issue is discussed in Chapter 10, “Central Site Considerations” on page 101.

    You should contact the vendor (asking about any impacts) if you change the CCSIDs of database files related to the vendor product.

- Are you currently using a multilingual environment where user profiles may specify the correct setting of the language ID and the CCSID?
  - Investigate user profile settings on your system. You can put the information from all user profiles into a file by using the following command:

    ```
    DSPUSRPRF USRPRF(*ALL) OUTPUT(*OUTFILE) OUTFILE(QTEMP/ALLUSERS)
    ```

    Create and run a query to summarize the language ID and the CCSID setting. Your report may look similar to the following example:

    ```
     Line   ....+....1....+....2....+..
            Language    Coded Character
            Identifier  Set Identifier
     000001 *SYSVAL                 2-
     000002          COUNT         85
     000003
     000004 *SYSVAL                424
     000005          COUNT          1
     000006
     000007 *SYSVAL            65,535
     000008          COUNT          8
    ```

    The -2 value for CCSID means that the system uses the system value. The user profiles with 65535 are usually IBM supplied user profiles and you never change those.

    In this example, we found one user profile specified with the 424 CCSID. This is probably the reason why we discovered database files tagged with CCSID 424 earlier. Any new database files created by this user will have CCSID 424.

Continue with Section 4.6.1, “Verify the Keyboard.”

4.6.1 Verify the Keyboard

Important

In some countries, it might have been a practice to modify device drivers for PC attached keyboards, displays, and printers to get support for the national characters. If your national language version is new in V4R1M0, consult the provider of your PC products.

If you know that the keyboard you are using corresponds to the one you have on your worksheet for the **target** column, you can continue with Chapter 5, “Preparing for Changing Language Sensitive Settings” on page 37.

Otherwise, here are some hints to evaluate your system for the keyboard. You can display the device description for your console by using the following command:

```
DSPDEVD QCONSOLE
```

The display looks similar to the following display:

```
                        Display Device Description                SYSTEM01
                                                                27/05/97  11:23:00
Device description . . . . . :   QCONSOLE
Option . . . . . :   *BASIC
Category of device . . . . . :   *DSP

Device class . . . . . :   *LCL
Device type  . . . . . :   3476
Device model . . . . . :   EA
Port number  . . . . . :   0
Switch setting . . . . . :   0
Online at IPL  . . . . . :   *NO
Attached controller  . . . . . :   QCTL
Keyboard language type . . . . . :   HEB
Character identifier . . . . . :   *KBDTYPE
Allow blinking cursor  . . . . . :   *YES
Print device . . . . . :   *SYSVAL
Output queue . . . . . :   *DEV
                                                                        More...
Press Enter to continue

F3=Exit   F11=Display keywords   F12=Cancel
```

On this display, Keyboard language type is the information you need. If the information (in this case HEB) corresponds to the **target** column on your worksheet for keyboard, you have found the correct setting for your target release.

If you have more than one type of keyboard attached to your system, you need to verify those keyboards against the database files you have on your system.

Chapter 5. Preparing for Changing Language Sensitive Settings

This chapter describes the necessary actions to perform when you prepare to change your AS/400 system′s primary language.

Before you use this chapter, you must have completed Chapter 4, “Planning for Changing Language Sensitive Settings” on page 23.

5.1 Preparing to Change CCSID of Database Files

If you need to change the CCSID of your database files, you can use the example program and command described in Appendix E, “Program and Command CHGLIBCCS” on page 133, to automate this task. Create the program and command on your system.

On your copy of the worksheet in Appendix D, “Worksheet for Planning Primary Language Change” on page 131, the Planning CCSID Change With CHGLIBCCS Command table is designed to document the changes you have to make to CCSIDs of database files.

In the Planning Primary Language Change table, you have written the evaluated database CCSIDs in the current column. If that value corresponds to the assumed CCSID for database files in the current column, this is your From CCSID. The To CCSID is the target column for the assumed CCSID for database files.

In this case, your Planning CCSID Change With CHGLIBCCS Command table can have just one entry, specifying *ALL for library, the From CCSID, and the To CCSID you just found.

The system we evaluated shows two CCSIDs, 37 and 424. However, the 424 CCSID corresponds to the target language assumed CCSID, so these files do specify the correct CCSID, and no action is needed.

The other CCSID is 37. This CCSID corresponds to the current language assumed CCSID. Therefore, our conclusion is that all files carrying the 37 CCSID have to change to 424. In the Planning CCSID Change With CHGLIBCCS Command table, you would write the following entry:

- Library: *ALL
- From CCSID: 37
- To CCSID: 424

If you have other CCSIDs, you need to look at the current CCSID tagging of database files on a library basis. This issue is discussed in Section 4.6, “Evaluate Database Information” on page 29.

Use the table to document the individual conversion tasks you have to make for the CCSIDs and libraries.

The table is your input to the changing process you will perform later.

5.2 Other Information

Similar to a release update, it is important that you have documented the changes you make to IBM supplied objects. We recommend documenting changes to IBM supplied objects in a CL program that can be used when you perform a release update.

Perform the following tasks prepare for changing your primary language.

1. Document the following information for IBM supplied objects:
   - All subsystem descriptions
   - Printer file descriptions
   - All system values
   - Changed command defaults
2. Check the software you received.
3. Make sure that you use the *Software Installation*, SC41-5120, for the release you are currently using for changing the primary language and review the information in the Preparing To Install Software chapter.

   Pay close attention to the information in the following list:
   - Availability of disk space, as new software requires additional disk space.
   - Verify the integrity of needed user profiles QSECOFR and QLPINSTL. Do not forget to make sure the QSECOFR user profile is not changed and has all special authorities. If not, use the CHGUSRPRF QSECOFR command to have these authorities.
   - Find the list with the system password that came from the factory when you ordered your AS/400 system (it might be necessary to have this information during the installation procedure). Check that you have all software distribution media required to complete the task.
   - Also check the status of the tape or optical device unit (they should be operational). Clean the tape device unit before you start the installation.
   - Ensure the system values QSYSLIBL and QUSRLIBL do not contain any QSYS29xx or other IBM supplied licensed program libraries (usually their names start with Q).
   - Change the system value QALWOBJRST to *ALL.
   - Be sure you have completed or canceled all pending re-synchronizations of commitment control definitions in your system.
4. Read about the F type keyboard issue on page 3-3 through page 3-24 of *Software Installation*, SC41-5120.

   Make sure that your console device is configured to support the default code page of the primary language you are going to install. Change the console device to one that supports the code page of the new primary language before installing V4R1M0.

   While F type keyboards are supported for many national language versions, they are not supported for Czech, Farsi, Hungarian, Russian, Polish, Slovakian, or Thai. If you change your primary language to one of these national language versions, the system console cannot have an F type keyboard. Change the system console to a device that does not have an F

type keyboard before installing any of these national language versions as a primary language. If you do not, an error occurs and the installation fails.

5.2.1 Secondary Language Currently Installed

If you currently have the language that you intend to use as your primary language as a secondary language, be aware of the following steps:

1. You must delete the secondary language before you change your current primary language.
2. You must remove any references to that secondary language (for example, QSYS29xx) in library lists, job descriptions, and programs.
3. If you are using Client Access/400 original clients, review Section 7.3.8, “Considerations for Changing the NLS Option on the PC” on page 53.

5.3 Obtain Preventive Service Planning Information

You should obtain Preventive Service Planning (PSP) information for the target release of OS/400. You can do that in one of the following two ways:

- Use the Electronical Customer Support (ECS) SNDPTFORD SF99VRM command where VRM is the version, release, and modification of the target OS/400.
- Use Internet to get the PSP at URL:

  `http://as400service.rochester.ibm.com`

Chapter 6. Changing Language Sensitive Settings

This chapter gives an explanation of which procedures to follow when you change a primary language.

Before you use this section, be sure to complete Chapter 4, “Planning for Changing Language Sensitive Settings” on page 23 and Chapter 5, “Preparing for Changing Language Sensitive Settings” on page 37.

We recommend using the following steps when you change your AS/400 system primary language.

- Save the entire system before you change the primary language.
- Use the instructions in the *Software Installation* book to change the primary language.
- Make any necessary changes to the CCSIDs of database files.
- Verify the Client Access settings.
- Verify the system settings and applications.
- Save the entire system after you change the primary language.
- Notify your software provider that you have a new primary language.

6.1 Save Entire System

You must save all of the system information on tapes before you start changing the system′s primary language. In this case, if a failure occurs, you can use the tapes to load the information back into the system. The tapes you create in this task become your master copy (keep them in a safe place). Have enough blank tape cartridges to save the system.

Use the following steps to complete the task.

1. Calculate the amount of disk space used to store the data on the system by using the WRKSYSSTS command. If, for example, you see that your system storage ASP is 10Gb and is 75% full, you need to save 7.5Gb of information. Check the capacity of the tape cartridges you are using for the save procedure and calculate the amount of tapes needed. All tape cartridges should have the same density (you cannot combine QIC120, QIC525, or other cartridges of a different density during the save procedure).

   The best practical way to know the amount of tapes needed for a system save is to check your last saving procedure tape set and add some extra tapes if new information was added to the system.

2. Define the name and status of the tape device for the save procedure by using the WRKCFGSTS CFGTYPE(*DEV) CFGD(*TAP) command. We recommend using an alternate IPL tape unit (as default, it is TAP01) for the save system procedure.

3. Initialize the tapes. Find the blank tape cartridges to use for saving the system and label them SAVT01, SAVT02, and so on (or other names you choose to use). First, load the tape into the tape unit. On the command line of the AS/400 Main Menu, type GO TAPE and choose option 2 from Tape menu. On the Initialize Tape display, type the following information:

- Tape device **TAP01**
- New volume identifier **SAVT01**
- Check for active files ***NO**
- Tape density ***CTGTYPE**

Leave all of the other values as they appear on the display and press the Enter key. After the first tape is initialized, the Tape menu appears with the following message:

```
Volume SAVT01 prepared for operation with owner ID *BLANK
```

Repeat the procedure for all of the other tapes you calculated for a save procedure.

4. Save the system information on tape. On the command line, type GO SAVE and choose option 21 (Save entire system). The Save the Entire System display appears and explains what happens during the save process. After reading this information, press the Enter key. The Specify Command Defaults display appears. Type the following information and press the Enter key:

   - Tape devices **TAP01**
   - Prompt for commands **N**
   - Check for active files **N**

   If any display appears, follow the instructions on that display. You may be asked to load another tape cartridge. After successfully completing the save procedure, the Save display appears with the following message:

   ```
   Save or restore option completed successfully
   ```

   If an error message appears, follow the problem definition and recovery procedure in the *Backup and Recovery*, SC41-5304.

6.2 Changing Your Primary Language

Use the *Software Installation*, SC41-5120, manual for changing your primary language. Carefully follow the instructions in the Changing Your Primary Language Using Secondary Language Media chapter (Chapter 8).

6.3 Make Changes to CCSIDs of Database Files

On your copy of the worksheet in Appendix D, “Worksheet for Planning Primary Language Change” on page 131, you have written the name of the libraries (or maybe *ALL) and the CCSIDs that have to be changed.

Use the information to run the CHGLIBCCS command.

Using our example, this command is:

```
CHGLIBCCS LIB(*ALL) FCCS(37) TCCS(424)
```

Note: The user running the CHGLIBCCS command must have adequate authority to perform changes to the physical files in the libraries.

Make sure to review and save (or print) the job log after running the command. Resolve any messages that may have occurred.

6.4 Verify CA/400 Settings

If you have Client Access/400 licensed programs installed on your system, you need to verify CA/400 settings. Please refer to Chapter 7, “NLS considerations for Client Access/400” on page 47 for necessary considerations to complete this step.

Also use the following references to locate the correct publication for verifying the settings. If the Client Access/400 you are using is for:

- Windows 95, see *Client Access/400 for Windows 95/NT - Setup*, SC41-3512
- Windows 3.1, see *Client Access/400 for Windows 3.1 - Setup*, SC41-5534

  The preceding manual replaces *Client Access/400 for Windows 3.1 - Getting Started* and also includes the installation information that was distributed in the informal documentation shipped with the product (USLT156U).
- DOS with Extended Memory, see *Client Access/400 for DOS with Extended Memory Setup*, SC41-3500
- DOS, see *Client Access/400 for DOS Setup*, SC41-3556
- OS/2 1.3, see *Client Access/400 for OS/2 Setup*, SC41-3520
- Optimized for OS/2, see *Client Access/400 Optimized for OS/2 - Setup*, SC41-3510

6.5 Verify System Values and Applications

We recommend using the following steps to verify your system after you change the primary language.

Print your system values using the following command:

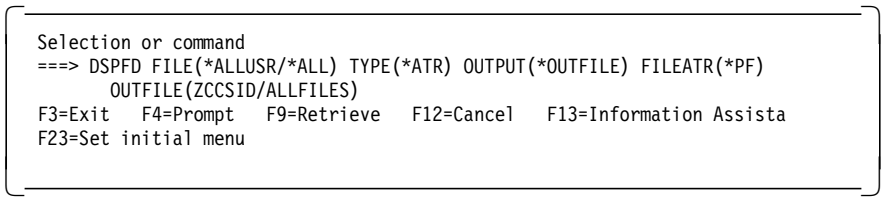
```
WRKSYSVAL *ALL OUTPUT(*PRINT)
```

Use your filled in copy of the worksheet in Appendix D, “Worksheet for Planning Primary Language Change” on page 131 and compare the following system values with the **Target** column:

- QCHRID
- QKBDTYPE
- QLANGID

Collect information about your database files to ensure the CCSIDs are correctly changed:

- Retrieve the information using the DSPFD command:



- Summarize the CCSIDs used in your database files by running the query you have created earlier:

  ```
  RUNQRY ZCCSID/CCSIDSUM
  ```

The summary should show those CCSIDs you have planned for your database files.

You can also verify your user applications and any clients that access your database files.

If all your database files now have the CCSID of your target release, you should also set QCCSID to your target value.

6.6 Save Entire System

Repeat the procedure described in Section 6.1, “Save Entire System” on page 41.

6.7 Notify Software Provider about Your New Primary Language

After you have finished all of the steps, do not forget to inform your software provider (IBM or business partner) that you have changed your software primary language so they can change your current status. This is important for future updates of your AS/400 system software.

Do not forget to delete your work library (ZCCSID) or whatever you called it.

Part 2. Background Information

Part 2 contains language oriented background information within the areas of Client Access/400, NLS considerations for TCP/IP, the printing environment, and some central site development considerations.

- Chapter 7 discusses the impact of NLS options within the area of Client Access/400.
- Chapter 8 contains information about NLS considerations in TCP/IP environments.
- Chapter 9 contains information about how to verify your printing environment when printing in a multilingual environment.
- Chapter 10 discusses considerations for multilingual system setup and central site application providers when delivering their applications to a multilingual environment.

Chapter 7. NLS considerations for Client Access/400

Client/server computing and the AS/400 system have a long history with PC Support/400. Client Access/400 was developed to help businesses better integrate personal computers with midrange systems. It has evolved into one of the most popular products used on the AS/400 system and has allowed businesses using AS/400 systems to start using client/server computing.

Client Access/400 provides many of the basic client/server functions such as file transfer, file serving (shared folder), remote database query, remote printing, and remote application access.

Client Access/400 contains both AS/400 server code and client code for each of the supported client environments. Each function in Client Access/400 uses both client and server programs. The Client Access/400 transfer function is an example of client/server computing. It is a client program on the PC communicating with a server program on the AS/400 system to get the requested file and send it to the PC in a format the client expects.

7.1 Original and Optimized Servers

The support that was provided with PC Support/400 is called the original support. There are original client products for DOS, DOS with Extended Memory, and OS/2, and there are original servers for those clients in the operating system:

- Transfer Function Server
- Remote SQL Server
- Data Queue Server (Original)
- Message Function Server
- License Management Server
- Virtual Print Server
- Shared Folder Server
- Remote Command Server

These servers are described in Chapter 3, ″Original Servers″ in topic 3.0 in the *AS/400 Client Access Host Servers*, SC41-5740.

The support first added in V3R1 is called the optimized support. There are four optimized clients. The Client Access for Windows 3.1 and the Client Access Enhanced for Windows 3.1 run as a native Windows 16-bit applications. The Client Access for Optimized OS/2 and the Client Access for Windows 95/NT run as 32-bit applications. There are optimized and some original servers for those clients in the operating system:

- File Server
- Database Server
- Data Queue Server
- Network Print Server
- Central Server
- Remote Command and Distributed Program Call Server
- APPC Password Management Server
- Signon Server

These servers are described in Chapter 4, ″Optimized Servers″ in topic 4.0 in the *AS/400 Client Access Host Servers*, SC41-5740.

The following tables map some Client Access/400 functions to the OS/400 servers. For a complete list, refer to the *AS/400 Client Access Host Servers*, SC41-5740.

Table 9. Client Access for DOS, Client Access for DOS with Extended Memory and Client Access for OS/2 Functions

Client Function	**OS/400 Server Used**
File Transfer Function	Transfer Function Server
Remote SQL APIs	Remote SQL Server
Database Query GUI	Database Server (Optimized)
Shared Folders	File Server (Optimized)
Message Function	Message Function Server
Data Queue APIs	Data Queue Server (Original)
Virtual Print	Virtual Print Server
Remote Command Functions	Remote Command Server

Table 10. Client Access for Windows 3.1 and Client Access Enhanced for Windows 3.1

Client Function	**OS/400 Server Used**
File Transfer Function	Transfer Function Server
Remote SQL APIs	Remote SQL Server
ODBC Driver	Database Server (Optimized)
Shared Folders	File Server (Optimized)
Message Function	Message Function Server
Data Queue APIs	Data Queue Server (Original)
Virtual Print	Virtual Print Server
Remote Command Functions	Remote Command Server

Table 11. Client Access Optimized for OS/2

Client Function	**OS/400 Server Used**
File Transfer Function GUI	Transfer Function Server
Database Access APIs	Database Server (Optimized)
ODBC Driver	Database Server (Optimized)
Database Query GUI	Database Server (Optimized)
Network Drives	File Server (Optimized)
Data Queue APIs	Data Queue Server (Optimized)
Network Print	Network Print Server
Remote Command Functions	Remote Command and Distributed Program Call Server (Optimized)
Distributed Program Call	Remote Command and Distributed Program Call Server (Optimized)

Table 12. Client Access for Windows 95/NT	
Client Function	**OS/400 Server Used**
File Transfer Function GUI	Transfer Function Server
Database Access APIs	Database Server (Optimized)
ODBC Driver	Database Server (Optimized)
Database Query GUI	Database Server (Optimized)
Network Drives	File Server (Optimized)
Data Queue APIs	Data Queue Server (Optimized)
Network Print	Network Print Server
Remote Command Functions	Remote Command and Distributed Program Call Server (Optimized)
Distributed Program Call	Remote Command and Distributed Program Call Server (Optimized)

7.2 Basic Principles

Basically, Client Access/400 applications follow the same CCSID logic described in previous chapters. Since an AS/400 host server job mediates the data transfer between the client program and the OS/400 object, CCSID conversions might occur in two places:

- Server conversion - between the OS/400 object and host server job
- Client conversion - between the client program and the host server job

The CCSID conversions between the OS/400 object and the host server job always follow the general OS/400 CCSID conversion logic. On the other hand, the conversion logic between the client program and the host server job is client program dependent.

This assumes that:

- The AS/400 object's CCSID matches the character data in the object.
- The AS/400 object's CCSID and the PC code page are compatible (the character set of the AS/400 object's CCSID matches the character set of the CCSID associated with the PC code page).

There are two important parameters of the data exchange process:

- What is the CCSID of the AS/400 host server job.
- Which translation tables are being used for the client conversion.

In the following sections, we look at how these parameters influence the data exchange process.

7.3 Original Clients

The host server job for the original clients is set to the CCSID value of the user profile that initiates the CA/400 session.

The conversion tables between the host EBCDIC CCSID and client ASCII code page are set by executing the INZPCS command with the proper values. The following example is for the Hebrew language only.

```
                    Initialize Client Access/400 (INZPCS)

Type choices, press Enter.

Keyboard type  . . . . . .   NCB          *DFT, AGB, AGI, ALI, BGB...
ASCII code page number . . . . .   862     *DFT, 437, 850, 851, 852...
EBCDIC code page number  . . . .   424     *DFT, 037, 273, 277, 278...
Language feature code  . . . . .   2961    *DFT, *ALL, 2911, 2912...

                                                                  Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

This essentially creates two tables:

- EBCDIC to ASCII table - PCSETOA.TBL
- ASCII to EBCDIC table - PCSATOE.TBL

These tables are in the MRI folder of the corresponding original client on the AS/400 system. These tables are later downloaded to the client as part of the Client Access/400 installation or update. So for a given AS/400 system and a given MRI, there are constant conversion tables for all original clients.

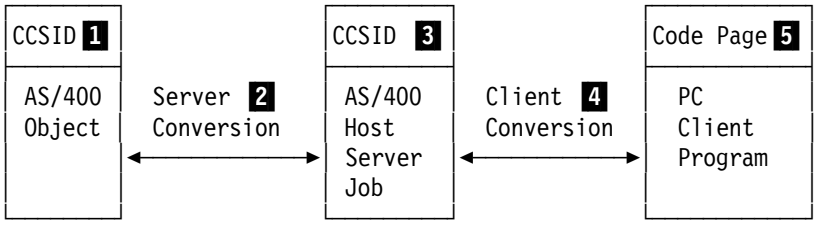


Figure 4. CCSID Conversion Process for Client Access/400 Original Clients

Since the client conversion 4 uses a fixed table for a given AS/400 system and a given MRI, the character data in the PC 5 must match the ASCII code page

used by Client Access/400. To identify which code page Client Access/400 is using (as set by the INZPCS command), issue the following commands from an AS/400 system command line:

- For the primary language:

  ```
  DSPDTAARA   DTAARA(QUSRSYS/QINZPCSDA)
  ```

- For a secondary language:

  ```
  DSPDTAARA  DTAARA(QSYSxxxx/QINZPCSDA)
  ```

Where xxxx is the feature code of the language. For a list of feature codes, see table Table 29 on page 123.

The ASCII code page is the first value (the first four bytes of the data area).

The fixed table used by the client conversion **4** also means that if CCSID support is being used on the AS/400 system, the CCSID parameter on the Client Access/400 user profile must match the EBCDIC code page used by Client Access/400 because this value determines the host server job CCSID **3**. To find out the EBCDIC code page used by Client Access/400, issue the preceding commands.

The EBCDIC code page is the second value (byte 6 to byte 9 of the data area).

If CCSID support is not being used on the AS/400 system, the CCSID parameter on the Client Access/400 user profile should be set to *HEX. Note that depending on the NLS settings on the AS/400 system, the CCSID parameter on the Client Access/400 user profile might be set explicitly to *HEX or implicitly by pointing to QCCSID system value.

.

Table 13. Example of ODBC with Different CCSID Values, INZPCS Parameters where ASCII Code Page = 862 - Hebrew PC Data, EBCDIC Code Page = 424 - Hebrew

AS/400 Object CCSID	**User Profile CCSID**	**Remarks**
424 - Hebrew EBCDIC	*HEX	CCSID support disabled, no server conversion, and correct client conversion to 862.
424 - Hebrew EBCDIC	424	CCSID support enabled, same CCSID - no server conversion, and correct client conversion to 862.
*HEX	Any valid value	CCSID support disabled, no server conversion, client conversion from 424 to 862.
37	*HEX	CCSID support disabled, no server conversion, client conversion from 424 to 862, might lose data integrity.
37	424 - Hebrew EBCDIC	CCSID support enabled, server conversion from 37 to 424, client conversion from 424 to 862, only graphic characters common to both 37 and 424 preserve data integrity.

7.3.1 Shared Folders Function

The Shared Folders function converts all file and directory (folder) names to the EBCDIC code page 500 (International Latin-1). This allows all personal computers to access folder documents regardless of the language and code page in use. In a multilingual environment, IBM recommends that you use only characters from the invariant character set.

The Shared Folders function does not alter data stored in folders. This means that personal computer files are stored in ASCII using the code page of the originating personal computer.

7.3.2 File Transfer Function

File Transfer Function CCSID conversions work as described in Figure 4 on page 50. In the Windows Clients data transfer, there are two options:

- ANSI transfer
- ASCII transfer

The transfer from the server to the client is always from EBCDIC to ASCII. If you are doing an ANSI file transfer, a second step is performed on the client to convert from ASCII to ANSI. For bidirectional languages, the ANSI file transfer needs to convert from visual to logical representation as described in Section 10.6, “Bidirectional Languages” on page 113. This is handled by calling a bidirectional version of the File Transfer DLL, EHNTFWB.DLL, instead of EHNTFW.DLL.

7.3.3 Remote SQL

Remote SQL supports CCSIDs through the attributes specified on the user profile of the job. The user profile is the same user profile used by the Client Access/400 router or the user profile specified as a parameter on the EHNRQSTARTSEC application program interface (API).

For data integrity, it is important that CCSID support be used if your environment has multiple languages using different code pages.

7.3.4 Message Function

The Message Function returns messages to the personal computer in the language used by the AS/400 server job.

7.3.5 Virtual Printer Function

When the Virtual Printer Function sends data to the AS/400 system, it maps the ASCII code points contained in the data to be printed to the equivalent EBCDIC code points. To print correctly, the personal computer code page used to create the data must be compatible with the Client Access/400 ASCII-to-EBCDIC translation table.

7.3.6 Workstation Function - 5250 Emulation

The language-specific control files for the Workstation Function (WSF) contain keyboard layouts, EBCDIC-to-ASCII translation tables, and other files used by national language support. Specific national language support exists for Thai character composition and for the bidirectional languages of Arabic and Hebrew.

The mapping table that WSF uses to perform EBCDIC-to-ASCII translation for displays substitutes the most appropriate character (the ″best fit″) when exact one-to-one mapping is not possible. This means that when an EBCDIC code point does not have a corresponding ASCII code point, the table provides a suitable substitution character. Because the actual data is maintained in EBCDIC, the data sent or received by the host may appear slightly different on the display.

Because WSF uses these mapping tables independent of the rest of Client Access/400, it is possible to emulate more than one language on a personal computer at a time. This is possible if exact one-to-one mapping is not required or if the two languages share the same code page. You can use the WSF configuration program (CFGWSF.EXE) to modify the mapping tables.

7.3.7 Workstation Function - Printer Emulation

WSF Printer Emulation also uses the ″best fit″ mapping tables to perform EBCDIC-to-ASCII translation for printers. This means that when an EBCDIC code point does not have a corresponding ASCII code point, the table provides a suitable substitution character. Since the actual data is maintained in EBCDIC, the data sent or received by the host system may appear slightly different on the printer.

Because WSF printer emulation uses these tables independent of the rest of Client Access/400, it is possible to use more than one language on a personal computer at a time. This is possible if exact one-to-one mapping is not required or if the two languages share the same code page. You can use the WSF configuration program (CFGWSF) to modify the mapping tables.

Another approach for multiple language printing that does not require sharing a code page is to use the host print transform function provided by the AS/400 system.

7.3.8 Considerations for Changing the NLS Option on the PC

You can change the NLS option on the PC by using the CFGPCS command. For details, consult the *Client Access/400 for DOS Setup*, SC41-3556. The result is a creation of a new MRI directory on the PC. To use the new MRI directory, you need to edit the EHNL.BAT file in the Client Access directory on the PC to set the value of the ″EHNL″ variable to the corresponding language feature code.

