



@server

iSeries

Globalization (Overview)







@server

iSeries

Globalization (Overview)



---

# Contents

|  |    |
|--|----|
| <b>OS/400 globalization</b> . . . . .          | 1  |
| What's new for V5R2 . . . . .                  | 2  |
| Print this topic . . . . .                     | 2  |
| Globalization overview . . . . .               | 3  |
| Multiple language support . . . . .            | 3  |
| What is a national language version? . . . . . | 3  |
| OS/400 translations . . . . .                  | 4  |
| National language design on OS/400. . . . .    | 5  |
| Linguistic and cultural values. . . . .        | 9  |
| Globalization checklists . . . . .             | 23 |



---

## OS/400 globalization

As companies integrate e-commerce on a global scale into their fundamental business processes, their prospective customers, established customers, and active partners can take advantage of increased revenue and decreased expenses through software globalization. They also can improve customer communications and increase savings. Globalized software gives you the following advantages:

- Increased customer satisfaction that can increase sales
- Enhanced customer support communications
- Enhanced global information dissemination
- A better return on Information Technology (IT) investments

This information shows you how to:

- Create an application efficiently and at minimal expense.
- Retrofit existing applications for globalization and create new applications designed for globalization. Designing an application for globalization, however, is usually less expensive than retrofitting an existing application.
- Ensure that the application design does not interfere with the current or planned design of other internationalized applications.

These pages provide a single source for the information you need to build applications for national and international audiences. You can also find information about what's new in this release and how you can print this topic.

### **Globalization overview**

This topic describes the way that globalization has been implemented on OS/400<sup>(R)</sup>, including topics that describe globalization-specific values on the system and topics that describe how services and functions in OS/400 support globalization.

### **Set up OS/400 with a national language version**

This topic describes the steps you need to take to properly install and configure a national language version on OS/400, with topics ranging from selecting and installing hardware, installing software, and configuring your environment to run in a globalized setting. You can use this information as you install your own servers, and you can apply the principles when you develop applications for customers who are installing their own national language version on OS/400.

### **Develop global applications**

This topic provides guidelines for designing, developing, and delivering globalized applications:

- Designing functions that are sensitive to national languages
- Supporting various types of hardware support
- Translating the textual data in your application
- Making your application available worldwide

### **Handle data in globalized applications**

This topic describes the ways in which OS/400 enables you to handle data in a globalized environment. Included in these pages are topics that describe Unicode and UCS-2 data, the Chinese standard GB18030, how to use CCSIDs to integrate multiple language environments consistently, and how to use bidirectional data, DBCS data, and locales.

### **Globalization reference information**

This topic provides detailed supporting information for the concepts and tasks discussed in the Globalization category.

### Globalization checklists

This topic gathers together all of the checklists that are contained with these pages. These checklists are useful reminders of the issues you need to consider as you create and work with global applications.

---

## What's new for V5R2

This release, the OS/400 Globalization topic has been reorganized for better online readability and accessibility.

Technical updates include the following:

- New support for International Components for Unicode (ICU).
- New system-supplied locales.
- New support for Unicode data.
- Support for GB18030, the new Chinese standard.
- Support for the QTQ\_DEFAULT\_CCSID system level environment variable.

### How to see what's new or changed

To help you see where technical changes have been made, this information uses:

- The



image to mark where new or changed information begins.

- The



image to mark where new or changed information ends.

To find other information about what's new or changed this release, see the Memo to Users



PDF.

---

## Print this topic

The Globalization topic is divided into smaller PDF documents that correspond to the major categories as they are presented in the iSeries<sup>(R)</sup> Information Center. The smaller size of each of the parts makes downloading and printing the Globalization information faster and easier. You can view or print the following topics:

- Globalization overview
- Set up OS/400 with an NLV
- Develop global applications
- Handle data in globalized applications
- Globalization reference information

If it is more convenient for you, you can also print the Globalization information as a single PDF.

You need Adobe Acrobat Reader to view or print these PDFs. You can download a copy from the Adobe



Web site ([www.adobe.com/prodindex/acrobat/readstep.html](http://www.adobe.com/prodindex/acrobat/readstep.html)).