```
@SET EHNL=2924
```

Figure 5. Sample EHNL.BAT for Language Feature 2924 - U.S. English

Note, however, that by default, part of the Client Access functions are not downloaded to the PC so when you use these functions, the programs and MRI for them need to be available on the shared folders on the AS/400 system.

The MRI folders for U.S. English always exist on the AS/400 system but all other MRI folders are created as part of the primary or secondary language installation process and deleted as well as part of the changing primary language or the deleting secondary language process.

Note!

After changing your primary language from a non-U.S. English version to any other version, you might find yourself in a situation where your PCs use an NLV that does not exist on the AS/400 shared folders. In this case, it might even happen that you cannot use the Client Access update function or configuration function because, by default, they do not exist on the PC, and there is a different MRI in the shared folders.

A way to solve this problem is to install the previous primary language as secondary (at least temporarily until all of the clients change to use the new MRI).

7.4 Optimized Clients

In optimized clients, the client conversion algorithm was changed to perform the correct conversions between the AS/400 object CCSID and the client code page by changing the host server CCSID and, if they don't already exist on the client, downloading the conversion tables to the client at run time.

7.4.1 Host Server CCSID

The AS/400 host server job CCSID is set according to the normal rules for setting a job CCSID with the exception that *HEX CCSID is not a valid option. The user profile being used for AS/400 host server jobs is QUSER so the CCSID 1 and LANGID are determined by the QUSER settings.

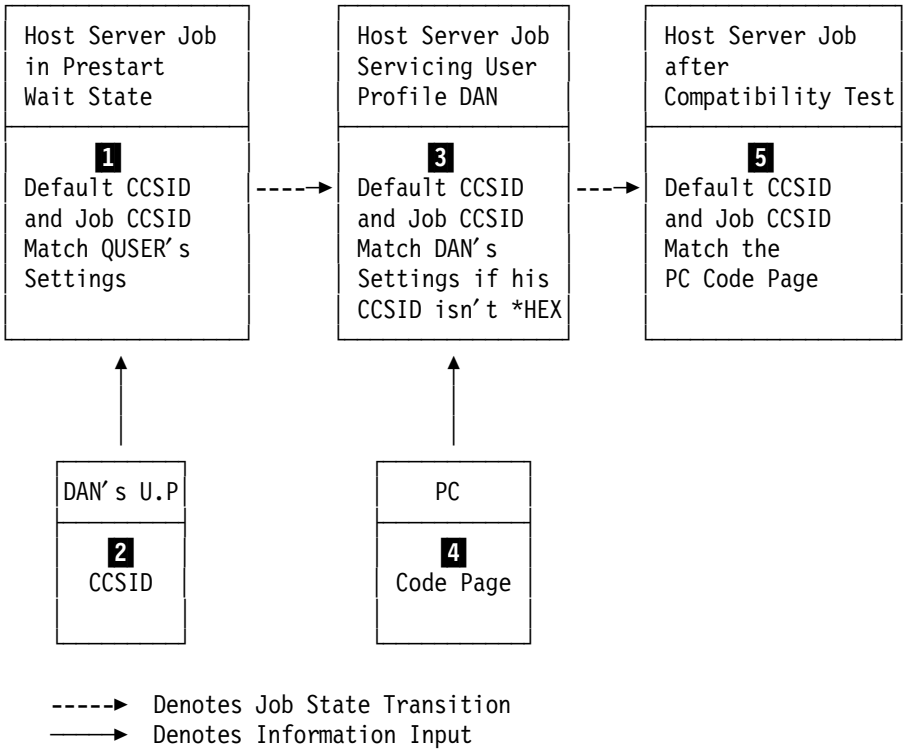


Figure 6. Host Server Job CCSID Determination Process

.

Table 14. Example of Database Server Prestart Job CCSID Settings when Job is in Prestart Wait State

QUSER		Host Server Job		Remarks
Language ID	CCSID	Default CCSID	CCSID	
HEB - Hebrew	*HEX	424	424	Server job CCSID matches the default CCSID for QUSER language ID.
Any valid language ID	424	424	424	Server job CCSID matches QUSER′s CCSID.
ENU - U.S. English	*HEX	37	37	Server job CCSID matches the default CCSID for QUSER language ID.
Any valid language ID	37	37	37	Server job CCSID match QUSER′s CCSID.

When a client application connects to its host server job, the client application sends the user ID **2** of the user that logged in to the AS/400 system and the PC code page **4** to the host server job. If the user profile CCSID is different than *HEX, the corresponding host server job CCSID **3** is set to the user profile CCSID.

Table 15. Example of Database Server Prestart Job CCSID Settings when Job is Servicing a Connected User

Host Server Job CCSID when in Prestart Wait State	User Profile CCSID of Connected User	Host Server Job CCSID when Servicing Connected User (before running the compatability tests)
424 - Hebrew	*HEX	424
424 - Hebrew	424	424
424 - Hebrew	37	37

A compatibility test is run between the PC code page and the host server job code page. Both code pages are compatible if the character set of the host server code page equals the character set of the CCSID associated with the PC code page. For a table of associated CCSIDs, look at table A-4 in Appendix A of the *International Application Development*, SC41-5603.

With Client Access for Windows 95/NT, if the code pages are compatible, the user can, using Data Transfer, specify the results to be stored in an ASCII file or an ANSI file. So the conversion table used depends upon the situation.

Host code page ⇔ PC code page

Host code page ⇔ PC ANSI code page

On the OS/2 clients, there is only one choice (OS/2 does not have ANSI code page).

If the code pages are not compatible, the host server job CCSID **5** is set to the value of the CCSID associated with the PC code page and the tables used during data conversion are as previously described.

Table 16. Example of Database Server Prestart Job CCSID Settings after Compatibility Tests

Host Server Job CCSID when Servicing Connected User	**1** Character Set of Host Server Job CCSID	PC Code Page	**2** EBCDIC CCSID Associated with PC Code Page	**3** Character Set of Associated CCSID	Final Host Server CCSID
424 - Hebrew	941	862 - Hebrew PC data	424 - Hebrew	941	424
277 - Denmark, Norway	697	850 - Multilingual Latin1 PC data	500 - International Latin1	697	277
285 - United Kingdom	697	862 - Hebrew PC data	424 - Hebrew	941	424

The character set of the host server job CCSID **1** and **3** in Table 16 is determined according to table G-2 in the *National Language Support*, SC41-5101.

The EBCDIC CCSID associated with the PC code page **2** is determined according to table A-4 in the *International Application Development*, SC41-5603, by checking the column for EBCDIC (encoding 1100).

By examining Table 16, you can see that in the first row, the code pages are compatible because the associated CCSID of the PC code page equals the host server job CCSID. In the second row, this is not always the case since the associated CCSID of the PC code page differs from the host server CCSID but both CCSIDs share the same character set; hence, they are compatible. The third row demonstrates incompatible CCSIDs.

As shown in the example, the net result of the host server job CCSID setting algorithm is that the host server CCSID is always compatible with the PC code page. Hence, when using the optimized clients, be careful with the conversion between the AS/400 object CCSID and the host server job CCSID.

Table 17 (Page 1 of 2). Example of Server Conversion using Client Access for Windows 95/NT Client

AS/400 Object CCSID	Windows 95 Version Used (Code Page)	Host Server CCSID	Remarks
424 - Hebrew	Hebrew (862)	424	CCSID support enabled but the CCSIDs are matching, no server conversion.

Table 17 (Page 2 of 2). Example of Server Conversion using Client Access for Windows 95/NT Client

AS/400 Object CCSID	Windows 95 Version Used (Code Page)	Host Server CCSID	Remarks
37 - U.S. English	Hebrew (862)	424	CCSID support enabled, server conversion is performed, only graphic characters common to both 37 and 424 preserve data integrity.
*HEX	Hebrew (862)	424	CCSID support disabled, no server conversion, no client conversion .unless force conversion is used.
424 - Hebrew	U.S. English (437)	37	CCSID support enabled, server conversion is performed, only graphic characters common to both 37 and 424 preserve data integrity.
37 - U.S. English	U.S. English (437)	37	CCSID support enabled but the CCSIDs are matching, no server conversion.
*HEX	U.S. English (437)	37	CCSID support disabled, no server conversion, no client conversion unless force conversion is used.

As shown in the second and fourth lines of Table 17 on page 56, when the AS/400 object CCSID does not match the host server CCSID, graphic characters data integrity is preserved only for the characters common to both code pages. If this is the case as in the second line in Table 17 on page 56 and the AS/400 object contains only graphic character data that is common to code page 424, consider changing the AS/400 CCSID tag to 424 to disable the unnecessary conversion.

The first and fourth lines in Table 17 on page 56 demonstrate the situation where the same user tries to get the same AS/400 object from two PCs with different language versions of Windows 95. In this case, if the AS/400 object contains graphic characters that are not common to both 37 and 424, the data on the two PCs might be different.

7.4.2 Forced Client Conversion for *HEX AS/400 Objects

When the AS/400 object is tagged as *HEX, no conversion takes place, either for a server conversion or a client conversion. The result is that the PC data is EBCDIC hex data. For data transfer and ODBC, there is a way to force a client conversion between the host server job CCSID and PC code page.

7.4.2.1 Force Conversion for File Transfer

Starting from Windows 95/NT (V3R1M2), Data Transfer allows conversion between the CCSID *HEX and the PC code page. The host server job CCSID is used for the conversion. This conversion is controlled by creating an INI file named CWBTFR.INI in the directory Windows where was installed. The INI file must contain the section heading and a ForceTranslation line.

```
[Client Access Data Transfer]
ForceTranslation=1.
```

*Figure 7. Sample CWBTFR.INI File for Translation of *HEX Data*

```
[Client Access Data Transfer]
ForceTranslation=0.
```

*Figure 8. Sample CWBTFR.INI File for No Translation of *HEX Data*

If the INI file is not found, or the correct section and value names are not present, Client Access Data Transfer defaults to no translation of *HEX data.

7.4.2.2 Force Conversion for Client Access ODBC

In the Client Access ODBC setup, you can choose to force translation of *HEX (65535) tagged AS/400 files or columns by setting this option in the ″Other″ property sheet.

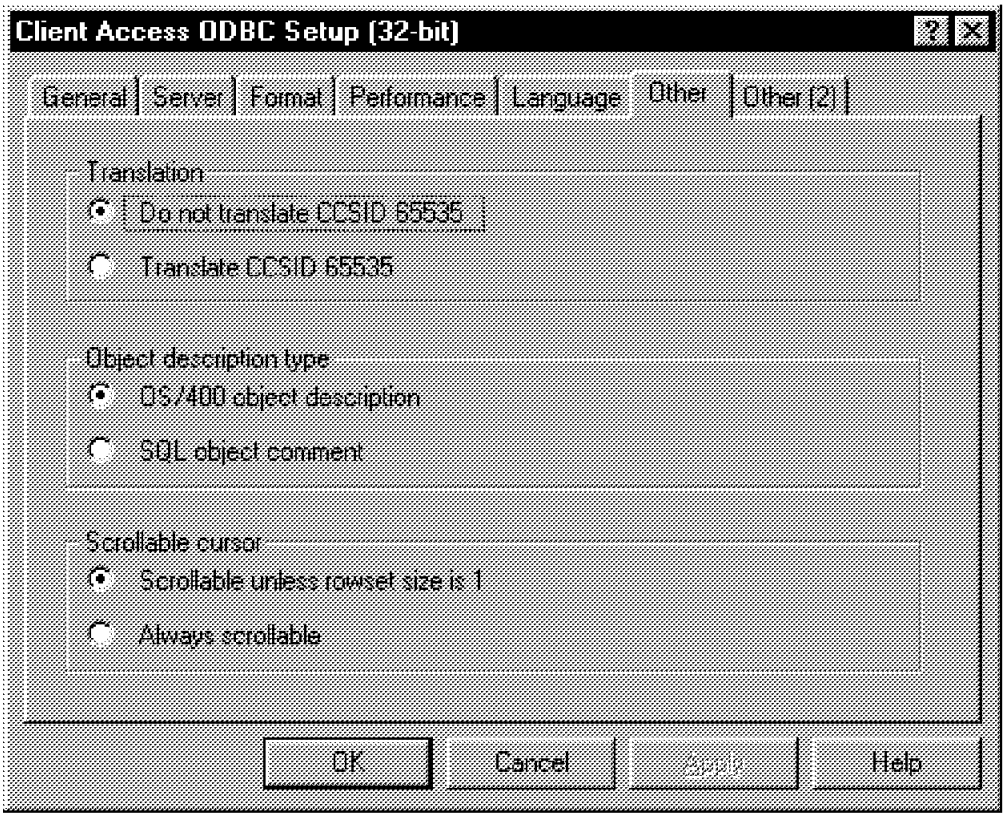


Figure 9. The Client Access ODBC Setup ″Other″ Property Sheet

Note!

Enable translation of *HEX data only if you are confident that data contained within the columns tagged with *HEX CCSID are in a defined CCSID that matches the host server job CCSID. Accessing *HEX CCSID data from a host server job with a CCSID that does not match the data may corrupt data within the database file.

Columns tagged with the *HEX CCSID are designed to not be converted when transferring to and from the PC. Use force conversion only when it is not possible to change the column or file CCSID from *HEX. Every attempt should be made to appropriately tag the data with the correct CCSID.

7.4.3 Downloading the Conversion Tables

The client program that is invoked to download the conversions tables from the AS/400 system is:

```
CWBNLTBL.EXE sourceCP  targetCP [host]
```

This program is invoked automatically if the required tables do not exist in the Client Access/400 directory. This program downloads the conversion tables between the source code page (sourceCP) and the target code page (targetCP) from the AS/400 system specified (host) and adds an informational entry to the Client Access/400 history log. If no system is specified, it uses the managing AS/400 system.

Note!

When no managing AS/400 system is defined and the required conversion tables do not exist in the Client Access directory on the PC, the tables are not downloaded from the AS/400 system and you do not get the desired results. You can check if this is happening in the history log in the Client Access service folder. The messages are logged by the NLS component.

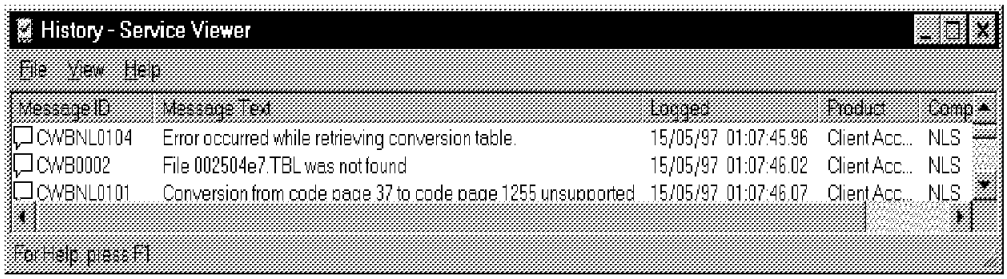


Figure 10. Windows 95/NT Client Access/400 History Log

7.4.4 How PC Code Page is Determined

The PC code page is the basis for the client conversion. Hence, it is important to understand how this value is retrieved.

7.4.4.1 Windows 95 Code Page

Windows 95 uses two code pages (the OEM code page and the ANSI code page). The personal computer OEM code page is determined by calling the WIN32 API GetOEMCP, which returns the value of ″OEMCP″ stored in the Windows 95 registry.

The personal computer ANSI code page is determined by calling the WIN32 API GetACP, which returns the value of ″ACP″ stored in the registry. Both values are located in the Windows 95 registry at:

″HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\Nls\Codepage″

See Figure 11.

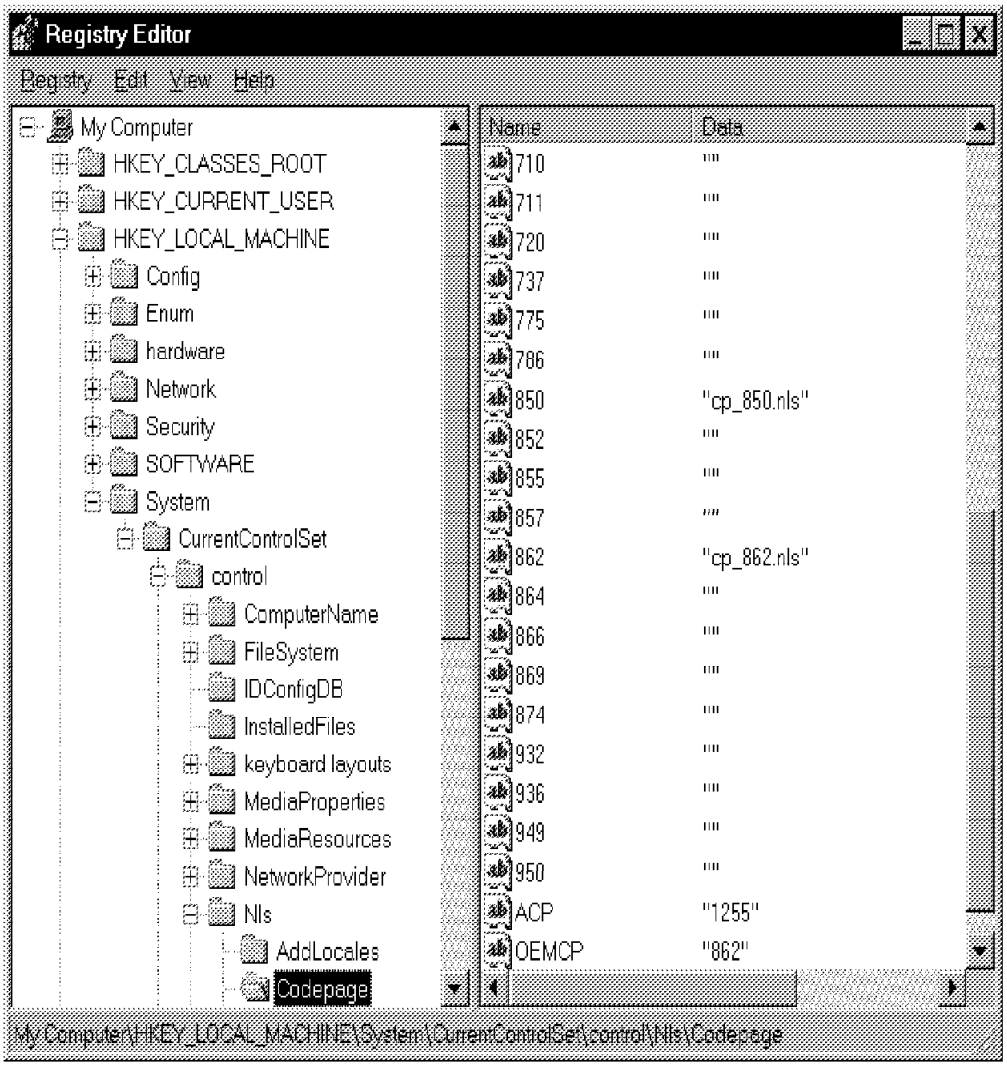


Figure 11. Windows 95 Registry

Important

Do not try to change the Windows 95 registry entries. This might cause problems in Windows 95 operation and even prevent it from booting. The previous information was provided just as an aid to understand the conversion process. If you need to change the OEM code page, refer to Chapter 34 (International Windows 95) in the *Microsoft Windows 95 Resource Kit* by Microsoft Press, ISBN 1-55615-678-2.

The OEM code page is used to determine the compatibility of the client and the host server CCSID.

7.4.4.2 OS/2 Code Page

OS/2 uses only one type of code page that is defined in the ″CONFIG.SYS″ file. This value is retrieved by calling the DosGetCp API. An easy way to find the OS/2 code page is to use the CHCP command at the OS/2 command prompt. As shown in Figure 12, in OS/2, you can prepare more than one code page. This feature allows you to change the active code page for a specific OS/2 session to meet your requirements for client conversion.