**Note:** Some of the illustrations in the PDF format of the Globalization topic do not display in the same manner as they do in the HTML format. The HTML format shows the illustrations as they are best viewed.

---

## Globalization overview

OS/400 supports many languages; you can work in the language of your choice. OS/400 also ensures that the data you send to and receive from the server appears in the form and order you expect. By supporting many different languages, the operating system works as you expect it to work, from both a linguistic and a cultural point of view.

OS/400 uses a common set of program code, regardless of which language you use on the system. For example, the program code on a U.S. English server and the program code on a Spanish server are identical. Different sets of textual data are used, however, for different languages.

**Textual data** is a collective term for menus, displays, lists, prompts, options, online help information, and messages. This means that you see *Help* for the description of the function key for online help information on a U.S. English server, while you see *Ayuda* on a Spanish server. Using the same program code with different sets of textual data allows OS/400 to support more than one language on a single server.

The following topics introduce and explain globalization support on OS/400.

- Multiple language support
- What is a national language version?
- OS/400 translations
- National language design on OS/400
- Linguistic and cultural values

## Multiple language support

OS/400 provides the tools and function you need to make your applications deliver your business information, such as dates and numbers, in formats that conform to the expectations of users in multiple cultures using multiple languages. You can enable your iSeries server to translate, present, and process data in a global environment.

When you install secondary languages on your system, OS/400 also lets you set up your system with user interfaces (that is, textual data) for any of the national language versions provided for the system. To support multiple languages concurrently, you must have adequate storage to install all the necessary secondary languages. You must also install the necessary hardware to support each language.

## What is a national language version?

A national language version (NLV) is a version of the Operating System/400<sup>(R)</sup> (OS/400) that contains a predefined set of language-dependent values like date format, time format, sort sequence, and so forth for a particular language.

When you order an OS/400 licensed program, 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, you can order additional languages. For example, if you are a German customer, you may need support for both German and French on one server. 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 server, you designate one of the versions as the primary language. The primary language you designate is the feature code identified when you ordered the OS/400 program. You designate all other national language versions as secondary languages.



If you intend to use languages that require double-byte character set (DBCS) support, you must select a DBCS national language version as your primary language.



You must order some of the licensed programs for your server with the same language feature code as the primary language of the system. If the language feature code of a licensed program differs from the language feature code of the primary language of the server, the licensed program may not install correctly. Licensed programs with different feature codes as the primary language can be installed as a secondary language. See the National language version feature codes topic for a complete listing of national language versions.

For additional information about working with national language versions, see Set up an iSeries server with a national language version.

## OS/400 translations

OS/400, or portions of OS/400, are translated into the following languages. Not all portions of OS/400 are translated into every language.

- Albanian
- Arabic
- Brazilian Portuguese
- Bulgarian
- Croatian
- Czech
- Danish
- Dutch (this includes Belgian Dutch)
- English (this includes English uppercase, English uppercase DBCS, English uppercase and lowercase, English uppercase and lowercase DBCS), and Belgian English
- Farsi
- Estonian
- Finnish
- French (this includes Belgian, Canadian, and MNCS character set French)
- German (this includes MNCS German)
- Greek
- Hebrew
- Hungarian
- Icelandic
- Italian (this includes MNCS Italian)
- Japanese
- Korean
- Laotian
- Latvian
- Lithuanian
- Macedonian
- Norwegian
- Polish
- Portuguese (this includes MNCS Portuguese)
- Romanian

- Russian
- Serbian
- Simplified Chinese
- Slovakian
- Slovenian
- Spanish
- Swedish
- Thai
- Traditional Chinese
- Turkish
- Vietnamese

## National language design on OS/400

The national language design in OS/400 defines the functions your application software can use to support national languages. These functions include:

- Character representation
- Character processing
- Character presentation
- Globalization hardware support
- Character data translation
- Locales

### Character representation

Character representation in the server is controlled by following elements of the Character Data Representation Architecture (CDRA). CDRA identifies characters by encoding scheme identifier (ESid), character set, pairs of character sets and code pages (as needed), and additional coding-related information (as necessary). This identification is established by a system of tags. The tags are handled by OS/400 in a way that ensures character set integrity.

The overall objective of CDRA is to define a method of assigning and preserving the meaning of coded graphic characters through various stages of processing and interchanging.

### For more information

- Encoding scheme
- Conversion of character data
- CCSID values
- Character data integrity

**Encoding scheme:** The Character Data Representation Architecture (CDRA) system of tags uses an encoding scheme to specify:

- The coding space (number and allowable value of code points in a code page)
- Rules for sharing the coding space between control and graphic characters
- 



Rules related to specific options, such as the number bytes required for each character (single-byte, double-byte, or mixed-byte) permitted in that scheme



- Rules related to code extension techniques (if used)

The rules for encoding schemes are followed when code points are assigned to graphic characters in a particular code page. Some common encoding schemes are Extended Binary Coded Decimal Interchange Code (EBCDIC) and American Standard Code for Information Interchange (ASCII).

**Conversion of character data:** The Character Data Representation Architecture (CDRA) system of tags ensures you can convert character data in a predictable, repeatable way.



Conversion pertains to converting the code points assigned to one or more characters in one code page to their corresponding code points in another code page. The conversion may cause a single character to map to a sequence of characters, or a sequence of characters to map to a single character.



Conversion should not be equated to translating from one language to another.

### Conversion methods:

The following methods are used for conversion:

- *Round-trip conversion.* The integrity of all character data is maintained from the source coded character set identifier (CCSID) to the target CCSID and back to the source.

When performing a round-trip conversion, you may see incorrect representation of the characters displayed in the target CCSID. The integrity is preserved, however. When the characters are converted back to the source CCSID, they regain their original hexadecimal values and representation.

- *Enforced subset match conversion (substitution).* Characters that exist in both the source and target CCSID have their integrity maintained. Characters in the source CCSID but not in the target CCSID are replaced. Replaced values are also referred to as substitution characters. For EBCDIC encoding, these appear on most display stations as a solid block. For ASCII encoding, these substitution characters appear differently.

This substitution is permanent when converting back to the source CCSID because it is not possible to retrieve the original hexadecimal values.

For a list of CCSID conversions that result in substitution characters, see the Default conversion that may use substitution table.

- *Linguistic conversion.* Also known as best-fit conversion, a partial mapping is done from the source code page to the target code page. The integrity of characters that are in both the target CCSID and the source CCSID are preserved. Characters that are not in the target CCSID are mapped to the most culturally acceptable alternative for that character.

For example, the source CCSID may support an A grave (

À

). The target CCSID may not support this character. During the conversion, the most linguistically acceptable character (a Latin capital A) is substituted for the A grave. After the conversion, characters that are not included in the target CCSID are presented to the user as the most linguistically acceptable substitution characters. This substitution is permanent. Any loss of character integrity is permanent.

Through an application programming interface (API), linguistic conversion is available from any supported single-byte CCSID to any other supported single-byte CCSID.

**CCSID values:** CDRA defines the following range of values for CCSIDs:

| CCSID value         | Purpose or meaning                    |
|---------------------|---------------------------------------|
| 00000               | Use next higher hierarchical CCSID    |
| 00001 through 65533 | IBM <sup>(R)</sup> -registered CCSIDs |
| 65534               | Refer to lower hierarchical CCSID     |

|       |   |
|-------|---|
| 65535 | No automatic conversion of data between this CCSID and any other CCSID. (This is the default setting of the QCCSID system value.) |
|-------|---|

CDRA uses a tag field to hold a CCSID value to identify the meaning of coded graphic characters. The tag field may be in a data structure that is logically associated with the data object (explicit tagging), or it may be inherited from the tag field associated with the other objects within the operating system (implicit tagging).

**Character data integrity:** The Character Data Representation Architecture (CDRA) system of tags uses coded character set identifiers (CCSIDs) to maintain data integrity when character data is passed from system to system or from user to user. CCSIDs assign a value that uniquely identifies the coded graphic character representation used for character data.