```
C:\>CHCP
Active code page:  437
Prepared system code pages:  437, 850
```

Figure 12. Sample Output from CHCP Command

7.4.5 Windows 3.1 ODBC

The host server CCSID is set as described in Section 7.4.1, “Host Server CCSID” on page 54.

The tables between the server job CCSID and the personal computer OEM code page are downloaded from the host as part of the client to host server conversation. The tables between the personal computer OEM code page and the personal computer ANSI code page are hard coded in the ODBC driver. The two pairs of tables are combined to create a single pair of tables between the host server code page and the personal computer ANSI code page. Those tables are being used for the subsequent conversions.

For bidirectional languages, the ODBC solution is not complete since the visual to logical conversion described in Section 10.6, “Bidirectional Languages” on page 113 is not handled by the ODBC driver. The ODBC standard defines a way to overcome this problem by specifying a translation DLL.

A translation DLL is a windows program that is called by the ODBC driver to perform any translation or conversion that is implemented in it. To enable the translation DLL, you need to specify it in the ODBC.INI file.

For example, for Hebrew visual to logical conversions, there is a translation DLL and documentation on how to set it up in the IBM Web site:

`http://www.ibm.co.il/html/products/odbc.htm`

7.4.6 File Transfer in Windows 95/NT and Optimized OS/2 Clients

The Optimized Clients′ File Transfer handles the translation in a different way. There are only two factors that influence the transfer process:

- The CCSID of the AS/400 file or file column
- The PC code page

The PC code page is set as described in Section 7.4.4, “How PC Code Page is Determined” on page 59. The EBCDIC ⇔ ASCII conversion is always performed at the receiving side:

- For file upload to the AS/400 system, the conversion is done on the AS/400 system.
- For file download to the client, the conversion tables are downloaded to the client as described in Section 7.4.3, “Downloading the Conversion Tables” on page 59 and the conversion takes place at the client side.

In this case, the conversion tables are always between the AS/400 file or file column CCSID and the PC code page, and they are not sensitive to the user or job CCSID.

If the PC file is in an ANSI code page, a second conversion is performed between the PC ASCII and ANSI code pages. This conversion is always executed in the client side by using a client platform API.

Note!

The algorithm concerning the conversion between EBCDIC and ANSI code pages is likely to be changed in the future.

7.4.7 ODBC in Windows 95/NT and Optimized OS/2 Clients

The AS/400 host server job CCSID is set according to the normal rules for setting a job CCSID with the exception that *HEX CCSID is not a valid option. The user profile being used for AS/400 host server jobs is QUSER so the CCSID and LANGID are determined by the QUSER settings.

When a client application connects to its host server job, the client application sends the user ID of the user that logged on to the AS/400 system and the PC code page to the host server job. If the user profile CCSID is different than *HEX, the corresponding host server job CCSID is set to the user profile CCSID.

The PC code page is set as described in Section 7.4.4, “How PC Code Page is Determined” on page 59.

For bidirectional considerations, refer to Section 7.4.5, “Windows 3.1 ODBC” on page 61. For Windows 95, however, specifying the use of the translation DLL is done in the registry.

7.4.8 Network Drives

Client Access allows you to use Network Drives to map to the Integrated File System (IFS) on an AS/400 system. This function is analogous to Shared Folders on the original clients, but Network Drives gives you access to the entire IFS, not just QDLS.

You can use Network Drives in two modes on a file basis:

- Binary mode
- Text mode

To set text mode, you need to specify the file extensions for automatic EBCDIC/ASCII conversion on the Network Drives property sheet in the Client Access properties program (see Figure 13).

When using Network Drives in text mode, the client supplies the requested client code page value. The host file server converts the data between the client code page and the EBCDIC CCSID.

The host server CCSID and the translation tables used are set as described in Section 7.4.1, “Host Server CCSID” on page 54.

The PC code page is set as described in Section 7.4.4, “How PC Code Page is Determined” on page 59 using the ICONV API described in the National Language Support APIs in *System API Reference*, SC41-5801,

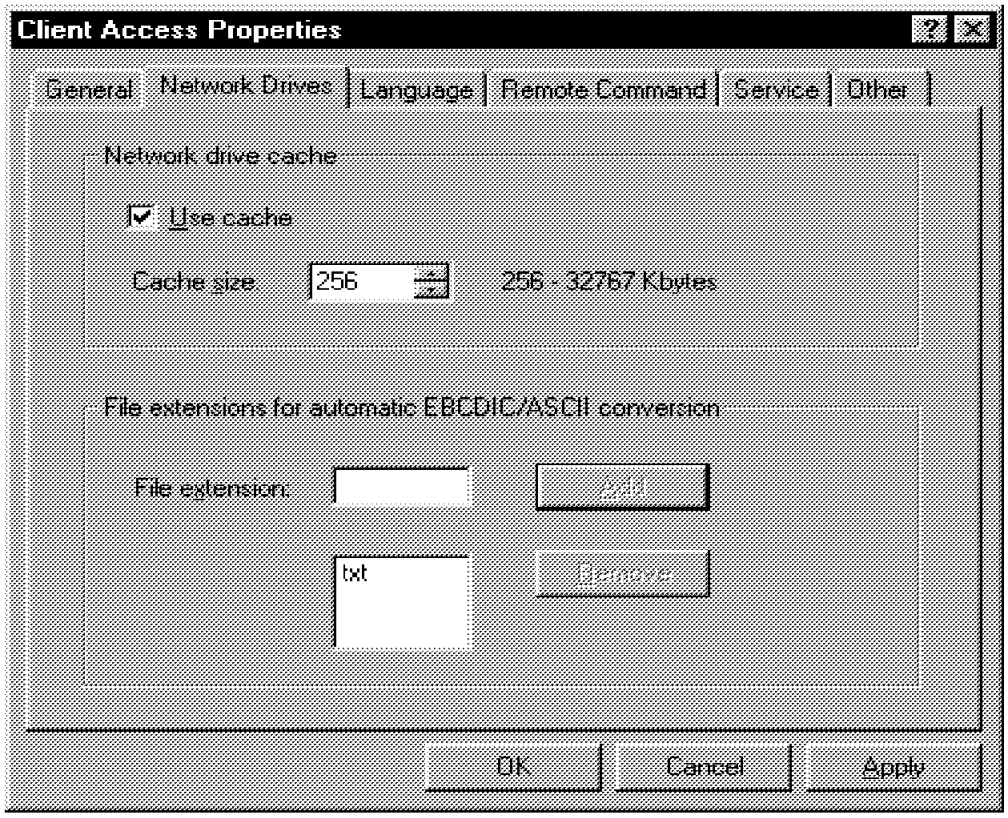


Figure 13. The Client Access Properties - Network Drives Property Sheet

7.4.9 Data Queue APIs

The host server CCSID and the translation tables used are set as described in Section 7.4.1, “Host Server CCSID” on page 54.

The PC code page is set as described in Section 7.4.4, “How PC Code Page is Determined” on page 59. The API defaults to use the PC OEM code page but it can be overridden.

7.4.10 Network Print

Client Access allows you to print to AS/400 attached printers through network printers (on original clients, network printers were called virtual printers). The network print function gets the data from the Windows 95/NT in UCS-2 and converts it to EBCDIC in two steps:

1. Convert from UCS-2 to Windows 95/NT OEM code page using the ″WideCharToMultiByte″ Win32 API.
2. Convert from OEM to EBCDIC using ″cwbNL_Convert″ Client Access API.

The PC OEM code page is set as described in Section 7.4.4, “How PC Code Page is Determined” on page 59.

The EBCDIC code page and the translation tables used are set as described in Section 7.4.1, “Host Server CCSID” on page 54.

If the UCS-2 print data stream contains characters that are not part of the OEM code page, data integrity problems might occur.

For example, if you use Client Access on Hebrew Windows 95 to create a network printer and share this printer using Windows 95 peer services, printing to the shared printer from Danish Windows 95 with Danish characters does not produce the correct results.

The problem is that the printer data stream from the Danish Windows 95 represented in UCS-2 and can contain UCS-2 codes of Danish characters, it is converted on the Hebrew Windows 95 system to Hebrew OEM code page that does not contain any code points for Danish characters.

Because of this problem, if you are using Client Access network print function, make sure that the sending Windows 95 system and the Windows 95 system to which the printer is attached, share the same OEM code page.

7.4.11 Remote Commands and Distributed Program Call APIs

The host server CCSID and the translation tables used are set as described in Section 7.4.1, “Host Server CCSID” on page 54.

The PC code page is set as described in Section 7.4.4, “How PC Code Page is Determined” on page 59. The APIs default to use the PC OEM code page but it can be overridden.

7.4.12 Special Instructions for Setting Up PC5250

To get the proper language support for PC5250, you need to configure the code page, the keyboard, and the fonts that are used. Refer to *Client Access for Windows 95/NT - Setup*, SC41-3512 for more information.

7.4.12.1 Host Code Page

Host code page specifies the table that is to be used to map EBCDIC codes from the AS/400 system to the appropriate ANSI graphics on the PC.

7.4.12.2 Keyboard Setup

The keyboard setup maps specific keys on the PC to the EBCDIC code points that are sent to the AS/400 system. It is extremely important to choose the correct keyboard because it is not set automatically and it does influence the actual data entered into the file.

7.4.12.3 Special Fonts for PC5250

To run PC5250 in the languages listed in Table 18, you need special font files.

.

Table 18. Languages that Need Special Font Files

Language	**Font File**
Latin2 (Central Europe)	PCS1250.FON
Cyrillic Multilingual	PCS1251.FON
Greek	PCS1253.FON
Turkish	PCS1254.FON
Arabic	PCSANSIA.FON
Hebrew	PCSANSIH.FON
Thai	PCSTHAI.FON

You can find detailed instructions in the online *Client Access User's Guide*.

7.5 Other Client/Server Related Conversions

When copying files between OS/400 EBCDIC file system and other ASCII file systems in the IFS, code page conversions take place.

7.5.1 Copy Personal Computer Document

The Copy from PC Document (CPYFRMPCD) command use the ASCII code page and EBCDIC code page from the INZPCS command. As a result, unpredictable translations may occur if the ASCII code page used to create the document does not match the ASCII code page from the INZPCS command.

For the reverse process, the Copy to PC Document (CPYTOPCD) command converts EBCDIC character data to ASCII code points corresponding to the INZPCS ASCII code page. For the CCSID-tagged data where the CCSID does not match the INZPCS EBCDIC code page, data integrity might be lost.

Both commands have a parameter that enables you to specify a translation table.

The ASCII and EBCDIC code pages are retrieved from the QUSRSYS/QINZPCSDA data area (set by the INZPCS command).

7.5.2 Copy Stream Files Commands

There are two copy stream file commands:

- Copy from Stream File (CPYFRMSTMF)
- Copy to Stream File (CPYTOSTMF)

If the data conversion option is *AUTO, these two commands use CCSID information during the copy process. Copy from Stream File (CPYFRMSTMF) does not create database files if they do not exist. For existing files, there is a conversion from the stream file code page to the CCSID of the database file. Copy to Stream File (CPYTOSTMF) can create streams files if they do not exist. If the stream file does not exist, it is created with a code page that matches the CCSID of the database file. If the stream file already exists, the CCSID conversion is between the CCSID of the database file and the code page of the stream file.

Chapter 8. NLS Considerations in TCP/IP Environment

This chapter provides a brief review of National Language Support considerations in such TCP/IP applications as:

- FTP (File Transfer Protocol)
- TELNET
- SMTP (Simply Mail Transfer Protocol)
- POP (Post Office Protocol)
- WSG (Workstation Gateway)
- HTTP (Hypertext Transfer Protocol)
- SLIP (Serial Line Interface Protocol)

8.1 FTP

The FTP client function allows you to send and receive copies of files to or from remote systems across a TCP/IP network. In addition, FTP client subcommands are provided for renaming, appending to, and deleting files.

Subcommands TYPE and LTYPE (local type) are important when using FTP between the systems with the different language environments.

- To specify the file-transfer type or the representation in which the transfer is to take place on local and remote system, use the TYPE subcommand in the following format:
  - TYPE A (specifies the transfer type as the ASCII). It has the same meaning as ASCII the subcommand.
  - TYPE C CCSID# (specifies the transfer type to any CCSID that is installed on the system). The CCSID number must follow C (for example, TYPE C 278).
  - TYPE E (specifies the transfer type as EBCDIC). It has the same meaning as the EBCDIC subcommand.
  - TYPE I (specifies the transfer type as image). It has the same meaning as the BINARY subcommand.
- To specify the file transfer type for only the local system, use the LTYPE subcommand in the following format:
  - LTYPE C CCSID# This is the same format as used for TYPE C CCSID# previously shown.

If no parameters are specified for TYPE or LTYPE, then the present setting for the local system type is displayed.

For more information on the LTYPE and TYPE subcommands, see the *TCP/IP Configuration and Reference*, SC41-5420.

8.1.1 TYPE E (EBCDIC) Transfers between Two EBCDIC Systems

This type of transfer might be negotiated, for example, between two AS/400 systems. When data is transferred using TYPE E (or EBCDIC), the data is stored as is and, therefore, is in the EBCDIC code page of the file that it came from. This can result in the stored file being tagged with an inappropriate CCSID value when the primary language of the two AS/400 systems is different.

For example, when data in code page 278 is sent using TYPE E to the QSYS.LIB file system on a machine where the file does not exist, the data is stored as is in a new file tagged with CCSID 65535. If the receiving file already exists, the data is received as is and tagged with the existing file CCSID which may not be 278.

To avoid incorrect CCSID tagging, you can use the TYPE C CCSID# subcommand (for example, TYPE C 278) to specify the CCSID of the data being transferred. When a CCSID is specified on a transfer and the data is written to an existing file, the data is converted to the CCSID of the existing file. If no target file exists before the transfer, a file is created and tagged with the specified CCSID. For an example, a file with a CCSID of 278 would be created on the receiving the system. When the target file already exists, then the data is converted from CCSID 278 to the CCSID of the target file.

8.1.2 TYPE A (ASCII) Transfers between Two EBCDIC Systems

When using FTP in an ASCII mode between two EBCDIC systems, the data on the system sending the file is converted from its stored EBCDIC code page to ASCII. The ASCII is sent to the other system where it is converted to the EBCDIC code page of the receiving system.

When starting FTP, the ASCII CCSID is set to the default 819. Usually this does not present a problem because the 7-bit ASCII code page used by the two systems is the same unless the EBCDIC characters on the sending system are not defined in the ASCII code page. Also, some characters in the ASCII code page may be mapped differently between the two different EBCDIC code pages. This might occur if some of the ASCII characters are variant (the character occupies a different hexadecimal code point in an EBCDIC code page). The variant character may be interpreted differently on the receiving system if the EBCDIC code page is different from that of the system sending the file.

8.1.3 TYPE A (ASCII) Transfers between EBCDIC and ASCII Systems

The ASCII CCSID used for ASCII type transfers (including the sending of FTP server subcommands to the remote system) is specified when starting the FTP client. The default ASCII CCSID for the STRTCPFTP command is 819.

FTP cannot be started when character conversion between the AS/400 system′s default job (EBCDIC) CCSID and the ASCII CCSID specified for FTP is not available. In this case, message TCP3C14 (″Unable to convert data from CCSID &1 to CCSID &2) is displayed. You can change the ASCII CCSID by specifying a value for the coded character set identifier parameter of the FTP command. CCSID 850, which contains the IBM Personal Computer Latin-1 coded character set, is an ASCII CCSID for which character conversions are available to all valid job CCSID values.

8.1.4 TYPE I (Binary) Transfers for Save Files

*SAVF files must be sent as images and, therefore, require the FTP BINARY subcommand to be run before the GET or PUT subcommands. When transferring a *SAVF, we recommend that you create files in advance on a target system for reasons of performance and integrity.

Transferring a save file because it is a file format peculiar to the AS/400 system can only be made usable if the sending and receiving systems are both AS/400 systems. However, a save file can be sent to a non-AS/400 system and stored there for backup purposes. The save file can be transferred later to the AS/400 system with FTP. No conversion is performed when using the BINARY transfer type.

8.1.5 CCSID Tagging for New Files when Using FTP

When FTP creates a new file on an AS/400 system, it is tagged with a CCSID or the code page of that CCSID to identify the character data in that file. For ″root″, QOpenSys, and QLANSrv, the file is always newly created except when an append is done. When appending data to an existing file, the tag of the file is not changed.

Table 19 summarizes how FTP assigns these values for different file systems and transfer types.

Table 19 (Page 1 of 2). CCSID Code Page Tagging for AS/400 Files

Receiving File System	**FTP TRANSFER TYPE A (ASCII)**	**FTP TRANSFER TYPE C (′CCSID′)**	**FTP TRANSFER TYPE E (EBCDIC)**	**FTP TRANSFER TYPE I (Image/Binary)**
QSYS.LIB	Related default EBCDIC CCSID of FTP default ASCII CCSID. If conversion table is specified, then 65535.	′CCSID′ if EBCDIC CCSID. If the ′CCSID′ is ASCII, then related default EBCDIC CCSID.	65535	65535
QDLS, QOPT, and other HFS file systems	Not supported	Not supported	Not supported	Not assigned by FTP
″root″, QOpenSys	Code page of default ASCII CCSID	′CCSID′ value specified in TYPE C CCSID# subcommand	Code page of Job CCSID if it is not 65535. If Job CCSID is 65535, assign code page of Default Job CCSID.	Code page of default ASCII CCSID

Table 19 (Page 2 of 2). CCSID Code Page Tagging for AS/400 Files

Receiving File System	FTP TRANSFER TYPE A (ASCII)	FTP TRANSFER TYPE C ('CCSID')	FTP TRANSFER TYPE E (EBCDIC)	FTP TRANSFER TYPE I (Image/Binary)
QLANSrv	ASCII code page of the network server description for file directory	ASCII code page of the network server description for file directory	Not supported	ASCII code page of the network server description for file directory

Note!

The default ASCII CCSID is defined when the FTP job is started:

- For the client, it is the CCSID parameter of the FTP command.
- For the server, it is the CCSID parameter of the FTP configuration attributes, which can be changed using the CHGFTA command.

QFileSvr.400 file assignments depend on the file system receiving the file.

For EBCDIC and related ASCII CCSIDs. refer to Table 29 on page 123.

8.2 TELNET

The TELNET protocol allows you to access and use resources of a remote system as if your workstation is locally connected to the remote system.

The following modes are supported on AS/400 system:

- 5250 full-screen mode
- 3270 full-screen mode
- VTxxx full-screen mode

  The term VTxxx throughout this chapter refers to VT220, VT100, and VT52 full-screen modes.

ASCII line mode is not negotiated for the AS/400 system. If any 5250, 3270, or VTxxx mode cannot be negotiated by TELNET client, it uses VT100 full-screen mode as the default.

8.2.1 5250 and 3270 Full-Screen Mode Considerations

- 5250 full-screen support is put into effect by negotiating 525x workstation support with the remote TELNET server application. 5250 full-screen support can only be negotiated with a TELNET server application running on an AS/400 system or a system that supports the TELNET 5250 server.
- 3270 full-screen support is negotiated with any TELNET server application that supports 3270 full-screen (System/370 or System/390).
- The keyboard language type you specify for your workstation using the keyboard language type parameter on the TELNET command must be the same as the keyboard language type parameter of the remotely attached

workstation. If you specify a keyboard language type that does not match, some of the characters are not displayed as expected.

- If you use a personal computer to establish 5250 or 3270 full-screen mode and your personal computer uses the Client Access Workstation Function (WSF), you can display the layout of your 5250 keyboard using the Workstation Function Keys (WSFKEYS) command. You can alter the style using the Configure Workstation Function (CFGWSF) command. These commands are discussed in the *Client Access/400 for DOS with Extended Memory Setup* manual. If your personal computer does not use the workstation function, refer to the appropriate documentation for your emulator (for example, CM/2) to view or change the keyboard style.

8.2.2 VTxxx Full-Screen Mode Considerations

There are three alternative methods of selecting character mapping between the client and server systems with VTxxx emulation. These are:

- Coded character set identifier (CCSID)
- Multinational mode
- National mode

If none of these modes is suitable, you may set up and specify your own user-defined mapping tables.

- Mode selection is done with the CCSID parameter of the TELNET command. The incoming ASCII-to-EBCDIC table (TBLVTIN) and outgoing EBCDIC-to-ASCII table (TBLVTOUT) parameters of this command allow the specification of user-defined mapping tables. If these are not required, the default value of *CCSID allows for character mapping by using the mode specified in the CCSID parameter.
- The multinational mode supports the DEC multinational character set, which is an 8-bit character set that contains most characters used in the major European languages. The ASCII character set is included in the DEC multinational character set. The DEC multinational character set is used by default.
- The national mode supports the national replacement character set, which is a group of 7-bit character sets. Only one national character set is available for use at any one time. VT220 also supports the standard 7-bit ASCII character set as part of the national mode. The VT220 terminal supports the following 7-bit ASCII national language character sets:
  - British
  - Danish
  - Dutch
  - Finnish
  - French
  - French/Canadian
  - German
  - Italian
  - Norwegian
  - Spanish
  - Swedish
  - Swiss
  - US English

To use a national mode, mapping tables are required to map incoming ASCII data into EBCDIC and outgoing EBCDIC data into ASCII when operating in VTxxx full-screen mode.

A national mode (NLS mapping table) may be selected with the CCSID parameter on the TELNET command (see the *VTxxx--Start TCP/IP TELNET Command* topic in the *TCP/IP Configuration and Reference* manual).

A numeric value representing a registered CCSID value in the range of 1 through 65533 may be entered to identify the appropriate mapping table. Details of registered CCSIDs are found in the *International Application Development* manual.

The NLS mapping tables are built dynamically to a remote system the first time TELNET is used and are based on DEC national replacement character sets. The character sets are based on 7 bits; they can contain only the unique characters from one country. The DEC multinational character set is based on 8 bits, and it has sufficient bits to allow the unique characters from a group of countries to be included.

8.3 SMTP and POP

The SMTP function allows you to send or receive electronic mail. SMTP supports distribution notes, messages, and final-form text documents.

POP enables electronic mail between registered POP clients. This includes clients running on Windows, OS/2, AIX, and MacIntosh.

8.3.1 SMTP

As AS/400 business computing systems use the EBCDIC form of hexadecimal data representation and most UNIX or non-IBM systems use ASCII for data representation, an EBCDIC-to-ASCII conversion is required when mail is sent from the AS/400 system.

You can specify the conversions in two ways:

- Configuring an ASCII single byte CCSID. The default CCSID for SMTP is 00819. This is the CCSID number for the ISO standard 8859-1 Latin-1 8-bit ASCII. If a 7-bit U.S. English ASCII is required, use CCSID 00367. Any ASCII CCSID supported by the system may be configured.

  The following entry options are available:

  – 1-65533- an ASCII CCSID number
  – *SAME- no change
  – *DFT- set the CCSID number to 00819.

- Specifying a mapping table. We recommend that you specify a mapping table that maps code page 500 to whatever code page the destination system uses because OfficeVision sends and receives mail using the EBCDIC code page 500 regardless of the national language installed. For example, if an AS/400 system wants to send mail to an RS/6000 system that uses ASCII code page 850, specify the outgoing and incoming mapping tables for SMTP. When mail is sent from an AS/400 system, SMTP converts the mail from code page 500 to code page 850 and vice versa when mail is received by the AS/400 system.

8.3.2 POP

Mail that is sent from the Internet through SMTP is placed in the client's mailbox in its original form or character set (ASCII is assumed). Mail sent from an application that uses EBCDIC such as OfficeVision or SNA is converted to a Multipurpose Internet Mail Extensions (MIME) standard ASCII character set, depending on the configuration. MIME standard allows users to interchange video, image, audio, or binary files as well as text messages.

The supported standard POP character sets are:

- Arabic
- Cyrillic
- Greek
- Hebrew
- Japanese
- Latin 1
- Latin 2
- Latin 5
- Seven-bit ASCII

Figure 14 on page 74 describes how the POP server handles the incoming mail ASCII-to-EBCDIC conversion.

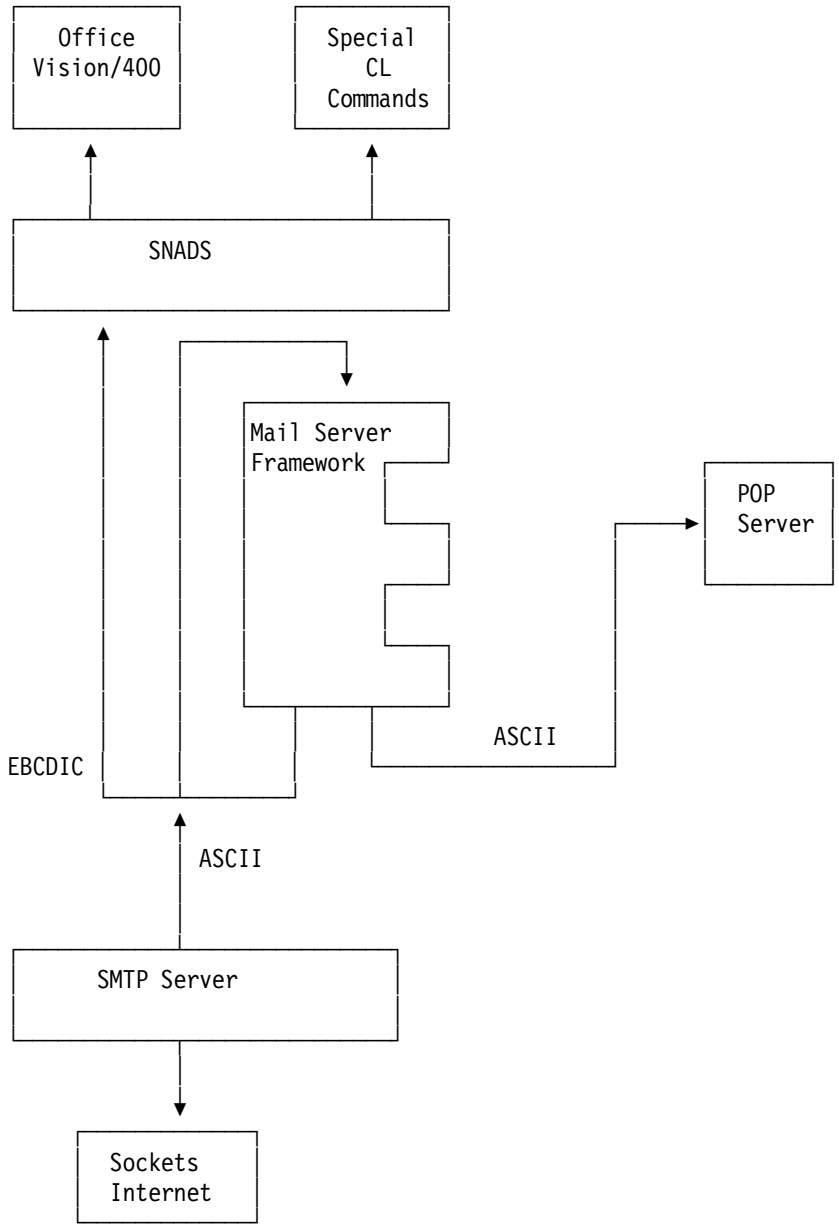


Figure 14. ASCII-to-EBCDIC Mail Handling by POP Server

The path from the Internet to POP only passes the mail through. The path to SNADS does ASCII-to-EBCDIC conversion. Simple SMTP is converted using the SMTP configuration information. MIME mail is converted using the POP configuration information. All TCP/IP mail, both simple mail (SMTP) and MIME-formatted mail, sent to OfficeVision for OS/400 must be converted from ASCII to EBCDIC. For this conversion to work in any country both SMTP (CHGSMTPA command) and the POP (CHGPOPA command) server must be configured.

Figure 15 on page 75 describes how the POP server handles outgoing mail EBCDIC-to-ASCII conversion.

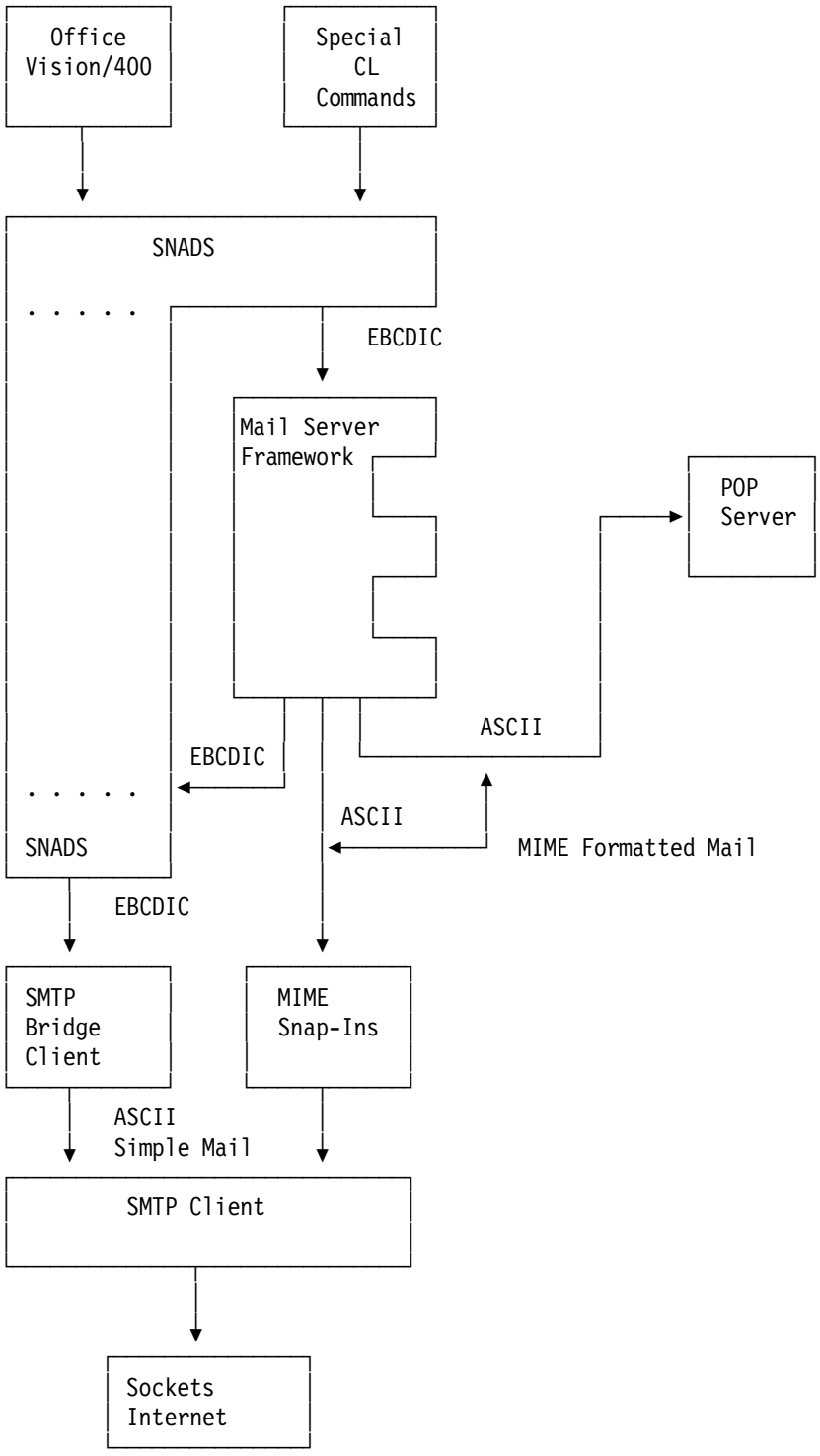


Figure 15. EBCDIC-to-ASCII Mail Handling by POP Server

There are two paths by which text is converted from OfficeVision format (EBCDIC) to Internet format (ASCII):

- The SMTP bridge
- The Mail Server Framework

Mail is converted in the SMTP bridge using the SMTP configuration (CHGSMTPA command). Mail is converted in the Mail Server Framework using POP configuration information (CHGPOPA command).

The MIME coded character set identifier configuration parameter (MIMECCSID) on the Change POP Attributes (CHGPOPA) command determines which CCSID to use for translation and under what conditions to use it. Once you specify a CCSID, you can choose to use it all the time (*ALWAYS), or only if a better fit cannot be found (*BESTFIT).

Note: For the mail between OV/400 and non Latin-1 countries, the OV nonconvert option must be set or conversion results will be unpredictable.

The CCSIDs supported for conversion are MIME or Internet-defined standard character sets:

Table 20. Supported MIME Standard Code Pages

MIME Standard	Name	ASCII CCSID	Character Set	EBCDIC CCSID
US-ASCII	US English	00367	103	00500
ISO-8859-1	Latin-1	00819	697	00500
ISO-8859-2	Latin-2	00912	959	00870
ISO-8859-5	Cyrilic	00915	1150	01025
ISO-8859-7	Greek	00813	925	00875
ISO-8859-8	Hebrew	00916	941	00424
ISO-8859-9	Latin-5	00920	1152	01026
ISO-2022-JP	Japan MBCS	05052	1064,1062,1121,1120	05026

8.4 WSG

The WSG transforms AS/400 5250 data streams to Hypertext Markup Language (HTML) for dynamic display on Web browsers. This allows you to run AS/400 applications from any workstation that has a Web browser.

The WSG server can support browsers that operate using national languages other than English. Browsers that negotiate requests specifying one of the following MIME standards are supported on a document by document basis.

Note!

The WSG server supports the same MIME standards and CCSIDs as the POP server. Refer to Table 20.

The WSG server can also support countries not covered by these specifications. By configuration of the CHGWSGA CCSID parameter, the server administrator specifies the default ASCII-to-EBCDIC conversions for all browsers′ requests for that server. The CHGWSGA CCSID parameter is intended to represent the ASCII CCSID of the browser side and the EBCDIC CCSID used for the AS/400 side is calculated (using system APIs) from the ASCII CCSID. This CCSID is the default. This default can be overridden with an NLV specific session request, however.

If the WSG session is initiated with an NLV request, the request itself indicates the desired EBCDIC CCSID. WSG calculates the ASCII CCSID to use that best fits the requested EBCDIC CCSID. In all cases, WSG attempts to use the ″best fit″ CCSID value based upon the NLVs installed on the AS/400 system. Please check the following Web site for additional information.

http://www.as400.ibm.com/products/wsg/wsgnews.htm

8.5 HTTP

The HTTP server is a TCP/IP application that allows your AS/400 system to serve multimedia objects such as Hypertext Markup Language (HTML) documents to World Wide Web browser clients.

Browser clients use Uniform Resource Locators (URLs) to make requests to the HTTP server. URLs are identifiers and instructions (such as programs) that tell the HTTP server what object to process and how to process it.

In most cases, it simply identifies the name of the object the client wants the HTTP server to serve.

The HTTP server sends data from all file systems except QSYS as binary objects without conversion. Therefore, documents that are created for a given code page in ASCII and stored on the AS/400 system are sent without conversion. For QSYS, the HTTP server can provide documents to browsers that operate using national languages other than English. Browsers that negotiate requests specifying one of the following MIME standards are supported on a document-by-document basis.

Note!

For supported MIME standards and CCSIDs by the HTTP server, refer to Table 20 on page 76.

The server can also support countries not covered by these specifications. By configuration of the assumed ASCII CCSID parameter, the server administrator can force a given set of ASCII-to-EBCDIC conversions for all browser requests for that server. Please refer to the following Web site for AS/400 Web builder′s information.

AS/400 Web Builders Workshop:

http://www.as400.ibm.com/workshop/webbuild.htm

8.6 SLIP

SLIP allows you to connect systems over a pair of modems that are connected through a telephone line. You can use SLIP to connect personal computers to your AS/400 system from the modem jack in a hotel or your house. You can also simply connect two computers together directly without any modems using an unswitched or NULL modem connection.

Once a SLIP connection is established, the application such as TELNET or FTP that is running over the connection must manage the ASCII-to-EBCDIC and EBCDIC-to-ASCII translation.

- If you are not using a connection script to establish a SLIP connection to or from a remote system, no other NLS considerations are necessary.
- If you are using a connection script, it may be important to understand how the AS/400 system translates connection scripts from EBCDIC to ASCII to be transmitted back and forth over the connection.

Connection scripts on the AS/400 system are stored in EBCDIC format. However, most systems that connect to the AS/400 system are probably ASCII systems that send their connection script data in ASCII format. Therefore, the AS/400 system must translate the ASCII data that is coming in to the AS/400 system over the connection from ASCII to EBCDIC. The AS/400 system can then compare the connection script to the EBCDIC connection script that is stored on the AS/400 system.

Before the AS/400 system can send data to a remote system over the SLIP connection, the AS/400 system must translate the connection script data from EBCDIC to ASCII. The default EBCDIC and ASCII CCSID values that the AS/400 system uses to translate connection scripts are shipped with the AS/400 system and located in members in file QUSRSYS/QATOCPPSCR.

This default connection script file, QATOCPPSCR, uses a CCSID of 500, which means character set 697 and code page 500. Therefore, all character data in each connection script member has a hex code point that this CCSID supports. The AS/400 system uses the EBCDIC CCSID value to translate connection script data when the AS/400 system sends the connection script data to or receives the connection script data from a remote system.

The default values cover most needs, but you can change the default values if you need support for other characters. See the *Connection Dialog Scripts* topic in the *TCP/IP Configuration and Reference*, SC41-5420 for details on using connection scripts and for the member names.

Chapter 9. NLS Considerations for Printing Environment

After you consider installing a different (new) primary language version on your AS/400 system, you also should be aware of the possible impact to your printing environment. When you do the installation, you must verify your printing environment that refer to the parameter values within your printer files, printer device descriptions, and if necessary within your user profiles. Otherwise, you may not receive the printout you expected when you print system or application-oriented spooled files.

After you install the new primary language, the NLV system cultural value setting is changed to the new values. The new QCHRID value may impact your printout whenever it points to a CHRID value used either in a printer file or a printer device description that uses the new QCHRID value instead of the old one. Changing to a new QCCSID may also impact user profiles using the *SYSVAL parameter for the CCSID and using printer files specifying *JOBCCSID as the CHRID value.

9.1 General Recommendations

- Do extensive testing of your printouts **before** changing any printer files or parameters.

  In many cases, if your printer files and device descriptions use the default values, you must not change anything as the printout is controlled by the panel setting of the device, or the printer file parameters override default settings within your printer device descriptions.

- Check all of your printer files system-wide using *SYSVAL for the CHRID parameter.

  These printer files obtain the new QCHRID value and may cause incorrect print output. You may use the information from 4.6, “Evaluate Database Information” on page 29, to modify the program to receive a list for all system-wide printer files. If you do not want to use the new QCHRID value, either change from *SYSVAL to *DEVD (default) or to a preferred CHRID value.

- Check all of your printer device descriptions:

  Note: If you have DEVDs containing a device type *IPDS, AFP *YES, and AFPATTACH *APPC and DEVDs containing Host Print Transform *YES (for ASCII printers), you may do the following changes:

  If CHRID on DEVD is *SYSVAL, change this to the preferred CHRID.

  You also may change printer files with CHRID *DEVD to use the preferred CHRID.

- Check printer files that have CHRID *JOBCCSID:

  When printing with this printer file the spooled file is tagged with the CHRID taken from the Job CCSID. If this is no longer needed (the reason may be a new QCCSID for the new NLV) change parm *JOBCCSID to the preferred CHRID or to default *DEVD.

9.2 Printing in a Multilingual Environment

When you consider printing in a multilingual environment, you should understand the concept of printer file support on the AS/400 system and understand the printer file support for CCSIDs. You need to know which parameters and elements on the printer file impact the layout of the printed characters. It is important to define the right character identifier (CHRID) or the right character set and code page corresponding to the selected language group and corresponding to the type of the target printer.

9.3 Printer Files

What is a printer file?

Printer files describe how the system is to operate on data as it passes between your application program and a printer. The printer file to be used is normally specified in your application program.

The types of printer files are:

- **Program-described printer files:**

  Program-described files rely on the high-level language program to define records and fields to be printed.

- **Externally described printer files:**

  Externally described printer files use DDS rather than the high-level language to define records and fields to be printed. If you use DDS, specify the name of your DDS source file in the SRCFILE parameter of the Create Printer File (CRTPRTF) command. DDS gives the application programmer much more control over how the printed output is formatted and printed.

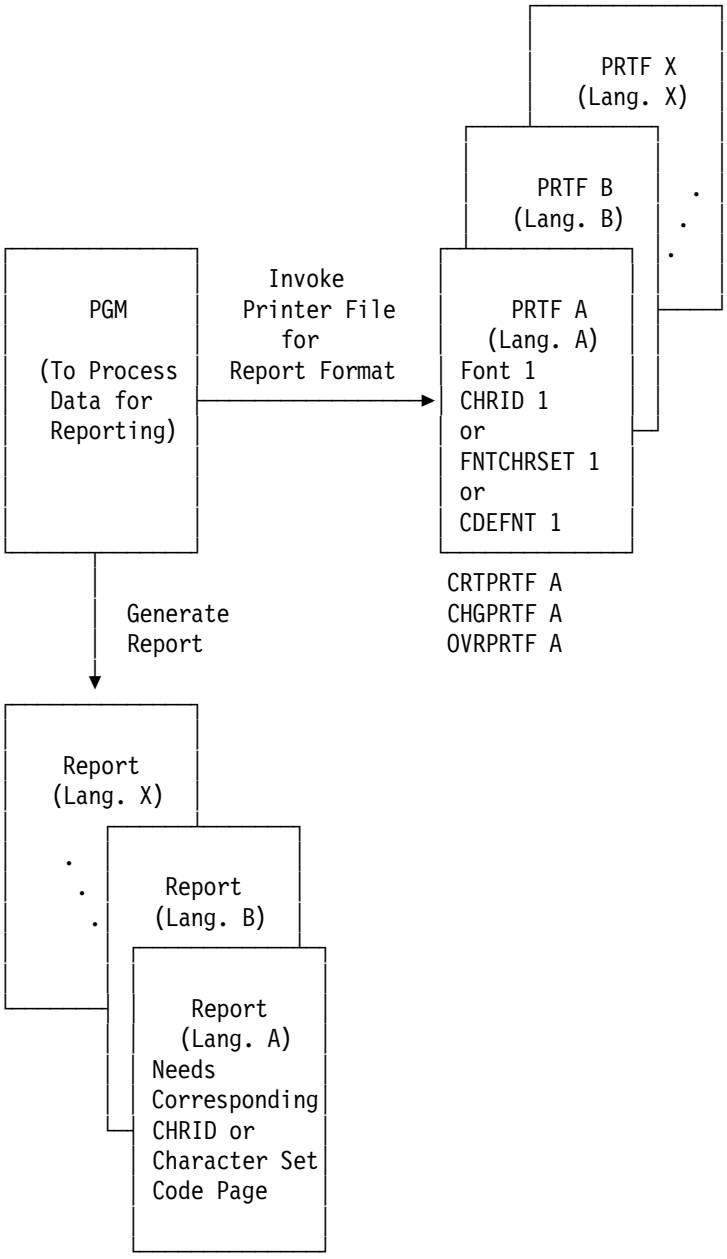


Figure 16. How Externally Described Printer Files Are Used

9.4 Printer File Parameters

The following list contains parameters from the printer file. These are parameters that the application program accesses or looks at when it opens the printer file. They are a subset of the parameters in the printer file, but not all of them. When the application looks at each parameter, it finds a value specified for each parameter.

FILE Specifies the qualified name by which the printer file being called is known.

DEV Specifies the name of a printer device description. The device description is an object that contains information describing a particular device (printer in this case) or logical unit that is attached to the system. For non-spooled output, this identifies the printer device used with the printer file to produce the printed output. For spooled output, if OUTQ(*DEV) is also specified, the default output queue for the specified printer is used for the spooled file.

DEVTYPE Specifies the type of data stream created for a printer file. This parameter indicates whether the resulting data stream should be:

```
*   Intelligent printer data stream (*IPDS)
*   SNA character stream (*SCS),
*   ASCII data stream (*USERASCII)
*   Advanced Function Printing data stream (*AFPDS)
*   Line data (*LINE)
*   Mixed data (*AFPDSLINE)
```

PAGESIZE

Specifies the length and width of the printer forms used by this printer file.

LPI Specifies the line spacing setting on the printer in lines per inch used by this printer file.

UOM Specifies the unit of measure to be used for various measurements used in printing. The possible values are:

***INCH** Use inches as the unit of measure.

***CM** Use centimeters as the unit of measure.

CPI Specifies the printer character density in characters per inch (CPI) used by this printer file.

OVRFLW Specifies the line number where an overflow message is issued.

RPLUNPRT

Specifies two sub-parameters:

1. Whether unprintable characters are replaced
2. Which substitution character (if any) is used.

An unprintable character is a character the printer is unable to print.

FIDELITY Specifies whether the spooled file should continue printing if errors are encountered.

***CONTENT** If errors are encountered, the spooled file continues to print.

***ABSOLUTE** If errors are encountered, the spooled file does not print.

FONT Specifies, for the 3130, 3812, 3816, 3820, 3825, 3827, 3829, 3831, 3835, 3900, 3912, 3916, 3930, 3935, 4028, 4224, 4230, 4234, 4247, 4312, 4317, 4324, and 5219 printers (including ASCII printers emulating the 3812 or 5219 printer), the font identifier (FGID), point size (if the font supports multiple point sizes) of the font used with this printer.

CHRID Specifies the character identifier (graphic character set and code page) for fields identified with the CHRID DDS keyword for externally described printer files. Otherwise it is the CHRID that is used for all printed output for that file.

*DEVD
*SYSVAL
*JOBCCSID

FNTCHRSET
Specifies the font character set and the code page to be used at printing time.

The font character set provides the size and style of the character.

The code page identifies which characters in a font are available for printing. These parameters only can be used with PRTF device type *AFPDS and the target printer should be configured AFP *YES (IPDS Laser printer).

CDEFNT Specifies the coded font to be used at printing time.

A coded font is an AFP resource that is made up of a font character set **and** a code page. This parameter only can be used with PRTF device type *AFPDS and the target printer should be configured AFP *YES (IPDS printer).

There are many more parameters available in a printer file.

9.5 Printer File Support for CCSIDs

When a printer file object is created, it is tagged with the coded character set identifier (CCSID) of the source file. At compile time, all character data is read from the primary source file without any character conversion being performed.

When printing to the device, if the *JOBCCSID value is specified on the CHRID parameter of the CRTPRTF command:

- Constant text from an externally described printer file is converted from the CCSID of the printer file to the CCSID of the job.
- Character data sent to output fields is assumed to be already converted to the job CCSID.

If the printer data stream is tagged with the character identifier CHRID derived from the CCSID of the job, the CHRID value is used by the printer to interpret the data. The CHRID value is ignored for printers not supporting this function.

If a specific value is set for the CHRID parameter on the CRTPRTF command:

- For externally described printer files, fields that specify the CHRID DDS keyword use the CHRID value specified on the printer file. The remainder of the file is printed as if *DEVD was specified for the CHRID parameter on the CRTPRTF command.
- For program-described printer files, the printer data stream uses the CHRID value specified on the printer file.

If the *DEVD parameter is specified on the CHRID parameter of the CRTPRTF command, no conversion is performed.

The CHRID information is determined by either the printer hardware or the device description. If the CHRID information is obtained from the device description, it is then sent to the printer.

9.6 Using Alternative Character Sets and Code Pages for Printer Output

If you want to print in a multilingual environment, you should know how to specify fonts, character identifiers (CHRIDs), character sets, code pages, and coded fonts on the printer file corresponding to a specific language or language group. These parameters also must refer to the type of the selected target printer.

9.6.1 Character Sets and Code Pages

Character sets are used with code pages to determine how each character appears in the printed output. Code pages consist of hexadecimal identifiers (code points) assigned to character identifiers. For example, in code page 037 (EBCDIC), the letter e is assigned a code point of hex 85.

Code pages come in two types:

- Character set and code page combination (CHRID)
- Code page (stand-alone)

  In multinational environments, data in one national graphic character set may need to be printed on devices that support another national character set. This is particularly true of characters with accents and other characters with diacritical marks (such as ç, ñ, and ü). For example, assume that a physical file on the system contains data in the Basic French character set and includes the character é. In the code page used with the Basic French character set, this character is hex C0. The data may have been entered on a display device that can handle the character or may have been sent to the system from another system over a communications line. When hex C0 is sent to a printer that is set up for the United States basic character set, the hex C0 is printed as {. Depending on the printer and the hexadecimal value sent, the hexadecimal value may be an unprintable character. The way the printer handles a specific hex code point (for example, hex C0) depends on the current value of the CHRID parameter in the printer file.

9.6.2 Character Set and Code Page Combination (CHRIDs)

A CHRID is a combination of a specific graphic character set and a specific code page.

CHRIDs are used for fonts that are resident on the printer. They are used in conjunction with a Font ID to obtain a resident font.

9.6.3 Naming Convention for CHRIDs

The names of character identifiers (CHRIDs) are made up of the two elements (graphic character set and code page). These two elements define a collection of characters. The following example shows multinational CHRID 697 500.

697 This is the name of the graphic character set. Some graphic character sets identify a character set that is a subset of the code page. Others identify a character set that is equivalent to the code page.

500 This is the name (number) of the code page.

9.6.4 Selecting CHRIDs

CHRIDs are selected by specifying a certain value for the CHRID parameter on the printer file. An additional font value must be specified for the FONT parameter on the printer file.

The CHRID value can be interpreted in several ways:

- With an explicit value specified for the CHRID parameter, the printer interprets the data as if the data were in the character set and code page specified.
- With CHRID(*SYSVAL) specified, the printer file takes the value specified in the QCHRID system value when the output is created.
- With CHRID(*DEVD) specified, the printer uses the CHRID that was set with the device control panel or that was specified when the printer device description was created.
- With CHRID(*JOBCCSID) specified, the printer interprets the data as if the data were in the character set and code page associated with the CCSID for the current job. For more information, see the *National Language Support Guide.*

If a CHRID is specified for a printer on which that CHRID is not supported , a message is sent to the operator. The AS/400 system substitutes the CHRID that most closely matches the one required by the application.

For program-described printer files, the value of the CHRID parameter determines the code page and character set used to print the data. However, for externally described printer files, the CHRID parameter is used only for fields that also have the CHRID DDS keyword specified. Fields that do not have the CHRID DDS keyword use the code page and character set as if CHRID(*DEVD) had been specified for the CHRID parameter on the printer file.

9.7 Character Identifier (CHRID) Values Supported

The following table lists all the character identifiers, the related national language groups, the correct code page, and which IBM printers support which character identifier. Please refer to the table D-7 in the *Printer Device Programming*, SC41-5713, for new NLVs and additional printers.

Table 21 (Page 1 of 4). CHRID Values and Applicable Printers (CHRID Parameter)

	Code Pages		Printers[1]							
Language Groups	CHRID Code Page xxx yyy[2,3]	Sub-stitute Code Page yyy[2,4]	3812[5] 3816[5]	4214[5]	4224[5] 4230[5] 4247[5]	4234[5]	5219	5224 5225	3112 3116 3912 3916 4028[5]	3130 3160 3935
Major Groups										
International (and US ASCII)	103 038	500	Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
Multinational	697 500		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
	337 256	500	Yes	N/A	N/A	N/A	N/A	Yes	Yes	Yes
	697 256	500	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
United States	101 037		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 037		Yes	Yes	N/A	Yes	N/A	N/A	Yes	Yes
Individual Countries/Languages										
Arabic	697 361		Yes	N/A	Yes	N/A	N/A	N/A	Yes	N/A
Arabic X/B	235 420	500	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
	697 420		Yes	N/A	4224-No 4230-Yes 4247-Yes	IPDS[7]	N/A	N/A	Yes	N/A
Austria/ Germany[6]	265 273		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 273		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
Austria/ Germany	697 286	273	Yes	N/A	Yes	IPDS[7]	N/A	N/A	N/A	Yes
	317 286		Yes	N/A	Yes	IPDS[7]	N/A	N/A	N/A	Yes
Belgium[6]	697 500		N/A	Yes	Yes	Yes	Yes	Yes	N/A	Yes
	269 274		N/A	Yes	Yes	Yes	Yes	Yes	N/A	Yes
	697 274		N/A	Yes	Yes	Yes	N/A	N/A	N/A	Yes
Brazil[6]	273 275		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 275		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
Canadian French[6]	277 276	297	Yes	N/A	N/A	N/A	Yes	Yes	Yes	Yes
	341 260	037	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
	697 260		Yes	N/A	N/A	IPDS[7]	N/A	N/A	Yes	N/A
Canada-Bilingual	038 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
	039 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Canada-English	037 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Cyrillic	960 880		N/A	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
Cyrillic multilingual	1150 1025				4224-No 4230-Yes 4247-Yes	IPDS[7]				Yes
Czechoslovakia/Czech	083 257		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Czechoslovakia/Slovak	085 257		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Denmark/Norway[6]	281 277		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 277		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
Denmark/Norway	697 287	277	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
	321 287		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
Farsi	1219 1097		N/A	N/A	Yes	IPDS[7]	N/A	N/A	N/A	N/A
Finland/Sweden[6]	285 278		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 278		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
Finland/Sweden	697 288	278	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
	325 288		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
France (1977)[6]	289 279	297	Yes	N/A	N/A	N/A	N/A	Yes	Yes	N/A

Table 21 (Page 2 of 4). CHRID Values and Applicable Printers (CHRID Parameter)

	Code Pages		Printers[1]							
Language Groups	**CHRID Code Page xxx yyy**[2,3]	**Sub-stitute Code Page yyy**[2,4]	**3812**[5] **3816**[5]	**4214**[5]	**4224**[5] **4230**[5] **4247**[5]	**4234**[5]	**5219**	**5224 5225**	**3112 3116 3912 3916 4028**[5]	**3130 3160 3935**
France (1980)[6]	288 297		Yes	Yes	Yes	Yes	Yes	N/A	Yes	Yes
	697 297		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
France	251 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
France/Belgium	031 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Germany/Austria	028 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
	029 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Greek	218 423		N/A	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
	925 875		N/A	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
Hebrew	941 424		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
	697 424		Yes	N/A	4224-No 4230-Yes 4247-Yes	IPDS[7]	N/A	N/A	Yes	N/A
	1147 803		N/A	N/A	4224-No 4230-Yes 4247-Yes	IPDS[7]	N/A	N/A	N/A	Yes
Hong Kong	119 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
Hungary	091 257		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Icelandic	697 871		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
	697 029		Yes	N/A	N/A	N/A	N/A	N/A	Yes	N/A
Italy[6]	293 280		Yes	Yes	Yes	IPDS[7]	Yes	Yes	Yes	Yes
	697 280		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
Italy	041 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Japan-English[6]	297 281		Yes	Yes	Yes	IPDS[7]	Yes	Yes	Yes	Yes
	697 281		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
	068 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
	069 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Japan-Katakana[6]	332 290		Yes	N/A	Yes	Yes	N/A	Yes	Yes	Yes
Korean	933 833		N/A	N/A	4230-Yes 4247-Yes 4224-N/A	IPDS[7]	N/A	N/A	N/A	N/A
	697 290		Yes	N/A	N/A	IPDS[7]	N/A	N/A	Yes	N/A
Latin	959 870		N/A	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
Latin America/Puerto Rico	025 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Netherlands	043 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Norway/Denmark	055 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Poland	093 257		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Portugal[6]	301 282		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 282		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
Portugal	697 831	282	Yes	N/A	Yes	N/A	N/A	N/A	Yes	N/A
	063 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Romania	087 258		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
South Africa	081 258		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Spain[6]	305 283	284	Yes	N/A	Yes	Yes	Yes	Yes	Yes	Yes
	697 283	284	Yes	N/A	Yes	N/A	N/A	N/A	Yes	Yes
	697 289	284	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
	329 289		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
	045 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Spanish Speaking[6]	309 284		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 284		Yes	Yes	Yes	Yes	Yes	N/A	Yes	N/A
	1149 284		N/A	N/A	N/A	N/A	Yes	N/A	N/A	Yes
Sweden/Finland	052 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
	053 256		Yes	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Switzerland/French	048 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A

Table 21 (Page 3 of 4). CHRID Values and Applicable Printers (CHRID Parameter)

	Code Pages		Printers[1]							
Language Groups	**CHRID Code Page xxx yyy**[2,3]	**Sub-stitute Code Page yyy**[2,4]	**3812**[5] **3816**[5]	**4214**[5]	**4224**[5] **4230**[5] **4247**[5]	**4234**[5]	**5219**	**5224 5225**	**3112 3116 3912 3916 4028**[5]	**3130 3160 3935**
Switzerland/German	049 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Thai	1102 889		N/A	N/A	Yes	IPDS[7]	N/A	N/A	N/A	N/A
	938 838		N/A	N/A	4230-Yes 4247-Yes 4224-N/A	IPDS[7]	N/A	N/A	N/A	N/A
Turkish	965 905		N/A	N/A	4230-Yes 4247-Yes 4224-Yes	IPDS[7]	N/A	N/A	Yes	Yes
	1152 1026		N/A	N/A	4230-Yes 4247-Yes 4224-N/A	IPDS[7]	N/A	N/A	N/A	Yes
United Kingdom[6]	313 285		Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	697 285		Yes	Yes	Yes	Yes	N/A	N/A	Yes	Yes
U.K./Israel	066 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
U.K./Israel-Latin	067 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
USA-Accounting	017 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
USA/Australia	001 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Countries of the former Yugoslavia	410 890		N/A	N/A	Yes	IPDS[7]	N/A	N/A	N/A	N/A
Countries of the former Yugoslavia-Latin	095 257		N/A	N/A	N/A	N/A	Yes	N/A	N/A	N/A
Non-country Languages										
APL	697 293		Yes	N/A	N/A	IPDS[7]	N/A	N/A	Yes	N/A
	380 293		Yes	N/A	4224-N/A 4230-N/A 4247-N/A	IPDS[7]	N/A	N/A	Yes	Yes
APL Alternate	697 310		Yes	N/A	4224-No 4230-Yes 4247-Yes	IPDS[7]	N/A	N/A	Yes	Yes
	963 310		Yes	N/A	4224-No 4230-Yes 4247-Yes	N/A	N/A	N/A	Yes	Yes
ASCII	103 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
DCF Compatibility	1132 1002		Yes	N/A	4230-Yes 4247-Yes 4224-No	IPDS[7]	N/A	N/A	Yes	Yes
DCF US Text	1133 1003		N/A	N/A	4230-N/A 4247-N/A 4224-N/A	N/A	N/A	N/A	N/A	Yes
DCF text with numeric space	1259 1068		N/A	N/A	4230-N/A 4247-N/A 4224-N/A	N/A	N/A	N/A	N/A	Yes
EBCDIC	101 256		Yes	N/A	N/A	N/A	Yes	N/A	Yes	Yes
GML List Symbols	1258 1039		N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes
International Typographic	697 361		Yes	N/A	N/A	N/A	N/A	N/A	Yes	N/A
OCR (unregistered)	697 340	500	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
OCR A	697 892	500	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
	968 892		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
OCR A (unregistered)	580 340	892	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
OCR B	697 893	500	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
	969 893		Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	Yes
OCR B (unregistered)	590 340	893	Yes	N/A	Yes	IPDS[7]	N/A	N/A	Yes	N/A
Personal Computer	697 437		Yes	N/A	4224-No 4247-Yes 4230-Yes	N/A	N/A	N/A	Yes	N/A
Symbols	340 259		Yes	N/A	N/A	N/A	N/A	N/A	Yes	Yes
Symbol-Selectric	201 259	500	Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A

Table 21 (Page 4 of 4). CHRID Values and Applicable Printers (CHRID Parameter)

	Code Pages		Printers[1]							
Language Groups	**CHRID Code Page xxx yyy**[2,3]	**Sub-stitute Code Page yyy**[2,4]	**3812**[5] **3816**[5]	**4214**[5]	**4224**[5] **4230**[5] **4247**[5]	**4234**[5]	**5219**	**5224 5225**	**3112 3116 3912 3916 4028**[5]	**3130 3160 3935**
Symbol-6640	202 259	500	Yes	Yes	N/A	N/A	Yes	N/A	Yes	N/A
Symbol-6670	203 259		Yes	N/A	N/A	N/A	Yes	N/A	Yes	N/A
Symbols, Adobe	1257 1087		N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes
Symbols Set 7	697 259		Yes	N/A	N/A	N/A	N/A	N/A	Yes	N/A
Symbols Mod Set 7	1191 1091		N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes
Symbols Set 8	630 363		N/A	N/A	N/A	N/A	N/A	N/A	N/A	Yes

1 The 5256, 5262, and 4245 workstation printers do not support the hardware function required for alternative CHRID processing. If a non-default character set and code page is selected for these printers, a diagnostic message is sent and processing continues using the default character set.

2 If the printer supports the code page specified (the second part *(yyy)* of the CHRID parameter) but not the character set *(xxx)*, the character set supported by the printer is used along with the specified code page. For example, if 337 037 (extended character set for displays) is specified for the 5224 and 5225 printers, the print file is printed with character set 101, code page 037.

3 In some cases, the printer substitutes a supported code page for an unsupported code page. Consult the various printer reference guides for defaults on the code page mapping.

4 If the printer does not support or map the code page specified, an attempt is made by the system to find a satisfactory substitute. This column shows the code page substitutes that are made if the specified printer supports the substitute.

5 The 3812, 3816, 4214, 4224, 4230, 4234, and 4247 printers support character set 697 (full character set). This character set contains all of the characters in the limited character sets. For example, 697 037 contains all of the characters in 101 037 or 337 037 (extended character set for displays).

6 This language is considered a primary language group. All other entries, if any, under the primary language group are considered as alternative language groups.

7 Supported by 4234 IPDS version only.

8 Supported by 4234 SCS version only.

9.8 Font Character Sets and Font Global Identifiers (FGID)

Fonts are a family or assortment of characters. Three elements usually provide a font identity:

- Type Family:

  Courier is an example of a type family.

- Typeface:

  Style, weight (for example, italics or bold), and width (normal or expanded) define typeface. Normal means the usual size of characters, while expanded means that the character is wider than normal.

- Type Size:

  Fonts can range from small (4 point) to large (72 point).

  For example, a font may be identified as:

  Type family Sonoran Serif
  Typeface Roman medium normal
  Type Size 10-point

9.8.1 Font Character Sets

Fonts are named in a number of ways. One way is with a character set name. These character sets are downloaded to the printer. Multiple code pages can be used with a single character set.

Some font character sets come with the AS/400 system; some can be downloaded from a System/390 to an AS/400 system; some can be received from another AS/400 system; and some are available as licensed programs (for example, AS/400 Font Collection, 6548-113).

The following printers accept downloaded font character sets:

- 3112 (has resident fonts also)
- 3116 (has resident fonts also)
- 3130 (has resident fonts also)
- 3160 (has resident fonts also)
- 3812 (has resident fonts also)
- 3816 (has resident fonts also)
- 3820
- 3825
- 3827
- 3828 (MICR printer)
- 3829
- 3831
- 3835
- 3900
- 3912 (has resident fonts also)
- 3916 (has resident fonts also)
- 3930 (has resident fonts also)
- 3935 (has resident fonts also)
- 4028 (has resident fonts also)
- 4312 (has resident fonts also)
- 4317 (has resident fonts also)
- 4324 (has resident fonts also)

Note: Any printer attached to Print Services Facility for OS/2 (PSF for OS/2) accepts downloaded fonts.

The use of font character sets provides consistent or similar fonts across printers. For example, a document created at one location using a specific font character set may be sent to a different location, printed on a different model printer, and still look the same. The printers in the preceding list support font character sets that are 240-pel with the exception of the 3112, 3116, 3160, 3912, 3930, 3935, 4028, 3130, 3935, 4312, 4317, 4324 printers that support 300-pel fonts. A pel is a picture element, representing the number of dots in a square inch (for example, 240 across and 240 down).

The 3130 printers support both 240-pel and 300-pel fonts. The operator can select which mode the printer is in through the printer operator panel.

9.8.2 Naming Convention for Font Character Sets

Font character set names on the AS/400 system can be up to 8 characters long. Each character or group of characters tells something about the font character set.

For example, in the font character set name C0D0GT10:

C0 The C0 means that this object is a font character set.

D The D indicates the origin of the font. In this example, C0D0GT10 is a font character set designed for Document Control Facility (DCF) for a 3800 Model 1 printer or a 3825 printer.

0 This 0 indicates that this font is for uniformly spaced and mixed-pitch font character sets.

GT10 The GT10 indicates the type family, typeface, and pitch for uniformly spaced and mixed-pitch fonts. In this example, GT10 means that this font character set is a Gothic Text style and the characters are 10 pitch or 10 characters per inch.

For more information about font character sets, see the manual *About Type: IBM's Technical Reference for 240-Pel Digitized Type*, S544-3516.

9.8.3 Selecting Font Character Sets

Selecting a font character set to use with an application program is done by specifying the 8-character font character set name as the value on the FNTCHRSET parameter of the printer file.

If you choose to use font character sets with your applications, you must also specify a code page (by providing a value for the CDEPAG parameter of the printer file being used).

The selected code page should correspond to the selected language group.

Printer file Example:

```
Printer Device Type   .... *AFPDS
  ..

  ..

FNTCHRSET
```

```
CHRSET      .... C0H200A0    Helvetica Latin1 Roman Med 11 Pt.
LIB         .... QFNT300LA1  Library for Latin1 Charsets 300-pel
CODE PAGE   .... T1V10273    Germany F.R./Austria (EBCDIC)
LIB         .... QFNTCDEPAG  FontCodepage library
```

A wrong code page not corresponding to the selected language group impacts the printout of the language oriented characters.

AS/400 AFP font collection libraries must be installed on the AS/400 system. The referred libraries must be in USERLIBL or SYSLIBL of the user initiating the print job. PSF/400 must be installed on the AS/400 system to download the objects. The target printer must be a 300 pel IPDS Laser printer with DEVD set to AFP *YES.

9.8.3.1 Substituting Font Character Set

Substitution is determined by the AS/400 system based on which font character sets are specified in the application, the type of printer to be used, and the value assigned to the fidelity parameter of the printer file being used (*CONTENT or *ABSOLUTE).

Example 1: Assume

- The application calls for font character set C0D0GB10 (Gothic Bold, 10 pitch).
- The printer supports only resident fonts.
- The fidelity parameter value is *CONTENT.

In this example, the spooled file prints with substituted font ID 39 (Gothic Bold 10 pitch) because the fidelity parameter value is *CONTENT. If the fidelity parameter value is *ABSOLUTE, the spooled file is held on the output queue and does not print.

Example 2: Assume

- The application calls for FGID 51 (Matrix Gothic).
- The printer supports only downloaded font character sets.
- The fidelity parameter value is *CONTENT.

In this example, the spooled file prints. The AS/400 system substitutes a font character set (C0S0CR10, Courier Roman 10 pitch) for FGID 51. This is not an exact match. The AS/400 system matched (as closely as possible) the font character to the FGID specified in the application.

Note: In this example, if the fidelity parameter is *ABSOLUTE, the spooled file is **held**.

9.8.4 Font Global Identifiers (FGIDs)

Another method of naming a font is by a font global identifier (FGID). An FGID names a type family and a typeface.

FGIDs are identified by a number such as 3, 8, 11, 46, 223, 2305, and so on.

There is a different FGID assigned for the same type family but different typeface. For example, a Courier Roman Medium 10 pitch (characters-per-inch) is FGID 11 and Courier Roman Bold 10 pitch (characters per inch) is FGID 46.

Printers with resident fonts use FGIDs to name the resident fonts. Depending on the technology used with the printer, resident fonts can be stored on font cards, diskettes, in the memory of the printer, or mechanically on a font element or daisy wheel.

The following printers have resident fonts:

- 5219
- 4224
- 4230
- 4234 Models 8 and 12
- 4247
- 3112 (can also accept downloaded fonts)
- 3116 (can also accept downloaded fonts)
- 3130 (can also accept downloaded fonts)
- 3160 (can also accept downloaded fonts)
- 3812 (can also accept downloaded fonts)
- 3816 (can also accept downloaded fonts)
- 3912, 3916, or 4028 (can also accept downloaded fonts)
- 3935 (can also accept downloaded fonts)
- 4312,4317,4324 (can also accept downloaded fonts)

To find out which fonts are supported by a printer, check the reference manual for that printer.

9.8.5 Selecting Resident Fonts

Selecting a resident font to use with an application program is done by specifying an FGID value on the FONT parameter of the printer file.

9.8.6 Font Substitution

Substitution can be one FGID for another, an FGID for a font character set, or a font character set for an FGID.

Example 1: Assume

- Your application calls for a font character set (FNTCHRSET specified on the printer file), for example, C0S0CR10 for Courier Roman medium 10 pitch.
- The printer is a 4224 and has resident fonts identified by FGIDs.
- FGID 11 is substituted for C0S0CR10 and sent to the printer.

In this example, the AS/400 system substitutes a font that is resident on that printer.

Example 2: Assume

- Your application calls for a font (specified on the FONT parameter of the printer file). The font specified is font 26 (Gothic Matrix, Roman medium 10 pitch) and the printer is a 3812.
- You decide to print the document on a 4019 printer. Font 26 is not supported on the 4019.

In this example, the AS/400 system substitutes font 11 (Courier, Roman medium 10 pitch)

Example 3: Assume

- Your application uses a font (specified on the FONT parameter of the printer file). The font specified is font 40 (Gothic, Roman medium 10 pitch).
- The printer you are going to print on supports only font character sets (for example, a 3827).

In this example, the AS/400 system substitutes font character set C0D0GT10, Text, Roman medium 10 pitch).

9.8.7 Stand-Alone Code Pages

Stand-alone code pages supply consistent or similar characters across systems. For example, a document created at one location using a specific code page can be sent to a different location, printed on a different model printer, and still look the same.

Stand-alone code pages must be downloaded to the printer for use.

The following printers accept downloaded code pages:

- 3112 (has resident fonts also)
- 3116 (has resident fonts also)
- 3130 (has resident fonts also)
- 3160 (has resident fonts also)
- 3812 (has resident fonts also)
- 3816 (has resident fonts also)
- 3820
- 3825
- 3827
- 3828 (MICR printer)
- 3829
- 3831
- 3835
- 3900
- 3912 (has resident fonts also)
- 3916 (has resident fonts also)
- 3930 (has resident fonts also)
- 3935 (has resident fonts also)
- 4028 (has resident fonts also)
- 4312,4317,4324 (has resident fonts also)

9.8.8 Naming Convention for Code Pages

The same as character sets, code pages are named in a number of ways. One way is with a code page name. These code pages are downloaded to the printer. The code page name can be up to 8 characters long.

Another way is with a code page global identifier (CPGID). CPGIDs are printer-resident code pages and have numbers for names (for example, 259 or 500). Generally, printers with resident fonts use CPGIDs to name the printer-resident code pages. CPGIDs are also used within CHRIDs.

For example, in the code page name T1V10500:

T The T means that this object is a code page.

1 This is always a 1.

V1 The V1 means that this is version 1 of this code page.

0500 The 0500 is the code page name, number, or category. In this example, 500 is the code page name, multilingual #5 (EBCDIC).

9.8.9 Selecting Code Pages

Code pages are selected by specifying a certain value for the code page (CDEPAG) parameter on the printer file.

If you choose to use code pages with your applications, you must also specify a font character set (by providing a value for the FNTCHRSET parameter on the printer file being used).

Printer file example:

```
  ..

Printer Device Type    .... *AFPDS
  ..

  ..

FNTCHRSET
    CHRSET      .... C0H200A0    Helvetica Latin1 Roman Med 11 Pt.
    LIB         .... QFNT240LA1  Library for Latin1 Charsets 240-pel

    CODE PAGE   .... T1V10500    Multilingual #5 (EBCDIC)
    LIB         .... QFNTCDEPAG  FontCodepage library
```

AS/400 AFP font collection libraries must be installed on the AS/400 system. The referred libraries must be in USERLIBL or SYSLIBL of the user initiating the print job. PSF/400 must be installed on the AS/400 system to download the objects. The target printer must be a 240-pel IPDS Laser printer with DEVD set to AFP *YES.

A wrong code page not corresponding to the selected language group impacts the printout of the language oriented characters.

9.8.10 Substituting Code Pages

Substitution of code pages occurs for the following reasons:

- The application specifies a code page that is resident on a printer and the printer being used does not have resident code pages.
- The application specifies a code page that is resident on the host system (AS/400 system) and the printer being used has resident code pages (not capable of accepting downloaded code pages).
- The job requesting the code page is not authorized to it.
- The code page cannot be found.
- The job is not authorized to the library where the code page is stored.

9.8.11 Coded Fonts

A coded font is the pairing of a font character set and a code page. Coded fonts allow users to specify a font character set and a code page with one value specified on the printer file.

Coded fonts available on the AS/400 system can be viewed by using the Work with Font Resources (WRKFNTRSC) command.

Coded font names are read by the AS/400 system and then translated to a font character set and a code page. These two elements are then sent to the printer.

9.8.12 Naming Convention for Coded Fonts

Unlike other uniformly spaced and mixed-pitch font components, coded font names are generally shortened by excluding the origin and reserved characters (the first two characters of their name). This is necessary because some Advanced Function Printing (AFP) licensed programs accept only six characters for coded font names. However, some applications can use coded fonts named with six or eight characters.

Coded font names on the AS/400 system are six or eight characters long. Each character or group of characters tells something about the coded font.

For example, in the coded font name X0GT10:

X0 The X0 means that this object is a coded font.

GT10 The GT10 indicates the type family, typeface, and pitch for uniformly spaced and mixed-pitch fonts. In this example, the GT10 means that this font character set is a Gothic Text style and the characters are 10 pitch or 10 characters per inch.

To find out which font character set and code page make up a coded font name, use the Work with Font Resources (WRKFNTRSC) command. This command allows you to specify the font resource to be worked with, the library it is in, and the attribute (coded font).

9.8.13 Selecting Coded Fonts

A coded font is selected by specifying the coded font name as the value on the coded font (CDEFNT) parameter of the printer file.

You can use the Work with Font Resources (WRKFNTRSC) command to view the coded fonts that are available on the AS/400 system.

Printer file example:

```
  ..

Printer Device Type    .... *AFPDS
  ..

  ..

Coded Font
  CDEFNT    .... X0H210AC    for Character set C0H200A0 and Code page T1V10500
  LIB       .... QFNTCF_LA1  Library for Latin1 Coded fonts
```

AS/400 AFP font collection libraries must be installed on the AS/400 system. The referred library must be in USERLIBL or SYSLIBL of the user initiating the print job. PSF/400 must be installed on the AS/400 system to download the object. The target printer must be a IPDS Laser printer with DEVD set to AFP *YES.

Note: If you use a coded font within your printer file, you only can print those characters corresponding to the character set and code page within the coded font. If you want to print a spooled file containing characters for a different language group, you must change the coded font value corresponding to the character set and code page of the different language group. Changing or overriding the printer file must be done or use different printer files containing different coded fonts.

9.8.14 Substituting Coded Fonts

No substitution of coded fonts takes place on the AS/400 system. If the coded font is not available, the document will not print.

9.9 Multilingual Support for ASCII Printers

Using an ASCII printer as the target printer for printing in a multilingual environment may cause problems as a correct printout showing the correct characters corresponding to the chosen CHRID within the printer file depends on many factors:

- Type of ASCII printer (manufacturer, type, model)
- Type of ASCII printer data stream (PPDS, PCL4/5, other)
- Type of supported resident ASCII code pages and fonts
- Type of additional code pages and fonts available through font cards
- Type of connection and emulator (InfoWindow, PC emulator)
- Type of configuration (using or not using Host Print Transform)

In general, we recommend using the Host Print Transform support on the AS/400 system to enhance and optimize the support for ASCII printers along with a corresponding EBCDIC/ASCII code page mapping to receive a language-oriented correct printout.

9.10 Host Print Transform Introduction

Using the host print transform function eliminates the problems of different ASCII data streams and limited printer support as well as providing other printing advantages. Customizing printers that use host print transform function provides more functions for more printer types than other methods of customizing printers.

9.11 Host Print Transform Functional Overview

Up until now, you have been able to attach ASCII printers to the AS/400 system using the three main approaches: through an ASCII workstation controller, through a non-programmable terminal, or through a PC. Taking into account the different emulation programs that you can use, there are approximately 15 different ways to attach an ASCII printer.

The AS/400 host print transform has three main objectives:

- To provide a single method of configuring an ASCII printer on the AS/400 system regardless of the attachment method.
- To help customers to fully utilize the functions available on their ASCII printers.
- Perhaps most importantly, to ensure consistent output regardless of the method of attachment.

The host print transform is an OS/400 function that converts an SNA character string (SCS) data stream or Advanced Function Printing data steam (AFPDS) into an ASCII data stream. You can use the host print transform on the AS/400 system to convert the EBCDIC data stream to an ASCII data stream for ASCII printers attached in the following ways:

- ASCII printers attached to twinaxial displays
- ASCII printers attached to a personal computer through 5250 emulation products (such as Client Access/400)
- ASCII printers directly attached to the AS/400 system through the ASCII workstation controller
- AS/400 ASCII LAN-attached printers

Note: The host print transform function can also be used when you send a spooled file with the SNDTCPSPLF (TCP/IP LPD and LPR) command, or when using a remote output queue by specifying TRANSFORM(*YES) in the CRTOUTQ command.

The host print transform lets you choose among a wide number of supported ASCII printers. Alternatively, if your printer is not supported, it is possible to create your own “printer profile,” called a workstation customization object (WSCST). These profiles are similar in function to the printer function tables (PFT) used under the workstation function. To initiate the Host Print Transform function, you simply must set the Transform parameter on the device description of the ASCII printer to *YES and additionally specify the Manufacturer Type and Model parameter (MFRTYPMDL) to the *WSCST driver that matches the name and model of the physically attached ASCII printer model. Press F4 at the MFRTYPMDL parameter to obtain a complete list of the system supported printers to select from.

See the *Workstation Customization Programming*, SC41-3605, for detailed information about Host Print Transform.

9.12 Customizing Your ASCII Printer Driver

After you install a new primary language on your AS/400 system, you may want to print your system or application printouts corresponding to the new language. If you use ASCII printers as target printers, you may see incorrect printouts as the currently used *WSCST ASCII printer driver may use only the default EBCDIC/ASCII Code page mapping, which does not include the ASCII code page you need to print with on your target printer.

EBCDIC-to-ASCII mapping tables convert an EBCDIC character specified in an application data stream into an ASCII character code value (for that same character). The EBCDIC/ASCII mapping table used for a given ASCII printer depends on the manufacturer, type, and model configured for the printer. Different types of ASCII printers support different ASCII code pages. The code

page in use at any time is determined by a command to select the ASCII code page.

The workstation customization functions allow you to customize code page support for an ASCII printer. You can:

- Customize EBCDIC-to-ASCII code page mapping.
- Support additional ASCII code pages (add additional language support).
- Override the default ASCII code page.

See the *Workstation Customization Programming*, SC41-3605, for detailed information and examples about customizing code page support (Chapter 10 through Chapter 19).

Chapter 10. Central Site Considerations

When installing an AS/400 system at a site where multiple languages are used, the system administrator must ensure that the system has the correct definitions for cultural system values and job attributes.

Central site application providers should be aware of many considerations before they deliver their applications to different countries with different languages. The providers must be sure they do not impact the CCSIDs of the current status of the customers database. They also should consider preparing or modifying their printer files to use in the customer's printing environment and that are used with different character sets and code pages to match the customers demand for language-oriented correct printouts.

10.1 Multilingual System Setup

One single AS/400 system is capable of handling multiple languages. This facility is enabled in the base operating system, and secondary languages are used to provide cultural functions for IBM supplied licensed programs. Additionally, secondary languages may provide translated parts of the operating system and licensed programs.

When different language users are connected to the same AS/400 system, they can store and access data:

1. All users access the same set of database files
2. Each language has its own set of database files

Figure 17 describes a multilingual single system with German as the primary language and English and French as secondary languages with all users entering data into the same database file.

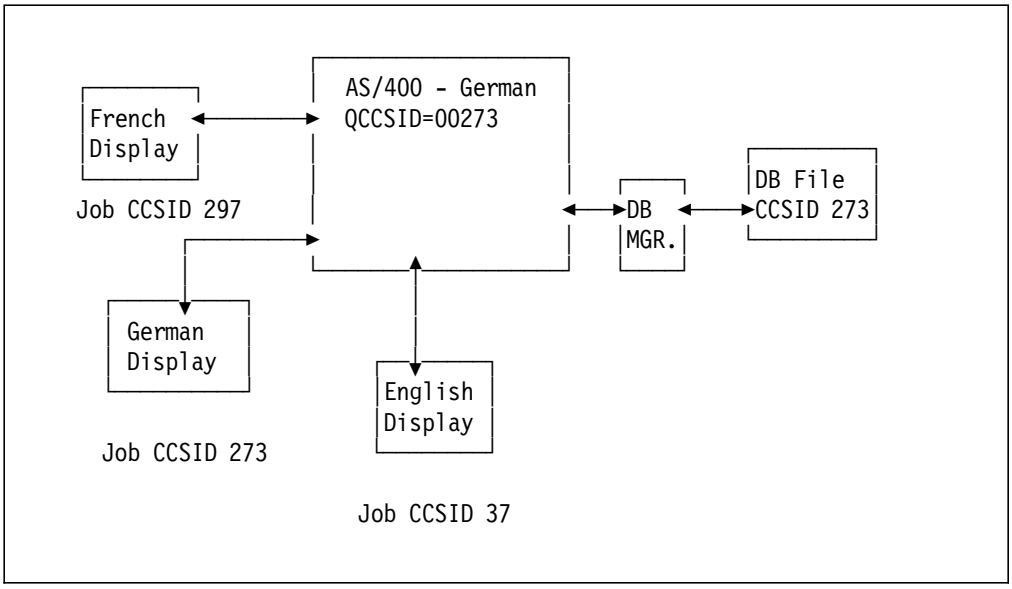


Figure 17. Multilingual Single System Setup

You should make sure that the application is enabled for using CCSIDs. Refer to Section 10.2, "Application Installation Considerations" on page 104 and Section

10.2.1, “CCSID Considerations for Application Providers” on page 105 for additional information.

When database sharing takes place, it is necessary that all users have the CCSID and the language they use correctly defined in their job attributes.

Since the system in Figure 17 on page 101 has German as the primary language installed, the cultural system values are as shown in Table 22, where the system value QCCSID is changed to 273.

Table 22. German Cultural System Values

System Value	Description	Value
QCCSID	Coded character set identifier	273
QCHRID	Character set/code page	697 273
QCNTRYID	Country identifier	DE
QDATFMT	Date format	DMY
QDATSEP	Date separator	.
QDECFMT	Decimal format	J
QKBDTYPE	Character set/code page for keyboard	AGB
QLANGID	Language identifier	DEU
QTIMSEP	Time separator	:

Since the system values are properly set for the German users, no special action has to be taken when creating German user profiles. For the English and French users, the cultural job attributes have to be set according to their needs. This can be accomplished by using an initial program in the user profile, for example, one program for each language:

```
                      Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile . . . . . .   USR1ENG       Name
  ...
Initial program to call  . . . .   ENGLISH       Name, *NONE
  Library  . . . . . .                 Name, *LIBL, *CURLIB
```

```
                      Create User Profile (CRTUSRPRF)

Type choices, press Enter.

User profile . . . . . .   USR1FR        Name
  ...
Initial program to call  . . . .   FRENCH        Name, *NONE
  Library  . . . . . .                 Name, *LIBL, *CURLIB
```

The initial program adds the secondary language library to the user’s library list and then calls a program to set the cultural job attributes according to the secondary language. These values are contained in message ID CPX8416 in message file QCPFMSG.

The following initial program is for English users:

```
/*  Initial program for English users.  Program:  ENGLISH        */
/*  Note:  Program must be run with a user profile having        */
/*  authority to the QSYSLIBL command, for example using         */
/*  QSECOFR and adopted authority.                               */
            PGM
            CHGSYSLIBL LIB(QSYS2924)
            CALL ZJOBATR
            ENDPGM
```

The following initial program is for French users:

```
/*  Initial program for French users.  Program:  FRENCH          */
/*  Note:  Program must be run with a user profile having        */
/*  authority to the QSYSLIBL command, for example using         */
/*  QSECOFR and adopted authority.                               */
            PGM
            CHGSYSLIBL LIB(QSYS2928)
            CALL ZJOBATR
            ENDPGM
```

The following program is the common program that retrieves the cultural values from message ID CPX8416 and sets the user's cultural job attributes. Since the initial program added the secondary language library in front of QSYS, the program will find the message in the secondary language library.