### Data integrity is not maintained using CCSID 65535 across countries

The following table shows the meaning of maintaining data integrity. A database file created by a U.S. user contains a dollar sign and is read by a user in the United Kingdom and in Denmark. If the application does not assign CCSID tags that are associated with the data to the file, users see different characters.

| Country | Keyboard type | Code page | CCSID | Code point | Character |
|---------|---------------|-----------|-------|------------|-----------|
| U.S.    | USB           | 037       | 65535 | X'5B'      | \$        |
| U.K.    | UKB           | 285       | 65535 | X'5B'      | £         |
| Denmark | DMB           | 277       | 65535 | X'5B'      | Å         |

### Data integrity is maintained by using CCSID tags

If the application assigns a CCSID associated with the data to a file, the application can use OS/400 CCSID support to maintain the integrity of the data. When the file is created with CCSID 037, the user in the United Kingdom (job CCSID 285) and the user in Denmark (job CCSID 277) see the same character. Database management takes care of the mapping.

| Country | Keyboard type | Code page | CCSID | Code point | Character |
|---------|---------------|-----------|-------|------------|-----------|
| U.S.    | USB           | 037       | 00037 | X'5B'      | \$        |
| U.K.    | UKB           | 285       | 00285 | X'4A'      | \$        |
| Denmark | DMB           | 277       | 00277 | X'67'      | \$        |

CCSID support is particularly important when:

- Multiple national language versions, keyboards, and display stations are installed on OS/400.
- Multiple iSeries servers are sharing data between systems with different national language versions.
- The correct keyboard support for a language is not available when you want to encode data in another language.

### For more information

- For a list of the OS/400 CCSIDs, refer to Coded Character Set Identifiers (CCSIDs) in the Globalization reference topic
- For a complete list of the CDRA CCSIDs, see the *Character Data Representation Architecture - Level 2, Registry*

- For an overview of the architecture, refer to *Character Data Representation Architecture Executive Overview*
- For a complete discussion of the architecture, refer to the *Character Data Representation Architecture - Level 2 Reference*

### **Character processing**

Character processing on OS/400 is controlled by following coding rules and guidelines that ensure consistent processing of character data. These rules and guidelines cover tasks such as:

- Converting character data to all uppercase or to all lowercase data
- Folding data (substituting printable or displayable characters for those that cannot be printed or displayed on a particular device)
- Processing character data strings
- Classifying characters
- Naming objects
- Determining data, file, and field lengths

Some of these rules and guidelines are described in *Develop global applications*.

### **Character presentation**

Character presentation on OS/400 is controlled by coding rules and algorithms that ensure consistent presentation of character data. These rules and algorithms cover tasks such as:

- Shaping characters
- Truncating characters
- Handling substrings of character data

These rules and algorithms are described in detail in *Develop global applications*.

### **Globalization hardware support**

Hardware, in this context, means the physical keyboards, displays, printers, and controllers that make up an iSeries server. The extent to which this hardware supports national languages may impose limitations on the degree of support that you can provide with an application. You must refer to the reference manuals for non-IBM hardware to determine what limitations, if any, are imposed by that hardware.

### **Character data translation**

Translating is changing the meaning of character data from a set of concepts, ideas, and statements in one human language to a culturally similar meaning in another human language. You can follow rules as guidelines to ensure translation goes smoothly. A subset of these rules is provided in the *User interfaces* topic.

### **Locales**

A locale is an object that can determine how data is processed, printed, and displayed. Locales are made up of categories that define language, cultural data, and character sets. These combinations of language, cultural data, and character sets comprise a locale. The locale support is provided to supplement the job value options that OS/400 previously has provided.

Many locales are shipped with OS/400. In addition, locale definition source files are provided for locale customizations. A locale definition source file contains one or more categories that describe (or make up) a locale.

See the following topics to learn more about using locales in a globalized environment:

- *Install and enable locales*
- *Work with locales*

## Linguistic and cultural values

Linguistic and cultural conventions include any server values, attributes, or settings that can be altered to suit a country or language. Examples of linguistic and cultural conventions on the system include date formats and currency symbols.

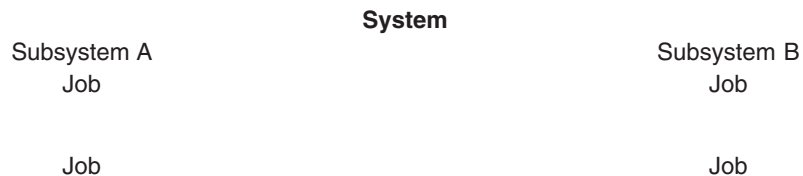
Some linguistic and cultural conventions may vary by language within a country. For example, language conventions vary in Canada. One set of linguistic conventions apply for French and another set of linguistic conventions apply for English.