```
/*  Set job cultural attributes.  Program:  ZJOBATR              */
/*  This program retrieves message ID CPX8416 and sets the       */
/*  cultural job attributes.                                     */
PGM
DCL             VAR(&STR)     TYPE(*CHAR) LEN(200)
DCL             VAR(&DATFMT)  TYPE(*CHAR) LEN(4)
DCL             VAR(&DATSEP)  TYPE(*CHAR) LEN(1)
DCL             VAR(&CCSID)   TYPE(*DEC ) LEN(5 0)
DCL             VAR(&TIMSEP)  TYPE(*CHAR) LEN(1)
DCL             VAR(&LANGID)  TYPE(*CHAR) LEN(3)
DCL             VAR(&CNTRYID) TYPE(*CHAR) LEN(2)
RTVMSG          MSGID(CPX8416) MSGF(QCPFMSG) SECLVL(&STR)
CHGVAR          VAR(&DATFMT) VALUE('*' *CAT %SST(&STR 58 3))
CHGVAR          VAR(&DATSEP)  VALUE(%SST(&STR 73 1))
CHGVAR          VAR(&CCSID)   VALUE(%SST(&STR 110 5))
CHGVAR          VAR(&TIMSEP)  VALUE(%SST(&STR 129 1))
CHGVAR          VAR(&LANGID)  VALUE(%SST(&STR 142 3))
CHGVAR          VAR(&CNTRYID) VALUE(%SST(&STR 157 2))
CHGJOB     DATFMT(&DATFMT) DATSEP(&DATSEP) TIMSEP(&TIMSEP) +
           LANGID(&LANGID) CNTRYID(&CNTRYID) CCSID(&CCSID)
ENDPGM
```

All users access the database file tagged with the German CCSID 273 and character data entered from the English and French terminals is being mapped between their job CCSID and CCSID 273 of the database file.

Note: If there are no secondary languages installed, the initial programs for each language can be used to set the cultural job attributes for each language, using the CHGJOB command. As an example, the ENGLISH program looks similar to:

```
/*  Initial program for English users.  Program:  ENGLISH        */
            PGM
            CHGJOB DATFMT(*MDY) DATSEP(/) TIMSEP(':')  +
                   LANGID(ENU) CNTRYID(US) CCSID(37)
            ENDPGM
```

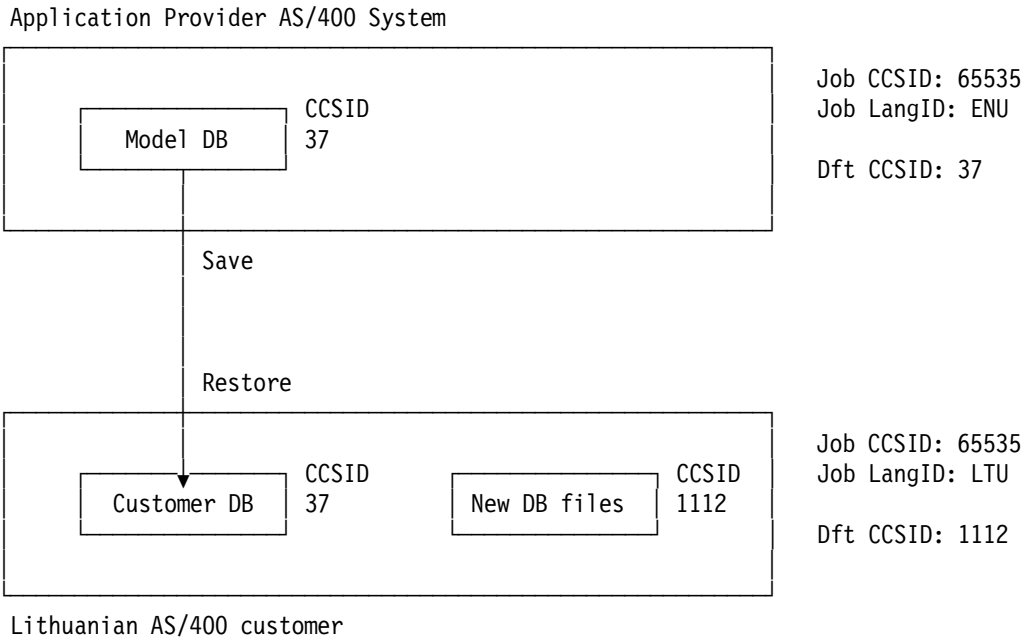
10.2 Application Installation Considerations

It has been a common practice to deliver an application with both application code and encoded database files, and install these components on the customer system.

Application providers must ensure that the database created on the customer system has the correct CCSID. In addition, ensure that source files have the correct CCSID specified, although source code should never use variant characters. Please refer to Section 10.3, “Additional Considerations for Source Files” on page 106 for additional information.

The picture shows how you unintentionally can have a customer database tagged with an improper CCSID.

How to get it wrong:



An application provider develops an application on a U.S. English system. The application is shipped with empty database files; thus, the customer just restores the program and file libraries on the system, in this case in Lithuania. The database files saved from the application provider system still carry the CCSID 37. This works fine as long as the Lithuanian customers keep the job CCSID as 65535 and do not attach a PC client with software that requests CCSID conversion. If they do, they receive unexpected results using the database files tagged with CCSID 37.

Notice that all new database files created on the Lithuanian system have the correct CCSID, 1112.

This problem is not limited to different language groups, but does affect languages that have different default CCSIDs.

How to get it right:

- Create a database on the customer system.

- Have the customer decide which CCSID to use.

Instead of restoring database files onto a customer system, you should provide a program that creates the production database on the customer system. This ensures a proper CCSID for database files.

The Change Physical File (CHGPF) command can be used to change the CCSID tag of a database file, even if it has logical files defined over it. Some restrictions apply using the CHGPF. Please refer to Section 2.4.3, “Changing CCSIDs of Physical Files” on page 15 for more information.

In addition, you have to take care of application updates. For example, if you are using the CPYF command to make changes to the layout of the database files, make sure that your process provides the same tagging of the customer database as before.

10.2.1 CCSID Considerations for Application Providers

- Any character fields within the database that are not used as proper character fields (for example, using a character field as a packed array in your program or a field with control information) must be defined as hexadecimal (data type X or field level CCSID (65535) in DDS) to avoid conversions if the job CCSID is different from the file CCSID.
- Applications that perform their own conversions must ensure that all database files are explicitly defined with CCSID (65535).
- When exchanging database files with systems using a different CCSID, make sure the files are created with the correct CCSID for the content.
- Ensure that the values for language ID, system value QLANGID, and user profile value LANGID are correctly specified, or specify the user′s correct CCSID in the user profile.
- Application providers must ensure that the database created on the customer system has the correct CCSID.
- Avoid hard coding CCSIDs in your DDS specifications for physical files unless absolutely necessary. Instead, change the CCSID of your job (using the CHJOB command) when creating different physical files for different languages. This way, one set of DDS source code can be maintained.
- Never assume that a specific character (symbol) is present in the user′s code page.
- In a multilingual environment where job CCSID is a non-65535 value, we recommend allowing the customer decide the CCSID for the database. The CCSID used for the database should match the CCSID used by the majority of the users to avoid too much conversion.
- Using CCSIDs, the system replaces all characters that cannot be converted between different code pages with X′3F′ (for example, when exchanging data from DBCS to SBCS). If you are using a user-defined data stream (UDDS) to format and lay out your display (instead of using DDS), you may get X′3F′ returned after database reads and insert that data in your user-defined data stream. Because X′3F′ is an invalid value in the 5250 data stream, you may get unpredictable results on the display.

10.3 Additional Considerations for Source Files

For all application development, avoid using characters other than those contained in the invariant character set for naming, literals, and constants in your source code. See Appendix B, “Invariant Character Set” on page 127 for a list of these characters. If you only use invariant characters, you will not experience any problems with your program source. For complete information on application development topics, refer to *International Application Development*, SC41-5603.

However, if you are using variant characters, these are the recommendations we suggest you follow:

- Display, printer file and panel group source:
  - Make sure that the CCSID of the source file matches the keyboard entering the source statements. Compile with that CCSID to enable customers to use the *JOBCCSID support in display and printer files and panel groups. For this support to work correctly, the CCSID in the objects must match the actual content. Therefore, you might need to recompile if you changed the CCSID of the source files.
- Program source:
  - Make sure that the CCSID of the source file matches the keyboard entering the source statements. If you are experiencing problems with compiling your programs, we recommend using CCSID(*HEX) for your source files as an intermediate solution.

There is a difference between the Original Program Model (OPM) compilers and the Integrated Language Environment (ILE) compilers on how CCSIDs of the source files may impact compilation.

- OPM compilers: You must compile the program with the job CCSID matching the source CCSID or the job CCSID should be 65535. If the job CCSID for the compilation is different from the source file CCSID, data management converts the source statements from the source file CCSID to the job CCSID.

  For example, the following source statements were entered on a U.S. system with a U.S. keyboard in a source file with CCSID 37.

```
     *  RPG Source   (Source file created with CCSID 037)
     *
    FFILE1   IF  E                    DISK
    C                     READ FILE1                    80
    C* Test char (example 1)
    C*
    C           FLD1      IFEQ '$'
    C                      ...
    C* Move char (example 2)
    C*
    C                     MOVE FLD1      FLD$
    C                      ...
    C*
    C                     SETON                     LR
```

  In example 1, the compilation of the program always produces executable code, but the dollar sign may represent a different code point than X'5B', if the job CCSID used for compiling is not 37 or 65535.

In example 2 the field name FLD$ may not be accepted by the compiler if compiled with a CCSID other than 037 or 65535 as as the dollar sign is variant and OPM compilers only expect the X'5B' of CCSID 37.

- ILE Compilers: ILE compilers take the CCSID of the primary source file, and any imbedded source files are converted from their CCSID to the primary source file.

Note: Using the Change Physical command (for example CHGPF FILE(srcfile) CCSID(ccsid)) against a source file impacts the last change date/time in all source members.

Recommendation: Recompile source if possible.

10.4 Printer File Recommendations for Application Providers

When designing printer files to be translated into a national language version:

- Use externally described printer files to define records and fields to be printed. Avoid using program-described printer files. Program-described printer files are described inside the high-level language program. Translators trying to translate text imbedded within the program can mistakenly translate literals that are within your program.
- Print data in one national graphic character set on printers that support the corresponding character sets and code pages. Not all printers support all CHRID parameters.
- Use the MSGCON keyword to access the constant text described in the message file. A printer file does not have the MSGID keyword. However, the techniques of direct coding as unnamed output field (literal) and storing text in a database file can be used to specify the constant text in a printer file.

Take culture conventions into consideration when bar codes are being described in the printer file. Different countries have different standards for bar codes.

Consider these parameters within your external printer files:

- PAGESIZE (page size):

  Different countries have different page-size standards (for example, letter, legal, A4, A3, B5).

- OVRFLW (overflow line number):

  The overflow line number must be less than or equal to the page size length.

- CHRID (character set and code page):

  Specify the correct CHRID corresponding to the selected language group. If the CHRID parameter of the printer file is set to *DEVD, the printer uses the character identifier that was set at the printer control panel or specified in the device description.

  Check the setting of the code page at the control panel on the printer and adjust if necessary to the code page (for some non-IBM manufacturers = language ID) you want to print.

  If the CHRID parameter of the printer file is set to a specific value, this value determines the code page and character set used to print the data. For externally described printer files, the CHRID parameter is used only for fields

that also have the CHRID DDS keyword specified. For all other fields, the code page and character set used is the same as if *DEVD was specified.

If the CHRID parameter of the printer file is set to *JOBCCSID, constant text from an externally described printer file is converted to the CCSID of the job. The printer data stream is tagged with the CHRID taken from the job CCSID, using this CHRID value to print the data. When using the *JOBCCSID value on the CHRID parameter, the CHRID DDS keyword is ignored.

Note: Not all code pages and character sets can be handled by all printers. Check the printer manual for supported code pages and character sets (fonts).

- FNTCHRSET (font character set):

  Use this parameter when you want to download character sets and code pages from AS/400 libraries to IPDS laser printers configured with AFP *YES. The printer file device type must be *AFPDS and PSF/400 must be installed on the AS/400 system. This applies to all IPDS printers **except 4224, 4230, 4234, 4247, and 64xx.** They are IPDS Matrix printers and you cannot download fonts, character sets, or code pages to these printers. Use resident fonts and code pages for these printers.

- CDEFNT (coded font):

  Use this parameter only when you want to combine a special character set with a certain code page. It means you only should do this when printing in the same language environment because the language-oriented code page specified in the CDEFNT parameter does not correspond to other languages. For parameter CDEFNT, you need the same printer environment as specified for FNTCHRSET.

  Using downloaded fonts and character sets on your IPDS laser printers means that you have a consistent output (same fonts and ″design″ on the printout) on all of your laser printers. You can be independent about which IPDS Laser printer type is your target printer.

  Using downloaded fonts and code pages, we recommend installing the IBM AFP Font Collection (program number 5648-113) on the AS/400 system. Having installed the libraries from tape or CD-ROM, you find Font character set libraries for all Latin group, Arabian, Cyrillic, Greek, Hebrew, and DBCS languages as well as some special symbol sets. They are available in either 240-pel or 300-pel versions named by different library names (for example, QFNT240LA1/QFNT300LA1). You also have coded font libraries (for example, QFNTCF_LA1) for all language groups and code page libraries including code pages for all language groups. Using the AFP Font Collection, you can print the output of your applications according to all language groups that might be needed from your customers.

  Using the AFP font collection character sets and code pages within your printer files, be careful that the AFP font collection libraries are installed on the customer site. Customers only need to install those libraries that are needed for the selected language groups. The customer site also must have PSF/400 installed and the IPDS laser printers must be set to AFP *YES.

  Make sure that all of the used libraries are in USRLIBL or SYSLIBL of the users initiating the print job; otherwise, font and code page substitution will occur.

10.4.1 Consider Printing Environment of Customer

AS/400 customers normally use a wide variety of different types of printers for the printing their applications. They use different attachment types and different printer data streams depending on parameters within the printer file and the device description of the printer. Application providers usually do not know all of the factors that impact the objective to receive ″correct″ printouts corresponding to different languages. But in any case, they may consider some tips and hints to prepare their applications for different printing and printer scenarios at the customers site.

10.4.2 Customer Printing Environment

The AS/400 customers printing environment depends on the following factors:

- Printer types, names, models, (IBM printers and non IBM printers)
- Type of printer (Laser, LED, Matrix, Inkjet, and so on)
- Resolution of laser printers (240/300/600 Pel)
- Type of supported printer data streams (SCS, IPDS, ASCII)
- Type of connection (Twinax attached, parallel attached through emulator, local or remote through control unit 5X94, channel attached through PSF/2)
- Type of emulator (InfoWindow, PCs, ASCII WSC)
- Device type of configuration (type *SCS (printer type number), *IPDS, *IPDS with AFP *YES, ASCII emulation type)
- Specified Host Print Transform in DEVD for ASCII printers?
- Installed PSF/400? (must be installed for AFP printing if *IPDS printer is set to AFP *YES)
- Installed AS/400 AFP Font Collection (5648-113)?

10.4.3 Printer Environment Impact on Language Oriented Printouts

Application providers should know how different printer types, attachment types, and printer data streams impact a language-oriented printout of the application at the customer site.

- Customer uses IPDS laser printers twinax attached to the AS/400 system:

  Because you cannot download fonts and code pages to IPDS laser printers that are **not** configured with AFP *YES, you can use a resident font along with adjusting the wanted code page through panel selection on the printer. Or the customer may change DEVD of the printer to the selected CHRID. This is only recommended if the specified printer always should print in the selected language. You also may specify the new CHRID using the OVRPRTF command on the external printer files. This CHRID parameter is used only for fields that also have the DDS CHRID keyword specified. For all other fields, the CHRID used is the same as if *DEVD was specified.

  Alternate Options

  If possible, let the customer change the DEVD of IPDS laser printers to AFP *YES. By doing this, you can download every character set and code page from AS/400 libraries to the laser printers. This makes you independent from resident fonts, code pages, device settings, and laser printer types. You receive consistent printout no matter which laser printers are used at the customer site.

Note: AFP printing means that PSF/400 downloads your character sets and code pages to the laser printer. It creates an AFPDS data stream, checks your printer file for font character sets, code pages, and other AFP resources, and converts an AFPDS data stream to IPDS before sending it to the printer. So it has impact on the performance of the customer′s printout, and it takes more time until the printout starts to print. The customer has to enhance the pool size of subsystem QSPL for AFP printing to at least 2MB.

- Customer uses matrix IPDS printers twinax attached to the AS/400 system:

  As you cannot download character sets and code pages to these printers even if the AFP parameter is set to *YES, use resident fonts and code pages corresponding to the selected language from the panel setting of the printer. The customer may change the DEVD of the printer to the selected CHRID. This is only recommended if the specified printer always should print in the selected language. You also may specify the new CHRID using the OVRPRTF command on the external printer files. This CHRID parameter is used only for fields that also have the DDS CHRID keyword specified. For all other fields, the CHRID used is the same as if *DEVD was specified.

- Customer uses SCS printers twinax attached to the AS/400 system:

  If there is no panel setting for the selected language on the panel, these printers cannot print in the selected language.

  Recommendation: Customers should not use these printers for printouts in different languages.

- Customer uses ASCII printers attached to different emulators, network attached (InfoWindow, PCs, Client Access/400, ASCII WSC, ASCII LAN attached, using SNDTCPSPLF through TCP/IP LPD/LPR):

  Recommendation: Customers always should use the Host Print Transform function on the AS/400 system. By specifying the Transform *YES parameter and the MFRTYPMDL parameter matching to the physical ASCII printer type, you receive a ″best fit″ support for your ASCII printer according to the technical built-in print functions of the printer. See 9.9, “Multilingual Support for ASCII Printers” on page 97 for additional information on this subject.

  Let the customer use the panel setting on the printer or let the customer use additional font cards, if available.

10.4.4 Summary

Using external printer files in your applications you can be flexible and simply override CHRIDs or font character sets and code pages according to the selected language and to the selected type of target printer at the customer site. See the recommendations in Section 10.4, “Printer File Recommendations for Application Providers” on page 107.

Downloading font character sets and code pages to IPDS laser printers configured with AFP *YES means your application is independent from the types of laser printers at the customer site and the customer receives consistent printout.

Downloading alternate font character sets and code pages has an impact on the performance of your print jobs.

If necessary, consider customizing *WSCST ASCII printer device drivers according to customers′ requirements for additional languages and provide these customized *WSCST tables with your application.

10.5 Decimal and Thousands Separators and Negative Numbers

There is no universal form for displaying numbers. The AS/400 system provides numerous edit codes to be used on display and printer files, and also up to five user-defined edit codes. These predefined (system and user) edit descriptions are stored in the system library QSYS and are unique to the system. Some of them are dependent on the system value QDECFMT. In addition, they are bound to the device file at creation time depending on the actual system value QDECFMT. Therefore, you cannot have any influence on editing during run time.

If these codes are not sufficient, user-defined edit words can be specified for every output field.

To have different editing options for different NLVs, these options force you to change display and printer files for each output field for each language, or write special program routines using culturally dependent values taken from external message files.

See the *Data Description Specification Reference*, SC41-5712, for more information on edit codes and edit words.

10.5.1 Decimal and Thousands Separators Mismatch between Different Languages

Application providers may experience a problem when compiling display files on an AS/400 system with the system value QDECFMT set to 1 if the production system QDECFMT is set to another value. A value of 1 sets a decimal point as a decimal separator.

However, the compiled display file object is used on an AS/400 system where the system value QDECFMT is set to 2 and the comma is used as the decimal separator. The decimal point is still used as it is determined at compile time and not run time.

Central development sites, in each case, want to avoid recompiling each display file used.

10.5.2 Current Logic on Editing Output and Input Fields on AS/400 System

1. Compile: When DDS source is compiled, any numeric fields having edit codes take the system values QDECFMT and QDATSEP and use these separators in the object. In other words, the separators are hard-coded in the objects and do not use system values or job attributes at run time.
2. Run time: Numeric input fields capable of having separators are treated according to the system value QDECFMT - meaning the thousand separator is stripped and alignment is performed for the decimal value.

Specifically what application providers want to see is that the decimal format used on the display file is left **unresolved** at compile time so that the display file object code can be used on any AS/400 system using its system value QDECFMT at run time for both input to the display file and output from the display file.

This can be overridden by the data area QWSDECFMT defined in APAR SA22846 and described in the following section.

Note: It is an accepted requirement to provide the same logic for output fields.

The DATE keyword used in DDS with EDTCDE(Y) uses the DATSEP separator from the job attributes.

10.5.3 Recommendations

If the application programs do not handle decimal formats by programming, the best method is to have two sets of DDS objects, one for decimal-point users and one for decimal-comma users. With the library list, supply a QWSDECFMT data area (provided through a PTF) that corresponds to the respective formatting requirement, when a single system is using multiple languages. When user profiles are created, ensure that attributes are set according to the user's location if the system supports multiple countries.

PTFs solving the problem:

```
Environment = V2R2  PTF = SF11141  Status = available
Environment = V3R1  PTF = SF22916  Status = available
Environment = V3R2  PTF = SF38496  Status = available
Environment = V3R6  PTF = SF26515  Status = available
Environment = V3R7  incorporated
```

Cover Letter from APAR SA22846:

The function provided in this PTF allows the user on the target system using a secondary language to provide the correct QDECFMT value so that input data editing is handled correctly.

The QDECFMT value for handling input data editing can be provided by creating data area QWSDECFMT in your secondary language library QSYSXXXX. If the data area is not created or is created but contains an invalid value, the system value is used.

We recommend placing the data area in the secondary language library. The system looks for the data area in the user's library list. Make sure that the data area only exists for users that need a value different from the system value.

The value must be a blank or uppercase J or I. Lowercase j and i are considered invalid values.

Creating or changing the data area does not have any effect on display files that are already open.

When using a blank value for input processing in the data area, do not use date fields for input in your display file having a period as a delimiter.

If you want to take advantage of this function, use one of the following commands:

```
CRTDTAARA DTAARA(QSYSXXXX/QWSDECFMT) TYPE(*CHAR) LEN(1)
VALUE(' ') AUT(*CHANGE)

CRTDTAARA DTAARA(QSYSXXXX/QWSDECFMT) TYPE(*CHAR) LEN(1)
VALUE('J') AUT(*CHANGE)
```

```
CRTDTAARA DTAARA(QSYSXXXX/QWSDECFMT) TYPE(*CHAR) LEN(1)
VALUE('I') AUT(*CHANGE)
```

10.6 Bidirectional Languages

The *bidirectional languages*,[1] are used mainly in the Middle East. They include Arabic, Urdu, Farsi, Hebrew, and Yiddish.[2]

In a bidirectional language, the general flow of text proceeds horizontally from right to left, but numbers are written from left to right, the same way as they are written in English. In addition, if some text (addresses, acronyms, or quotations) in English or another left-to-right language is embedded, it is also written from left to right.

10.6.1 Aspects of Bidirectional Language Writing Systems

This section discusses aspects of bidirectional texts related to directionality, shaping, and national numbers as well as keyboard input and compliance with Common User Access (CUA) guidelines.

10.6.1.1 Bidirectionality

In the context of bidirectionality, the following key notions are:

- Segments
- Global orientation
- Text-types and associated reordering methods
- Symmetrical swapping

These attributes are described in the following paragraph.

Directional Segments: A bidirectional text may consist of a main part that has one directionality (for example, an Arabic text written from right to left), and portions that have an opposite directionality (for example, an English address written from left to right). The portion of text with a different directionality is called a *segment*. A bidirectional text, thus, might have a body of right-to-left text with embedded left-to right segments. Sometimes a segment with one directionality might itself have embedded or "nested" within it an additional segment with an opposite directionality. It is conceptually possible to have many *levels of nesting*; in most cases, however, there are no more than two levels.

One level of nesting is necessary for the entry of numbers within Arabic or Hebrew text. In Hebrew, it is customary to write the name of the street before the number of the house, as shown in the following example:[3]

---

[1] The term **bidirectional languages** is used throughout this document in lieu of "languages with a bidirectional script." The latter is the correct form, but is more cumbersome.

[2] Arabic is spoken mainly in Algeria, Bahrain, Egypt, Iraq, Israel, Jordan, Kuwait, Lebanon, Libya, Morocco, Oman, Qatar, Saudi Arabia, Sudan, Syria, Tunisia, United Arab Emirates, and Yemen. Urdu is spoken mainly in Pakistan. Farsi is spoken mainly in Iran. Hebrew is spoken mainly in Israel, and Yiddish is spoken mainly in Israel, Europe, and North America.

[3] Since not every reader has a knowledge of Hebrew or Arabic writing, the following convention is used in the forthcoming examples to simulate bidirectional scripts: Hebrew and Arabic text is represented by lowercase English letters, while English text is represented by uppercase letters.

Example of Nesting

```
b ecnartne 25 teerts elpam
<--------- -> <-----------
```

The street name is entered from right to left. The flow then has to be reversed to allow correct entry of the number from left to right (this being the nested left-to-right segment). Then, again, the flow must be reversed to allow the entry of the entrance information from right to left.

Imagine someone writing a letter in English to someone who can read Hebrew also, and writing the address in Hebrew. In this case, the address in Hebrew is actually a nested segment of the English text.

```
MY ADDRESS IS b ecnartne 25 teerts elpam THIS MONTH.
------------><--------- -> <----------- ---------->
NEST LEVEL:00000000000000111111111112211111111111000000000000
```

Since the nested segment of the address has itself a nested segment (the street number), we end up with two levels of nesting.

Global Orientation: Bidirectional text may consist of mainly right-to-left text with some left-to-right nested segments (such as an Arabic text with some information in English), or vice versa (such as an English letter with a Hebrew address nested within it). The predominant direction, which we call the *global orientation*, cannot always be quickly deduced from the general context. Consider the following example :

Global Orientation Not Obvious from the Context Itself

```
FRED DOES NOT BELIEVE  taht yas syawla i
```

This sentence has one meaning when the reading is from left to right (Fred does not believe I always say that), and another meaning when read from right to left (I always say that Fred does not believe). In the first half of the preceding example, the global orientation of the text is left-to-right and in the second half it is right-to-left.

Since the global orientation is not always obvious from the context,[4] it has to be known to the application developer whose product is processing the bidirectional data.

Not to be confused with the global orientation of the text is the physical orientation of the presentation device. A display terminal has, for example, a right-to-left physical orientation if the first character on the window is the one in the upper right-hand corner and the general cursor movement is from right to left (and top to bottom).

[4] In some specific cases, one might choose to have a *contextual global orientation*, where the global orientation is set according to the directional characteristic of the first character in the data stream that has a distinct directionality.

Text-Types and Associated Reordering Methods

In a bidirectional text, we must clearly distinguish between the *physical order* in which the text is presented, and the *logical order* in which its segments are processed (or pronounced, if read aloud). Some segments may need to be reordered to a logical or physical order.

There are different approaches to how bidirectional text is to be reordered, and at present none can be said to be prevalent. The notion *text-type* is used to point to which approach is applicable for a specific text. The physical and logical order and the different text-types are discussed further in the following example.

Logical Order versus Physical Order

MY WIFE'S NAME IS ilin

The global orientation is left-to-right. The first letter in the text is M, followed by Y, and so forth. In the **physical order**, after the letters I and S comes the letter i of the segment containing my wife's name in Hebrew. Note, however, that my wife's name is pronounced "nili." In the **logical order**, the first letter of the name segment is thus the letter n, followed by i, l, and i.

Sometimes (for example, in online help), it is convenient to store the bidirectional text exactly as presented — that is, in the physical order. But if there is an intent to process the text (for example, to sort it), one must have the segments stored in their **logical order**. There is no meaning in the preceding example to sort the name "ilin"! It makes sense to reorder the text so the directional segment containing the name "nili" is inverted before being stored for further processing. The logical order is the preferred sequence for entering text and for processing. Conceptually, any storage device can be seen as storing the data from left to right. If one wants to perform straightforward processing on the stored text (sorting, collating, indexing) without the need to preprocess each segment, the bidirectional data must be stored in its logical sequence. This means reversing segments whose direction is opposite to the global orientation.

Text-Types and Reordering Techniques

Different text-types require different approaches to reordering:

- **Visual text-type**:

  The oldest approach, dating from the time when there was no processing capability at the workstation, is to simply copy the entire display to storage and vice versa (possibly inverting every row, depending on the physical orientation of the display). It is up to each application programmer to know where the embedded segments are located and to process them accordingly. This text-type is called *visual* because it is a replication of the presented form. Many legacy applications[5] and their files have this type of text.

- **Implicit text-type**:

[5] Legacy applications are those that have been "inherited" from a prior era. They may be obsolete, but must be supported.

In the *implicit* text-type, it is assumed that the letters of the Latin alphabet have a strong inherent left-to-right directionality, and those of the Arabic, Farsi, Urdu, and Hebrew alphabets have a strong right-to-left inherent directionality. An algorithm of implicit text processing recognizes segments based on their inherent directional characteristics, and segment inversion is performed automatically. The concept of an implicit algorithm is simple to understand. Its main limitation is that it cannot correctly handle some strings that have numbers and intermixed left-to-right and right-to-left letters.

- **Explicit text-type**:

  The *explicit* text-type assumes that there are additional control characters embedded in the text that instruct an *explicit* algorithm to perform segment inversions, shaping or numeral selections, and other transformations.

Thus, a text with visual text-type is stored in its physical order, and a text with an implicit text-type is stored in its logical order, which is better suited for automatic processing. A text with an explicit text-type is usually stored in logical order, but because of the embedded controls in the text, the automatic processing is not always straight forward.

The terms Visual-to-Logical and Logical-to-Visual conversion refer to the process of converting text strings between visual text-type and implicit text-type.

There is no one type of text that can be used in all cases. The implicit techniques are usually heuristic and, thus, have some limitations as noted previously. The explicit techniques, while alleviating the limitations of implicit techniques, introduce other limitations, such as the need for automatic processes to cope with embedded controls.

One specific technique, the basic display algorithm,[6] tries to be a bridge between the implicit and explicit techniques. In principle, it is an implicit reordering algorithm, but it can deal with a few specific directional controls embedded in the text.

There are applications and related databases for all three text-types. It is possible for bidirectional text that is presented one way to be stored in a different layout. One has only to know what text-type or reordering algorithm was used to correctly transform or process the bidirectional text.

Symmetrical Swapping

Some characters, such as the greater-than sign (>), have an implied directional meaning and have a complementary symmetric character with an opposite directional meaning (the less-than sign (<)). If it is used within a segment that is presented right-to-left but is inverted (left-to-right) when stored for processing, such a character might have to be replaced by its symmetric sibling to ensure that the correct meaning of the text is preserved. The replacement of such a character by its complement during transformation of a bidirectional text is called *symmetrical swapping*.

---

[6] The basic display algorithm was initially published in the Directionality Appendix A of “The Unicode Standard,” Version 1.0, volume 1, The Unicode Consortium, 1991.

Example of Symmetrical Swapping

On a right-to-left, the expression:

```
b < a
```

is read as **a** is **greater** than **b**. In storage, the orientation is always left-to-right; the first character in storage will thus be **a** followed by **<** and then **b**. So we end up having in storage:

```
a < b
```

which is, of course, incorrect. In this case, to preserve the correct meaning of the expression, the **<** character must be exchanged in storage with **>**.

Other graphic characters that require symmetrical swapping include the parentheses, square brackets, braces, and so on.

Although symmetrical swapping is a characteristic of bidirectional languages, it is not always mandatory for the software functions that transform different bidirectional-language text layouts. Sometimes this function is performed automatically by the workstation hardware or microcode.

Character Composition, Ligatures, and Diacritics

In complex-text languages, it might happen that there is not a one-to-one correspondence between the number of characters of text stored for processing and the number of characters of the presented text. Sometimes two or more characters might be represented by a single glyph occupying one presentation cell:

- *Ligatures* in the cursive languages use one glyph to represent two or more specific letters. For example, the ligature *Lamalef* is used to represent the frequently used pair of letters *Lam* and *Alef*.
- *Diacritics* (marks above, near, within, or below a consonant) are used in bidirectional languages, among other functions, to represent vowels. When kept in storage for processing, these marks occupy physical positions, but if used for representation, they might occupy the same cell as the associated consonants. As a compromise, given existing limitations (in the graphical capabilities and resolution of the display devices and the number of code points available), bidirectional languages such as Hebrew have in many implementations given up the ability to represent vowels by diacritics. The vowel sounds have to be surmised by readers based on their knowledge of the language and according to the semantics of the text. However, this guess work is not acceptable for specific applications such as poetry or processing of a classical text, which requires the use of diacritics.

  In Arabic, spacing diacritics are currently used as a compromise. In the present Arabic systems, some or all the Arabic diacritics are implemented as separate characters to be rendered following the character to which the diacritics belong.

10.6.1.2 National Numbers

In both Latin-based languages and Hebrew, numbers are represented using the so-called “Arabic” digits (1, 2, 3, 4, 5, 6, 7, 8, 9, 0). But the cursive languages (Arabic, Farsi, and Urdu) as well as many other complex-text languages,[7] have their own national glyphs for digits. The local name for numbers used in the cursive languages is not “Arabic numbers,” but rather *Hindi* or sometimes *Arabic-Indic* numbers. The direction of the numbers is always left-to-right. Mathematical formulas in Arabic are written from right to left and, in Farsi, they are written from left to right.

It is important to understand that, in most cases, the text stored for processing has numbers encoded in their Arabic (Western) code. When it comes to presentation, these numbers might be presented using either national glyphs for digits or ordinary Arabic digits, according to the intent of the user or application developer.

10.6.1.3 Bidirectional Data Entry Aspects

To those unfamiliar with bidirectional languages, understanding how segments of text with different directionality can actually be entered from a keyboard is somewhat of a puzzle. The following section explains how this is done.

Keying Order

The order in which bidirectional data is typed into a workstation is the order in which the text is meant to be read — that is, the logical order.

Bidirectional Keyboards

The keyboards used for bidirectional languages are similar to those used for English, but on the same keytops on which Latin characters and symbols are engraved, character symbols specific to the other language are added (In the case of the cursive languages, such as Arabic, the character symbols engraved are the basic characters only). Special key combinations are used to switch between the English keyboard layer and the national-language keyboard layer (For example, in some cases, the Hebrew layer is made active on a Hebrew keyboard by simultaneously pressing the Alt and Right Shift keys). Such key combinations are also used to enter appropriate input modes. For example, in some environments, *push mode*[8] is entered by simultaneously pressing Shift and Num Lock. Other combinations may be in use for that purpose in those environments that restrict the usage of the Num Lock key.

Bidirectional Typing Interfaces

To allow for bidirectional text entry from a keyboard, the interfaces must be able to intercept and process each keystroke. These interfaces can be part of the terminal and associated controller′s hardware or microcode, or they can be a specific routine that is added to the operating system.

---

7 Other complex-text languages that have their own national glyphs for decimal digits are Devanagari, Gurmukhi, Gujarati, Oriya, Bengali, Tamil, Telugu, Malayalam, Sinhalese, Khmer (Cambodian), Lao, Mongolian, Chinese, Tibetan, and Thai.

8 Push mode is a keyboard input mode in which characters are ″pushed″ in the direction opposite to the base direction of the segment and the cursor does not move in the same way the digits behave on the display of a pocket calculator.

Kinds of Bidirectional Typing Interfaces

There are two typing interfaces to consider:

- Manual typing method
- Automatic (logical) typing method

Manual Typing Method

In the manual typing method, the user ″tells″ the system in which direction the characters are to be typed. For mixed-direction typing, the user makes extensive use of the Push and End Push keyboard functions.

The manual method also supports an Automatic Push (Auto Push) mode. When the Auto Push setting is active, the Push Mode is started and terminated automatically, according to the actual characters being typed.

When the manual typing method is active, the keyboard language layer and cursor direction are handled separately by the system. This means that the user has separate control for:

- The direction of the field — controlled by the Field Reverse keyboard function
- The direction of the typed text — controlled by the Push and End Push keyboard functions
- The keyboard language layer — controlled by the keyboard language layer switching keys

Automatic (Logical) Method

This convention provides some ″automatic″ handling of directionality. When this method is active, the system determines the directionality of each part of the text (each segment) based on the actual characters being typed, using a set of predefined rules. The method is called ″logical″, because the direction of the text is ″logically″ deduced based on the language of the characters. With this method, the system automatically determines how to display characters in the ″correct″ order when the user switches keyboard language groups.

Another feature of this method is that it handles text in ″typing order″; that is, the system stores text in the typing order. It then uses a set of predefined rules to determine how the text is displayed, processed, and deleted by the application.

Logical Typing and Contextual Orientation: If the cursor is in the Home position (the first logical position in the field/window), and a character of a language (other than the default language of the current orientation) is entered, the display or window orientation is reversed automatically. That is, if the character entered is Hebrew, the window orientation is right-to-left; if the character is English, the window orientation is left-to-right.

10.6.2 Common User Access and Bidirectional Languages

The basic rule for applications that are to conform to Common User Access (CUA) guidelines is that ".. all pieces of data must be displayed in the orientation that is correct for the application user. Data input must be supported in the orientation that is natural for users."

Comprehensive guidelines on language usage, data entry, orientation of windows, right-to-left flavor of CUA-interface components, and so on, are

thoroughly covered in publications such as Appendix D of *Object-Oriented Interface Design: IBM Common User Access Guidelines* published by QUE, ISBN 1-56529-170-0, SC34-4399.

10.6.3 Conclusions and Guidelines

Though so different in their appearances, all the bidirectional languages such as Arabic, Farsi, Urdu, Hebrew, and Yiddish have a distinct common characteristic: the form of the rendered text is different from that of the stored text. The transformation functions needed to perform the changes between rendered and stored text depend on descriptive information pertaining to the attributes of complex-text languages: global orientation, text-type, symmetrical swapping, shaping, and national numbers.

Application developers should be aware of the fact that in the complex-text languages, there is a need for transformations between the different text formats. They should allow for user or system exits to facilitate invoking these transformations, in those places where a transformation might be expected (at input, before output, before a collating process, and so on). Programs must be able to identify the location and contents of the complex-text attributes and be able to change their content if needed.

Just as for any other language, an application meant to be used for complex-text languages should utilize the appropriate language code page and cultural data (date and time format, collating sequence, monetary format, and so on).

Application developers should design their products in such a way that they use, as much as possible, the standard functions and controls provided by the operating systems for these languages. They might choose to use the application program interfaces (APIs) offered in the national language versions of the operating systems to perform such transformations (when available).

It is a good practice to concentrate all of the national language-related functions or function calls in a definite program area for easy maintainability and change support.

10.7 DBCS Language Features for DBCS Environments

OS/400 has some national language features for DBCS environment. OS/400 can support plural languages with single OS, but it needs to setup some language features for primary and secondary languages. Especially, DBCS environment needs DBCS specific language feature for primary language. This section shows a list of DBCS language feature codes and recommended DBCS test environment. To retrieve a DBCS related document (Planning Guide for Developing DBCS/NLS Software Products written by IBM Yamato Lab) from IBM TOOLS disk, enter the following command on the VM:

```
TOOLS SENDTO KGNVM2 TOOLS NLSTOOLS GET PLNGDBCS PACKAGE
```

10.7.1 Language Feature Codes for DBCS

The primary language feature code is 29XX. All V3 and V4 OS/400s use the same feature code.

Table 23. Primary Language Feature Code for DBCS

Feature Code	Description
2938	English Uppercase DBCS
2984	English Uppercase and Lowercase DBCS
2962	Japanese
2986	Korean
2987	Traditional Chinese
2989	Simplified Chinese
Note: OS/400 level: V3R1, V3R2, V3R6, V3R7, and V4R1	

The secondary language feature code is 5XXX. V4R1 OS/400 uses 55XX, V3R6 and V3R7 OS/400 use 56XX, and V3R1 and V3R2 OS/400 use 57XX.

Table 24. Secondary Language Feature Code for DBCS (V4R1)

Feature Code	Description
5538	English Uppercase DBCS
5584	English Uppercase and Lowercase DBCS
5562	Japanese
5586	Korean
5587	Traditional Chinese
5589	Simplified Chinese
Note: OS/400 level: V4R1	

Table 25. Secondary Language Feature Code for DBCS (V3R6 and V3R7)

Feature Code	Description
5638	English Uppercase DBCS
5684	English Uppercase and Lowercase DBCS
5662	Japanese
5686	Korean
5687	Traditional Chinese
5689	Simplified Chinese
Note: OS/400 level: V3R6 and V3R7	

Table 26 (Page 1 of 2). Secondary Language Feature Code for DBCS (V3R1 and V3R2)

Feature Code	Description
5738	English Uppercase DBCS
5784	English Uppercase and Lowercase DBCS
5762	Japanese
5786	Korean

Table 26 (Page 2 of 2). Secondary Language Feature Code for DBCS (V3R1 and V3R2)	
Feature Code	**Description**
5787	Traditional Chinese
5789	Simplified Chinese
Note: OS/400 level: V3R1 and V3R2	

10.7.2 Recommended DBCS Test Environment

Basically, all DBCS language features have common DBCS enabling modules. The difference between DBCS feature codes is only MRI. However, several software programs have language specific modules for DBCS language features. These software programs must be tested with these language features. For example, Integrated PC Server applications should be tested with language specific features.

Table 27. Internationalized Software			
Country	**Primary**	**Secondary**	**QCCSID**
Japan	2984	5X62/5X38	5026 and 5035
Korea	2984	5X86	933
T-Chinese	2984	5X87	935
S-Chinese	2984	5X89	937

Table 28. Language Specific Software			
Country	**Primary**	**Secondary**	**QCCSID**
Japan	2962	5X84/5X38	5026 and 5035
Korea	2986	5X84	933
T-Chinese	2987	5X84	935
S-Chinese	2989	5X84	937
All DBCS	2984/2938		5026, 5035, 933, 935, 937

Appendix A. National Language Versions on AS/400 System

The following tables show national language version feature codes and some of the values associated with each national language version.

Table 29 (Page 1 of 2). National Language Versions on AS/400 System

National Language Version (NLV)	Feature Code	Release	Lang ID	EBCDIC CCSID	Key-board	CHRID	ASCII (*)
Albanian	2995	V4R1M0	SQI	500	ALI	697 500	850
Arabic	2954	V3R1M0	ARA	420	CLB	235 420	864
Belgian Dutch	2963		BGR	500	BLI	697 500	850
Belgian English	2909	V4R1M0	ENB	500	BLI	697 500	850
Belgian French	2966		FRB	500	BLI	697 500	850
Brazilian Portuguese	2980		PTB	37	BRB	697 37	850
Bulgarian	2974	V4R1M0	BGR	1025	BGB	1150 1025	855
Canadian French MNCS	2981		FRC	500	CAI	697 500	850
Croatian	2912	V3R1M0	HRV	870	YGI	959 870	852
Czech	2975	V3R1M0	CSY	870	CSB	959 870	852
Danish	2926		DAN	277	DMB	697 277	850
Dutch Netherlands	2923		NLD	37	NEB	697 37	850
English Uppercase	2950		ENP	37	USB	697 37	437
English Uppercase and Lowercase	2924		ENU	37	USB	697 37	437
English Uppercase DBCS (Japanese, no Lowercase)	2938		ENP	5026	JKB	1172 290	897
English Uppercase DBCS (Japanese, Upper/Lowercase)	2938		ENP	5035	JKB	1172 1027	897
English Uppercase and Lowercase DBCS (Traditional Chinese)	2984		CHT	937	TAB	1175 37	1114
English Uppercase and Lowercase DBCS (Simplified Chinese)	2984		CHS	935	RCB	1174 836	1115
English Uppercase and Lowercase DBCS (Korean)	2984		KOR	933	KOB	1173 833	1088
Estonian	2902	V4R1M0	EST	1122	ESB	1307 1122	922
Farsi	2998	V3R1M0	FAR	1097	IRB	1219 1097	1098
Finnish	2925		FIN	278	FNB	697 278	850
French	2928		FRA	297	FAB	697 297	850
French MNCS	2940		FRS	500	SFI	697 500	850
German	2929		DEU	273	AGB	697 273	850
German MNCS	2939		DES	500	AGI	697 500	850
Greek	2957		ELL	875	GNB	925 875	869

Table 29 (Page 2 of 2). National Language Versions on AS/400 System

National Language Version (NLV)	Feature Code	Release	Lang ID	EBCDIC CCSID	Key-board	CHRID	ASCII (*)
Hebrew	2961	V3R1M0	HEB	424	NCB	941 424	862
Hungarian	2976	V3R1M0	HUN	870	HNB	959 870	852
Icelandic	2958		ISL	871	ICB	697 871	850
Italian	2932		ITA	280	ITB	697 280	850
Italian MNCS	2942		ITS	500	ITI	697 500	850
Japanese (Katakana) DBCS	2962		JPN	5026	JKB	1172 290	897
Korean DBCS	2986		KOR	933	KOB	1173 833	1088
Latvian	2904	V4R1M0	LVA	1112	LVB	1305 1112	921
Lithuanian	2903	V4R1M0	LTU	1112	LTB	1305 1112	921
Macedonian	2913	V4R1M0	MKD	1025	MKB	1150 1025	855
Norwegian	2933		NON	277	NWB	697 277	850
Polish	2978	V3R1M0	PLK	870	PLB	959 870	852
Portuguese (**)	2922		PTG	37	PRB	697 37	850
Portuguese MNCS (**)	2996		PTG	500	PRI	697 500	850
Romanian	2992	V4R1M0	ROM	870	RMB	959 870	852
Russian	2979	V3R1M0	RUS	1025	RUB	1150 1025	866
Serbian Cyrillic	2914	V4R1M0	SRB	1025	SQB	1150 1025	855
Simplified Chinese DBCS	2989		CHS	935	RCB	1124 836	903
Slovakian	2994	V3R1M0	SKY	870	SKB	959 870	852
Slovenian	2911	V3R1M0	SLO	870	YGI	959 870	852
Spanish	2931		ESP	284	SPB	697 284	850
Swedish	2937		SVE	278	SWB	697 278	850
Thai	2972	V3R1M0	THA	9030	THB	1279 838	874
Traditional Chinese DBCS (ROC)	2987		CHT	937	TAB	1175 37	1114
Turkish	2956		TRK	1026	TRB	1152 1026	857

(*) Most commonly used PC code page. Actual used code page depends on PC setup.

(**) The language ID for Portuguese and Portuguese MNCS is the same, PTG. Customers using Portuguese MNCS with the PRI keyboard must ensure that the CCSID job attribute is set to 500.

Table 30 (Page 1 of 2). Languages without National Language Version

Language (country)	Lang ID	EBDDIC CCSID	Key-board	CHRID	ASCII (*)
Afrikaans (South Africa)	AFR	37	USB	697 37	437
Australian English (Australia)	ENA	37	USB	697 37	437
Byelorussian (Byelorussia)	BEL	1025	RUB	1150 1025	1131
Irish Gaelic (Ireland)	GAE	285	UKB	697 285	850
Lao (Lao People's Democratic Republic)	LAO	1132	LAB	1341 1132	1133

Table 30 (Page 2 of 2). Languages without National Language Version

Language (country)	Lang ID	EBDDIC CCSID	Key-board	CHRID	ASCII (*)
Serbian Latin (Serbia)	SRL	870	YGI	959 870	852
Spanish (Argentina)	ESP	284	SSB	697 284	850
UK English (United Kingdom)	ENG	285	UKB	697 285	850
Ukrainian (Ukraine)	UKR	1123	UAB	1326 1123	1125
Urdu (Pakistan)	URD	918	PKB	1160 918	868
Vietnamese (Vietnam)	VIE	1130	VNB	1136 1130	1258

Appendix B. Invariant Character Set

All characters in this character set have the same code point (hexadecimal value) in most code pages. The only exceptions are Turkish code pages 1026 and 905, where the ″ (quotation mark) is variant, Japanese code page 290, non-extended has no lowercase a-z, and Japanese code page 290, extended, which contains lowercase a-z, but are variant.

Table 31. Invariant Character 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
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			
+	<	=	>	%	&	*	″	′	(	)	,	_
-	.	/	:	;	?							

Appendix C. Language Groupings

We have chosen to group languages that use the same basic set of characters or have similar characteristics to explain data exchange and integrity. Languages within one group of languages can exchange data without losing the integrity of character data when proper support is used, even if the encoding of the characters are different.

When character data is exchanged between language groupings, only those characters that are common in the involved code pages can be exchanged without losing integrity. An example of characters that are common for all code pages are shown in Appendix B, “Invariant Character Set” on page 127.

Some language groupings have a suffix identifier, (x). Within languages in a group having the same suffix identifier, data can be exchanged without losing integrity.

Characters that are not common in the involved code pages are mapped to a unique or ambiguous code point in the receiving code page, depending on the conversion method chosen.

Latin-1

- Afrikaans
- Albanian
- Australian English
- Belgian French
- Belgian Dutch (Flemish)
- Brazilian Portuguese
- Canadian French
- Catalan
- Danish
- Dutch
- Finnish
- French
- German
- Icelandic
- Irish Gaelic
- Italian
- Latin American Spanish
- Norwegian
- Portuguese
- Spanish (Castillian)
- Swedish
- Swiss French
- Swiss German
- Swiss Italian
- UK English
- US English

Latin-2

- Croatian
- Czech
- Hungarian

- Polish
- Romanian
- Serbian Latin
- Slovakian
- Slovenian

Latin-4

- Estonian (a)
- Latvian (b)
- Lithuanian (b)

Latin-5

- Turkish

Cyrillic

- Bulgarian (a)
- Byelorussian (a)
- Macedonian (a)
- Russian (a)
- Serbian (a)
- Ukrainian (b)

Other

- Greek (a)
- Lao (b)
- Thai (c)
- Vietnamese (d)

Bidirectional

- Arabic (a)
- Farsi (Iran) (b)
- Hebrew (c)
- Urdu (Pakistan) (d)

DBCS

- Japanese (a)
- Korean (b)
- Simplified Chinese (c)
- Traditional Chinese (d)

Appendix D. Worksheet for Planning Primary Language Change

This appendix provides worksheets when you plan to change primary language.

Table 32. Planning Primary Language Change

Item	Description	Current	Target
1	OS/400 release		
2	Primary language		
3	Secondary language(s)		
4	System value QLANGID		
5	System value QCCSID		
6	System value QKBDTYPE		
7	System value QCHRID		
8	Assumed CCSID for database files		
9	Evaluated database CCSIDs		

Table 33. Planning Software Order

Task	Description	Action	Feature
1	Do you need to upgrade your current release to a newer release before you can change the current primary language?		
2	Are there any secondary languages you need on your target release?		_ _
3	Identify the secondary language you need to order for changing the primary language on you system.	**ORDER**	_ _

Table 34. Planning CCSID Change with CHGLIBCCS Command

Library	From CCSID	To CCSID

Appendix E. Program and Command CHGLIBCCS

This example program and command can help you automate the task of changing CCSIDs of your database files.

To enter the source statements for the CL program, create a source file (QCLSRC) in your library (ZCCSID) and add the member CHGLIBCCS.

Important Information

This example program has not been subjected to any formal testing. It is provided "AS-IS", this example should be used for reference only. Please refer to the Special Notices section at the back of this redbook for more information.