The following topics describe specific values that affect your globalized server environment:

- User profiles
- Subsystems
- Job attributes
- System values
- Device descriptions
- Display and printer files
- Database files
- UIM menus and panel groups

### Where you can change linguistic and cultural values on OS/400:

Settings of cultural and linguistic conventions are supported at different levels in OS/400. The system is structured in the following way:



Some linguistic and cultural conventions can be set or changed at the system level, some at the subsystem level, some at the user profile level, and some at job run time. In addition, some cultural and linguistic settings can be set or changed in device descriptions. For example, keyboard types can be changed when creating or changing a display device description.

### For more information

For more information on linguistic and cultural conventions by country, see *National Language Design Guide: National Language Support Reference, Volume 2*.

### User profiles

Individual users can store customized cultural and linguistic values in their user profiles. These customized values can differ from the system default values and can be used by OS/400 when setting job attributes and object attributes for the individual user. Job attributes can also be used as defaults for setting object attributes created or changed under the control of that job.

If you have a single server supporting multiple languages, you should change the user profile to use language and culture-appropriate values. When you change the character set identifier (CCSID) parameter in the user profile, ensure the CCSID:

- Is set to an SBCS CCSID or to CCSID 65535 for SBCS users
- Is set to a mixed CCSID or to CCSID 65535 for DBCS users
- Is set to a SBCS CCSID for SBCS users on a DBCS system

You can use the Create User Profile (CRTUSRPRF) and the Change User Profile (CHGUSRPRF) commands to customize a user profile.

## Subsystems

A subsystem is a single, predefined operating environment through which the server coordinates the work flow and resource use. The server can contain several subsystems, all operating independently of each other. Subsystems manage resources. The run-time characteristics of a subsystem are defined in an object called a subsystem description.

You can use subsystems to support users in a multilingual environment. You should create a separate subsystem for each set of users with differing needs. For more information on subsystems, see the Work Management topic.

### Subsystem descriptions for secondary language users

You can create a subsystem description to be used in a multilingual environment. For example, you can create a subsystem for secondary language users (such as QGPL/GERMAN for German language users).

A subsystem description defines how, where, and how much work enters a subsystem, and which resources the subsystem uses to perform the work. An active subsystem takes on the simple name of the subsystem description.

A subsystem description consists of three parts:

- Subsystem attributes
- Workstation entry
- Routing entries

#### Notes:

1. You can work with existing work entries while the subsystem is active.
2. An IBM-supplied subsystem on a DBCS server is shipped with a workstation entry to support DBCS display devices.

For more information about subsystems, see the following:

- Start a subsystem

**Create a subsystem description:** IBM-supplied subsystem descriptions have been provided as examples and as backup for user-created subsystem descriptions. Therefore, we do not recommend modifying the subsystem descriptions in libraries QSYS and QGPL. You should make copies of the subsystem descriptions from these libraries and make changes to the copies.

You can create a subsystem description in two ways. You can copy an existing subsystem description and change it or you can create an entirely new description.

#### To copy an existing subsystem description:

1. On a command line, type CRTDUPOBJ, to create a duplicate object of an existing subsystem description.
2. Change the sign-on display file and the system part of the library list for the secondary language.

#### To create an entirely new subsystem description:

1. Create a subsystem description (CRTSBSD). Specify a sign-on file from the national language version library and specify the national language version library (QSYSnnnn) as the system-library list entry.
2. Create a job description (CRTJOBDD).
3. Add work entries to the subsystem description.



- a. ADDWSE (Add work station entry)
  - b. ADDJOBQE (Add job queue entry)
  - c. ADDCMNE (Add communications entry)
  - d. ADDAJE (Add autostart job entry)
  - e. ADDPJE (Add prestart job entry)
4. Create a class (CRTCLS).
  5. Add routing entries to the subsystem description (ADDRTGE).

**Subsystem attributes:** Subsystem attributes provide the overall characteristics of the subsystem. Attributes include the system-library list entry and a text description of the subsystem description.

For example, you can specify subsystem attributes to support secondary language users:

1. Specify the national language version for the subsystem library entry parameter.  
By creating a subsystem for each secondary language on your system, you can ensure that secondary language users have access to textual data in their own language. Within each subsystem, you can arrange the order of libraries in the library list so the textual data for the appropriate secondary language is at the top of the system library list. For example, if you have a primary language of Danish, and a secondary language of German, you can add a library at the top of the system library list in the German subsystem. Jobs running in the German subsystem then use the library at the top of the system part of the library list and a search for German textual data is successful.  
If you add a subsystem-library list entry for a national language version library:
  - Do not add the library to the QSYSLIBL system value.
  - Be sure that there are no more than 14 libraries in the QSYSLIBL list before adding your additional library entry. (The maximum number of list entries for the system part of the library is 15.)
2. Specify the sign on display using the national language version library.
3. Create or duplicate objects that all users of the secondary national language version need in the national language version library.
4. Add workstation entries for these workstations that are specifically configured for this national language version.

**Workstation entry:** You can specify the following items in a workstation entry. Parameter names are given in parentheses.

- Workstation name or type (WRKSTN or WRKSTNTYPE)
- Job description to be used for jobs started through this workstation entry
- Maximum number of interactive jobs that can be active at the same time through the entry (MAXACT)
- When the work stations are to be allocated, either when the subsystem is started or when an interactive job enters the subsystem through the Transfer Job (TFRJOB) command.

### Adding, changing, or removing workstation entries

The following commands allow you to add, change, or remove workstation entries from a subsystem description.

To add a workstation entry to a subsystem description, use the Add Work Station Entry (ADDWSE) command. The following is an example of adding a workstation entry:

```
ADDWSE SBSDB(USERLIB/ABC) WRKSTN(DSP12)
JOBDB(USERLIB/WSE)
```

To specify a different job description for a previously defined workstation entry, use the Change Work Station Entry (CHGWSE) command. The following is an example of changing a workstation entry:

```
CHGWSE SBSDB(USERLIB/ABC) WRKSTN(DSP12)
JOBDB(USERLIB/NEWJD)
```

To remove a workstation entry from a subsystem description, use the Remove Work Station Entry (RMVWSE) command. The following is an example of removing a workstation entry:

```
RMVWSE SBS(D(USERLIB/ABC) WRKSTN(DSP12)
```

**Start a subsystem:** Once you have created a subsystem that meets your needs, you need to start the subsystem. To start a subsystem, use the Start Subsystem (STRSBS) command:

```
STRSBS SBSD('library name/subsystem name')
```

For example:

```
STRSBS USERLIB/ABC
```

## Job attributes

Job attributes are set at the time a job starts. Some job attributes are set from the user profile. Other job attributes come from system values, from locales, from a Submit Job (SBMJOB) command, a job description, and



the Change Job (CHGJOB) command (from which you can change values for attributes while the job is running). The following attributes are especially useful for globalized environments:



- Coded character set identifier job attribute (CCSID)
- Job default coded character set identifier (DFTCCSID)
- Job library list

**Coded character set identifier job attribute:** When an interactive job is started, the job CCSID value is taken from the user profile. When a batch job is started, the current job CCSID is used unless a CCSID is specifically entered on the SBJOB command.

For every mixed-byte coded character set CCSID, there is a corresponding SBCS CCSID that is valid. If you specify a mixed-byte coded character set CCSID for an SBCS system, the job CCSID is changed to the corresponding SBCS CCSID.

If a job CCSID is specified as an SBCS CCSID, the job cannot handle DBCS data. If a job CCSID is specified as a mixed CCSID, the job can handle DBCS data. You must use a DBCS-capable display device, though, for the DBCS data in a job to display correctly. You can specify a mixed-byte CCSID for a job only if the DBCS system value (QIGC) value is set to 1 (on). A QIGC value of 1 indicates that a DBCS national

**Job default coded character set identifier (DFTCCSID):** A job attribute, job default CCSID (DFTCCSID), is created for jobs with a CCSID of 65535. The DFTCCSID value is used by some system code when a CCSID other than 65535 is needed.