```
/*  CHGLIBCCS:     Program to change CCSID of physical files in     */
/*  one library or all user libraries, excl. IBM user libraries     */
             PGM        PARM(&LIB &FCCS &TCCS)
             DCL        VAR(&LIB) TYPE(*CHAR) LEN(10) /* Library +
                          Name or *ALL */
             DCL        VAR(&FCCS) TYPE(*DEC) LEN(5 0) /* From CCSID */
             DCL        VAR(&TCCS) TYPE(*DEC) LEN(5 0) /* To CCSID */
             DCL        VAR(&JCCS) TYPE(*DEC) LEN(5 0) /* Job CCSID */
             DCL        VAR(&CNT) TYPE(*DEC) LEN(7 0) /* Counter */
             DCL        VAR(&CNTX) TYPE(*CHAR) LEN(7) /* Counter */
             DCL        VAR(&CNE) TYPE(*DEC) LEN(7 0) /* Counter ERR*/
             DCL        VAR(&CNEX) TYPE(*CHAR) LEN(7) /* Counter ERR*/
             DCL        VAR(&CNTM) TYPE(*CHAR) LEN(65) /* Message */
             DCLF       FILE(ALLFILES) ALWVARLEN(*YES)
             RTVJOBA    CCSID(&JCCS)
/* TEST LIB EXIST  */
             IF         COND(&LIB *NE '*ALL') THEN(DO)
             CHKOBJ OBJ(QSYS/&LIB) OBJTYPE(*LIB)
             MONMSG     MSGID(CPF9801) EXEC(DO)
             SNDPGMMSG  MSG('Library not found') TOMSGQ(*TOPGMQ) +
                          MSGTYPE(*DIAG)
             GOTO ENDIT
             ENDDO
             ENDDO
/* TEST FROM CCSID  */
             CHGJOB CCSID(&FCCS)
             MONMSG     MSGID(CPF1854 CPD0085) EXEC(DO)
             SNDPGMMSG  MSG('From CCSID not valid') TOMSGQ(*TOPGMQ) +
                          MSGTYPE(*DIAG)
                          GOTO ENDIT
             ENDDO
/* TEST TO CCSID  */
             IF         COND(&FCCS *EQ &TCCS) THEN(GOTO CMDLBL(STUPID))
             CHGJOB CCSID(&TCCS)
             MONMSG     MSGID(CPF1854 CPD0085) EXEC(DO)
STUPID:      SNDPGMMSG  MSG('To CCSID not valid') TOMSGQ(*TOPGMQ) +
                          MSGTYPE(*DIAG)
                          GOTO ENDIT
             ENDDO
/* SET TO *HEX    */
```

```
             CHGJOB     CCSID(*HEX)
/* GET FILES    */
             IF         COND(&LIB *EQ '*ALL') THEN(DSPFD +
                          FILE(*ALLUSR/*ALL) TYPE(*ATR) +
                          OUTPUT(*OUTFILE) FILEATR(*PF) +
                          OUTFILE(QTEMP/ALLFILES))
             ELSE       CMD(DSPFD FILE(&LIB/*ALL) TYPE(*ATR) +
                          OUTPUT(*OUTFILE) FILEATR(*PF) +
                          OUTFILE(QTEMP/ALLFILES))

             OVRDBF     FILE(ALLFILES) TOFILE(QTEMP/ALLFILES)
GETIT:
             RCVF
             MONMSG     MSGID(CPF0864) EXEC(GOTO CMDLBL(ENDIT))
/* OMIT LIBRARIES STARTING WITH # OR Q  */
/* NOTE:  CHARACTER # MUST BE THE CHARACTER CORRESPONDING TO   */
/*        CODE POINT X'7B' IN USER'S CODE PAGE.                */
             IF         COND(%SST(&PHLIB 1 1) *EQ '#') THEN(GOTO +
                          CMDLBL(GETIT))
             IF         COND(%SST(&PHLIB 1 1) *EQ 'Q') THEN(GOTO +
                          CMDLBL(GETIT))
/* PROCESS ONLY THOSE MATCHING FROM CCSID */
             IF         COND(&FCCS *NE &PHCSID) THEN(GOTO +
                          CMDLBL(GETIT))
/* FOR *ALL WE GOT IT */
             IF         COND(&LIB *EQ '*ALL') THEN(GOTO CMDLBL(GOTIT))
/* OTHERWISE, CHECK THE LIBRARY  */
             IF         COND(&LIB *NE &PHLIB) THEN(GOTO CMDLBL(GETIT))
GOTIT:
             CHGPF      FILE(&PHLIB/&PHFILE) CCSID(&TCCS)
/* IF CHANGE NOT POSSIBLE, FIND CPD322D IN JOBLOG  */
             MONMSG     MSGID(CPF7304) EXEC(DO)
             CHGVAR     VAR(&CNE) VALUE(&CNE + 1)
             GOTO       CMDLBL(GETIT)
             ENDDO
/* COUNT THEM */
             CHGVAR     VAR(&CNT) VALUE(&CNT + 1)
             GOTO       CMDLBL(GETIT)
ENDIT:
             CHGVAR     VAR(&CNTX) VALUE(&CNT)
             CHGVAR     VAR(&CNEX) VALUE(&CNE)
             CHGVAR     VAR(&CNTM) VALUE(&CNTX *BCAT 'FILES CHANGED +
                          IN LIBRARY ' *BCAT &LIB)
             SNDPGMMSG  MSG(&CNTM) TOMSGQ(*TOPGMQ) MSGTYPE(*COMP)
             CHGVAR     VAR(&CNTM) VALUE(&CNEX *BCAT 'FILES ARE NOT +
                          CHANGED IN LIBRARY ' *BCAT &LIB)
             SNDPGMMSG  MSG(&CNTM) TOMSGQ(*TOPGMQ) MSGTYPE(*COMP)
/* BACK TO NORMAL */
             CHGJOB     CCSID(&JCCS)
             ENDPGM
```

To compile the program, you need library ZCCSID with the file (ALLFILES) that was created in Section 4.6, "Evaluate Database Information" on page 29 (step 2) in the library list.

To enter the source statements for the command, create a source file (QCSRC) in your library ZCCSID and add the member CHGLIBCCS.

```
/* CHGLIBCCS:  Command source for changing CCSID of files    +
               in one or all user libraries           */
     CMD PROMPT('Change CCSID of Database Files')
          PARM       KWD(LIB) TYPE(*NAME) LEN(10) SPCVAL((*ALL)) +
                       MIN(1) PROMPT('Library name or *ALL:')
          PARM       KWD(FCCS) TYPE(*DEC) LEN(5 0) RANGE(37 +
                       65535) MIN(1) PROMPT('From CCSID:')
          PARM       KWD(TCCS) TYPE(*DEC) LEN(5 0) RANGE(37 +
                       65535) MIN(1) PROMPT('To CCSID:')
```

E.1 The CHGLIBCCS Command

When the CL program and command is created, you have a command interface that can assist you in changing database files that have an incorrect CCSID tag.

```
                  Change CCSID of Database Files (CHGLIBCCS)

Type choices, press Enter.

Library name or *ALL:  . . . . .              Name, *ALL
From CCSID:  . . . . . . . . . .              37-65535
To CCSID:  . . . . . . . . . . .              37-65535




                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

The three parameters must all be specified

E.1.1 How Program CHGLIBCCS Works

If the library parameter specifies *ALL, all user libraries on the system, excluding all libraries starting with # or Q, are processed. Otherwise, only the specified library is processed.

The attributes of all physical files in the specified library (*ALLUSR if *ALL specified) are extracted and stored in a file (ALLFILES) in QTEMP.

The program reads all entries in ALLFILES in QTEMP and compares the actual file CCSID with the one supplied in the From CCSID parameter. If they match, the physical file is changed to reflect the CCSID specified in the To CCSID parameter.

Upon completion, messages are sent to the program message queue, for example:

```
0000121 FILES CHANGED IN LIBRARY NAV89
0000000 FILES ARE NOT CHANGED IN LIBRARY NAV89
```

The job log contains information about the files actually changed. You may want to save the job log to document the changes made.

If the number for not changed is greater than zero, you must review the job log for the actual cause, which is documented in message ID CPD322D.

Appendix F. Code Page Example

This example shows the EBCDIC code page 37. The hexadecimal (binary) value is found by the first digit combined with the second digit.

For example, the dollar ($) sign is located in hexadecimal '5B'.

HEX DIGITS 1ST → 2ND ↓	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	(SP) SP010000	& SM030000	- SP100000	ø LO610000	Ø LO620000	° SM190000	µ SM170000	^ SD150000	{ SM110000	} SM140000	\ SM070000	0 ND100000
-1	(RSP) SP300000	é LE110000	/ SP120000	É LE120000	a LA010000	j LJ010000	~ SD190000	£ SC020000	A LA020000	J LJ020000	÷ SA060000	1 ND010000
-2	â LA150000	ê LE150000	Â LA160000	Ê LE160000	b LB010000	k LK010000	s LS010000	¥ SC050000	B LB020000	K LK020000	S LS020000	2 ND020000
-3	ä LA170000	ë LE170000	Ä LA180000	Ë LE180000	c LC010000	l LL010000	t LT010000	• SD630000	C LC020000	L LL020000	T LT020000	3 ND030000
-4	à LA130000	è LE130000	À LA140000	È LE140000	d LD010000	m LM010000	u LU010000	© SM520000	D LD020000	M LM020000	U LU020000	4 ND040000
-5	á LA110000	í LI110000	Á LA120000	Í LI120000	e LE010000	n LN010000	v LV010000	§ SM240000	E LE020000	N LN020000	V LV020000	5 ND050000
-6	ã LA190000	î LI150000	Ã LA200000	Î LI160000	f LF010000	o LO010000	w LW010000	¶ SM250000	F LF020000	O LO020000	W LW020000	6 ND060000
-7	å LA270000	ï LI170000	Å LA280000	Ï LI180000	g LG010000	p LP010000	x LX010000	¼ NF040000	G LG020000	P LP020000	X LX020000	7 ND070000
-8	ç LC410000	ì LI130000	Ç LC420000	Ì LI140000	h LH010000	q LQ010000	y LY010000	½ NF010000	H LH020000	Q LQ020000	Y LY020000	8 ND080000
-9	ñ LN190000	ß LS610000	Ñ LN200000	` SD130000	i LI010000	r LR010000	z LZ010000	¾ NF050000	I LI020000	R LR020000	Z LZ020000	9 ND090000
-A	¢ SC040000	! SP020000	¦ SM650000	: SP130000	« SP170000	ª SM210000	¡ SP030000	[SM060000	(SHY) SP320000	1 ND011000	2 ND021000	3 ND031000
-B	. SP110000	$ SC030000	, SP080000	# SM010000	» SP180000	º SM200000	¿ SP160000	] SM080000	ô LO150000	û LU150000	Ô LO160000	Û LU160000
-C	< SA030000	* SM040000	% SM020000	@ SM050000	ð LD630000	æ LA510000	Ð LD620000	¯ SM150000	ö LO170000	ü LU170000	Ö LO180000	Ü LU180000
-D	(SP060000	) SP070000	_ SP090000	' SP050000	ý LY110000	¸ SD410000	Ý LY120000	¨ SD170000	ò LO130000	ù LU130000	Ò LO140000	Ù LU140000
-E	+ SA010000	; SP140000	> SA050000	= SA040000	þ LT630000	Æ LA520000	Þ LT640000	´ SD110000	ó LO110000	ú LU110000	Ó LO120000	Ú LU120000
-F	| SM130000	¬ SM660000	? SP150000	" SP040000	± SA020000	¤ SC010000	® SM530000	× SA070000	õ LO190000	ÿ LY170000	Õ LO200000	(EO)

Code Page 00037

Figure 18. Code Page 37

Appendix G. Special Notices

This publication is intended to help customers, BPs, and IBM technical specialists who implement national language environment on the AS/400 system. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/400. See the PUBLICATIONS section of the IBM Programming Announcement for V4R1 of OS/400, 5769-SS1 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of

including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AFP	AIX
Application System/400	AS/400
Client Access	Client Access/400
Common User Access	CUA
DB2	Distributed Relational Database Architecture
DRDA	IBM
InfoWindow	Integrated Language Environment
IPDS	OfficeVision
OfficeVision/400	OS/2
OS/400	Print Services Facility
PSF	S/370
S/390	Selectric
System/36	System/38
Ultimedia	400

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix H. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

H.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 143.

- *IBM AS/400 Printing IV*, GG24-4389

H.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

H.3 Other Publications

These publications are also relevant as further information sources:

- *CDRA Reference and Registry*, SC09-2190
- *International Application Development*, SC41-5603
- *National Language Support*, SC41-5101
- *National Language Design Guide*, GE09-8005
- *Work Management*, SC41-5306
- *DDS Reference*, SC41-5712
- *TCP/IP Configuration and Reference*, SC41-5420
- *Printer Device Programming*, SC41-5713
- *Distributed Database Programming*, SC41-5702
- *DB2 For AS/400 Query manager Use*, SC41-5212
- *DB2 For AS/400 Database Programming*, SC41-5701
- *Software Installation*, SC41-5120
- *Local Device Programming*, SC41-5121
- *Backup and Recovery*, SC41-5304
- *System API Reference*, SC41-5801
- *Client Access Host Servers*, SC41-5740

- *Client Access/400 for Windows 95/NT-Setup*, SC41-3512
- *Client Access/400 for Windows 3.1-Setup*, SC41-5534
- *Client Access/400 for OS/2 Setup*, SC41-3520
- *Client Access/400 Optimized for OS/2 Setup*, SC41-3510
- *Client Access/400 for DOS Setup*, SC41-3556
- *Client Access/400 for DOS with Extended Memory Setup*, SC41-3500
- *Workstation Customization Programming*, SC41-3605
- *About Type: IBM's Technical Reference for 240-Pel Digitized Type*, S544-3516
- *Microsoft Windows Resource Kit* by Microsoft Press, ISBN 1-55615-678-2

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO: type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Web Site on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.htm). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.htm). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

AFP	Advanced Function Printing
AFPDS	Advanced Function Printing Data Stream
ANSI	American National Standards Institute
APA	All Points Addressable
ASCII	American National Standard Code for Information Interchange
ASP	Auxiliary Storage Pool
CCSID	Coded Character Set Identifier
CDRA	Character Data Representation Architecture
CHRID	Character Identifier
CPGID	Code Page Global Identifier
CPI	Characters per Inch
DBCS	Double-Byte Coded Character Set
DCF	Document Composition Facility
DDS	Data Description Specifications
DLL	Dynamic Link Library
DRDA	Distributed Relational Database Architecture
EBCDIC	Extended Binary Coded Decimal Interchange Code
ECS	Electronic Customer Support
FGID	Font Global Identifier
FTP	File Transfer Protocol
GUI	Graphical User Interface
GML	Generalized Markup Language
HFS	Hierarchical File System
HPT	Host Print Transform
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IFS	Integrated File System
ILE	Integrated Language Environment
IPDS	Intelligent Printer Data Stream
ISO	International Organization for Standardization
ITSO	International Technical Support Organization
LPD	Line Printer Daemon
LPI	Lines per Inch
LPR	Line Printer Requester
MIME	Multipurpose Internet Mail Extensions
MNCS	Multinational Character Set
MRI	Machine Readable Information
NLS	National Language Support
NLV	National Language Version
ODBC	Open Database Connectivity
OEM	Original Equipment Manufacturer
OPM	Original Program Model
OS/400	Operating System for AS/400
PCL	Printer Control Language
PEL	Picture Element
POP	Post Office Protocol
PPDS	Page Printer Data Stream
PROFS	Professional Office System
PSP	Preventive Service Planning
SBCS	Single-Byte Coded Character Set
SCS	SNA Character String
SLIP	Serial Line Internet Protocol
SMTP	Simple Mail Transfer Protocol
SNA	Systems Network Architecture
TCP/IP	Transmission Control Protocol/Internet Protocol
TELNET	U.S. Dept. of Defense's Virtual Terminal Protocol, based on TCP/IP
UCS	Universal Character Set
UOM	Unit of Measure
URL	Universal Resource Locator
WSF	Workstation Function
WSG	Workstation Gateway

Index

Special Characters

*Common User Access guidelines 113
*SAVF 69

Numerics

3270 full-screen mode 70
5250 full-screen mode 70

A

abbreviations 147
acronyms 147
additional languages 7
AFP *YES 108
AFP *YES AND AFPATTACH *APPC 79
AFP Font Collection 108
algorithm 116
application code 104
Application Program Interfaces (APIs) 120
application providers 101
application updates 105
Arabic 113
ASCII 123
ASCII CCSID (default) 68
ASCII printers 110
ASCII-to-EBCDIC 74
attributes of complex-text languages 120
automatic (logical) typing method 119

B

basic characters only 118
basic display algorithm 116
bibliography 141
bidirectional data entry
  bidirectional keyboards 118
  bidirectional typing interfaces 118
  keying order 118
bidirectional language guidelines 120
bidirectional Languages 113
bidirectional text 113, 116
bidirectionality 113

C

CCSID 11, 13, 17, 19, 33, 42, 49, 51, 55, 83, 101, 102, 103, 104, 123
CCSID (coded character set identifier) 6
CCSID parameter 71
CCSID special value 16
CCSID tagging 69
CCSIDs 43
CDEFNT 83, 108
changing CCSIDs 15, 37
changing primary language 19, 42, 54
changing system values 20
character composition 117
character identifier (CHRID)
  by language group 86
  supported for each printer 86
  supported on the &sys. system 86
CHCP command 61
CHGLIBCCS command 133
CHGPF 105
CHRID 83, 107, 123
CHRID (character identifier)
  by language group 86
  supported for each printer 86
  supported on the &sys. system 86
CHRID *JOBCCSID 79
CNTRYID 11
code page 4, 137
  OS/2 61
  Windows 95 60
code page substitution 108
coded fonts 96
collating 115
command, CL
  copy from PC document (CPYFRMPCD) 65
  copy from stream file (CPYFRMSTMF) 65
  copy to PC document (CPYTOPCD) 65
  copy to stream file (CPYTOSTMF) 65
  display system value (DSPSYSVAL) 11
  DSPSYSVAL (Display System Value) 11
  send program temporary fix order (SNDPTFORD) 39
  SNDPTFORD (Send Program Temporary Fix Order) 39
  work with system values (WRKSYSVAL) 11
  WRKSYSVAL (Work with System Values) 11
Common User Access 119
Common User Access guidelines 119
complex-text attributes 120
complex-text languages 117, 118, 120
connection script 78
contextual orientation 119
control characters 116
copy from PC document (CPYFRMPCD) 65
copy from stream file (CPYFRMSTMF) 65
copy to PC document (CPYTOPCD) 65
copy to stream file (CPYTOSTMF) 65
CPI 82
CPX8416 12, 102, 103
CPYFRMPCD 65
CPYFRMSTMF 65

CPYTOPCD 65
CPYTOSTMF 65
CRTOUTQ command 98
CRTPRTF command 83
CUA 119
cultural convention
  date format and separator 3
  decimal format 3
cultural data 120
cultural job attribute value 11
cultural system value 10
cultural values 7
cursive languages 118
cursor movement 114
customizing of ASCII printer drivers 111

D

data entry 119
data queue APIs 63
database sharing 102
database support
  multi-system 13
  multilingual 13
DATFMT 11
DATSEP 11
DBCS environment 120
DBCS test environment 122
DDS CHRID 109
decimal and thousand separators 111
default CCSID 11, 14, 54, 55, 104
device
  file 111, 112
DEVTYPE 82
DFTCCSID 11
diacritics 117
direction of the field 119
direction of the typed text 119
directional controls 116
directional segments 113
directionality 113
display system value (DSPSYSVAL) command 11
displaying
  system value 11
distributed program call APIs 64
downloading fonts and code pages 110
DSPSYSVAL (Display System Value) command 11

E

EBCDIC (Extended Binary Coded Decimal Interchange Code) 4
EBCDIC CCSID 123
EBCDIC-to-ASCII 75
EBCDIC/ASCII mapping tables 99
ECS (electronical customer support) 39
edit
  code 111, 112
editing output and input fields 111
EDTCDE (Y) 112
electronical customer support (ECS) 39
embedded 113
embedded controls 116
empty database file 104
End Push 119
English keyboard layer 118
English uppercase and lowercase DBCS primary language 121
English uppercase DBCS primary language 121
evaluate database information 29
example of nesting 114
explicit algorithm 116
explicit text-type 116
externally described printer files 107

F

Farsi 113
  alphabet 116
FIDELITY 82
file transfer
  force conversion 57
file transfer function 52
FNTCHRSET 83, 108
FONT 82
font character sets 90
Font Global ID (FGID) 90
force conversion
  file transfer 57
  ODBC 58
FTP (File Transfer Protocol) 67

G

global orientation 113, 114, 115, 120

H

Hebrew 113
host code page 64
host print transform 97
host print transform (HPT) 109
host server CCSID 54
HTML (Hypertext Markup Language) 76, 77
HTTP (Hypertext Transfer Protocol) 67
HTTP server 77

I

impact of CCSIDs 101
implied directional meaning 116
importance of CCSID 13
indexing 115
information
  evaluate database 29
invariant character set 5, 127

inverted 116
INZPCS 50, 51

J

Japanese primary language 121

K

keyboard input 113
keyboard setup 65
Korean primary language 121

L

LANGID 11, 105
language grouping 7, 129
language ID 105
language sensitive setting 10
language usage 119
lay out your display 105
left-to-right 119
legacy applications 115
levels of nesting 113, 114
ligature Lamalef 117
ligatures 117
logical order 115, 116, 118
logical typing 119
logical typing method 119
LPI 82
LTYPE 67

M

machine readable information (MRI) 9
manual method 119
manual typing method 119
matrix IPDS printers 110
message file 111
message function 52
message TCP3C14 68
MFRTYPMDL parameter 110
Middle East 113
MIME 76
MIME (Multipurpose Internet Mail Extensions) 73
MRI (machine readable information) 9
MRI29xx 9
MSGCON keyword 107
multi-system database support 13
multilingual
  system 101
multilingual database support 13
multilingual single system 101
multinational mode 71

N

national character set 84
national glyphs for digits 118
national graphic character set 107
national language support (NLS) 9
national language version 123
national language versions 120
national mode 71
national numbers 113, 118, 120
national-language keyboard layer 118
negative numbers 111
nested 113
nested left-to-right segment 114
nested segment 114
network drives 62
network print 64
NLS (national language support) 9
NLS mapping tables 72
NLV 123
NLV (National Language Version) 7
numbers within Arabic or Hebrew text 113
numeral selections 116

O

ODBC
  force conversion 58
  Optimized OS/2 62
  Windows 3.1 61
  Windows 95/NT 62
opposite levels of nesting 113
Optimized OS/2
  ODBC 62
orientation of windows 119
OS/2
  code page 61
OVRFLW 82, 107
OVRPRTF 109

P

packed array 105
PAGESIZE 82, 107
panel setting on printer 110
PC5250
  setup 64
  special fonts 65
PCL4/5 97
physical order 115, 116
physical orientation of the presentation device 114
picture element (PEL) 91
planning
  software order 27
  target release 27
POP 72, 73
POP (Post Office Protocol) 67
PPDS 97
presented right-to-left 116
preventive service planning (PSP) 39
primary language 9

primary language feature code for DBCS
English uppercase and lowercase DBCS 121
English uppercase DBCS 121
Japanese 121
Korean 121
simplified Chinese 121
traditional Chinese 121
primary language feature for DBCS 121
primary language, changing 42
printer data streams 109
Printer file parameters 81
printer file support 80
printer function tables 98
printing environment of customer 109
program temporary fix order
sending 39
program-described printer files 107
PSF/2 91
PSF/400 109
PSP (preventive service planning) 39
push 119
push mode 118

Q

QCCSID 10, 13, 14, 21, 26, 44, 51, 79, 102, 122
QCHRID 10, 12, 21, 26, 27, 43, 79, 85
QCNTRYID 10, 13
QCONSOLE 35
QCURSYM 10, 12
QDATFMT 10, 12
QDATSEP 10, 12
QDECFMT 10, 12, 111, 112
QKBDTYPE 11, 21, 26, 27, 43
QLANGID 11, 13, 14, 21, 25, 27, 29, 43, 105
QLEAPADJ 11, 13
QSRTSEQ 11
QSYS29xx 9
QTIMSEP 11, 13
QWSDECFMT 112

R

re-ordering methods 113, 115
remote commands 64
remote SQL 52
reordering algorithm 116
reordering techniques 115
resident fonts 93
reversing segments 115
RPLUNPRT 82

S

save entire system 41
SBCS (Single-Byte Coded Character Set) 4
SCS printers 110
secondary language 9
secondary language feature for DBCS 121
segment 113
segment inversions 116
selecting CHRIDs 85
selecting code pages 95
selecting coded fonts 97
send program temporary fix order (SNDPTFORD)
command 39
sending
program temporary fix order 39
setup
keyboard 65
PC5250 64
SF99VRM 39
shaping 113, 120
shaping selections 116
shared folders function 52
simplified Chinese primary language 121
SLIP 77
SLIP (Serial Line Interface Protocol) 67
SMTP 72
SMTP (Simply Mail Transfer Protocol) 67
SNDPTFORD (Send Program Temporary Fix Order)
command 39
SNDTCPSPLF 110
software order
planning 27
sorting 115
source files 104, 106
spacing diacritics 117
special fonts for PC5250 65
specific character symbol 105
SRTSEQ 11
stand-alone code pages 94
standard functions and controls provided by the
operating systems 120
strong inherent left-to-right directionality 116
strong right-to-left inherent directionality 116
substituting code pages 95
substituting coded fonts 97
substituting font character set 92
support additional ASCII code pages 99
symmetric sibling 116
symmetrical swapping 113, 116, 117, 120
system CCSID support 16
system cross reference files 19, 20
system value
displaying 11
system values
CCSIDs 43
QCHRID 43
QKBDTYPE 43
QLANGID 43
working with 11

T

target release
  planning 27
TCP/IP LPD-LPR 98
TELNET 67, 70
text-type 113, 115, 116, 120
TIMSEP 11
traditional Chinese primary language 121
transform 116
transformations between the different text formats 120
TYPE 67
Typing interfaces
  Manual typing method 119
typing order 119

U

UCS-2 64
UOM 82
Urdu 113
  alphabet 116
user profile 11, 14, 15, 35, 51, 54, 55, 102

V

variant characters 104
verify CA/400 setting 43
verify keyboard 35
virtual printer function 52
visual text-type 115, 116
vowels 117
VTxxx full-screen mode 71

W

window orientation 119
Windows 3.1
  ODBC 61
Windows 95
  code page 60
Windows 95/NT
  ODBC 62
work with system values (WRKSYSVAL) command 11
working with
  system values 11
Workstation Function (WSF)
  5250 emulation 52
  printer emulation 53
writing systems 113
WRKSYSVAL (Work with System Values) command 11
WSG 76
WSG (Workstation Gateway) 67

Y

Yiddish 113

ITSO Redbook Evaluation

Speak the Right Language with Your AS/400 System
SG24-2154-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

_____

_____

_____

_____

_____

Printed in U.S.A.