The DFTCCSID attribute can only be retrieved or displayed. The value of this attribute is determined as follows:

- If the job CCSID is not 65535, the DFTCCSID equals the job CCSID.
- If the job CCSID is 65535, the DFTCCSID value is based on an appropriate value derived from the job language identifier (LANGID).

Once the job is running, the system determines the default CCSID for a job using the following logic (you can find the corresponding CCSID for LANGID in default CCSID table):

1. If the job CCSID is set to a value, it uses that value.
2. If the job CCSID is set to \*USRPRF, then the system checks the user profile for the value.

3. If the user profile is set to a value, it uses that value.
4. If the user profile is set to \*SYSVAL, the system checks the system value.
5. If the system value for QCCSID is set to a value, it uses that value.
6. If the system value is set to 65535, the system checks the job's language ID.
- 7.



If the job's LANGID is set to a value, the QTQ\_DEFAULT\_CCSID environment variable is checked for that LANGID value. If the QTQ\_DEFAULT\_CCSID environment variable contains a value for that LANGID, the CCSID specified in the QTQ\_DEFAULT\_CCSID environment variable is used. If the QTQ\_DEFAULT\_CCSID environment variable does not contain a value for the LANGID, the system converts that LANGID to a CCSID.



8. If the job's LANGID is set to \*USRPRF, the system checks the user profile's language ID.
- 9.



If the user profile's LANGID is set to a value, the QTQ\_DEFAULT\_CCSID environment variable is checked for that LANGID value. If the QTQ\_DEFAULT\_CCSID environment variable contains a value for that LANGID, the CCSID specified in the QTQ\_DEFAULT\_CCSID environment variable is used. If the QTQ\_DEFAULT\_CCSID environment variable does not contain a value for the LANGID, the system converts that LANGID to a CCSID.



- 10.



If the user profile's LANGID is set to \*SYSVAL, the QTQ\_DEFAULT\_CCSID environment variable is checked for that LANGID value. If the QTQ\_DEFAULT\_CCSID environment variable contains a value for that LANGID, the CCSID specified in the QTQ\_DEFAULT\_CCSID environment variable is used. If the QTQ\_DEFAULT\_CCSID environment variable does not contain a value for the LANGID, the system converts that LANGID to a CCSID.



Language identifiers and associated default CCSIDs contains a list of language identifiers and the DFTCCSID values associated

**Job library list:** The language used for textual data (displays, messages, printed output, and online help information) is controlled by the library list for the job.

Users can place their national language library before QSYS (the primary language library) and before any other national language libraries in their library lists. In this way, users can customize which national language versions of information are presented to them.

For more information on the national language libraries and proper authority to change these lists, see System Library List (QSYSLIBL) system value.

## System values

The system values of the primary language on the server are used as server-wide cultural and linguistic defaults. Therefore, if you change the primary language on the server, each varying system value resets to the default system value for the new primary language.

The following list shows the cultural and linguistic system values. To display or change these values, use the Work with System Value (WRKSYSVAL) command. A subset of language-dependent default system values (QCCSID, QCHRID, QCNTYID, QCURSYM, QDATFMT, QDATSEP, QDECFMT, QKBDTYPE, QLANGID, and QTIMSEP) are shown in Default system values in the Reference section.

- Century (QCENTURY)
- Character identifier (QCHRID)
- Character identifier control system (QCHRIDCTL)
- Coded character set identifier (QCCSID)
- Coordinated universal time offset (QUTCOFFSET)
- Country or region identifier (QCNTYID)
- Currency symbol (QCURSYM)
- System date (QDATE)
- Date format (QDATFMT)
- Date separator (QDATSEP)
- Day of month (QDAY)
- Day of week (QDAYOFWEEK)
- DBCS system indicator (QIGC)
- DBCS font name (QIGCCDEFNT)
- Decimal format (QDECFMT)
- Language identifier (QLANGID)
- Language indicator for keyboards (QKBDTYPE)
- Leap-year adjustment (QLEAPADJ)
- Locale (QLOCALE)
- Month of the Year (QMONTH)
- Set job attributes (QSETJOBATR)
- Sort sequence (QSRTSEQ)
- System library list (QSYSLIB)
- Time separator (QTIMSEP)
- Year (QYEAR)

### For more information

See System values in the Work Management topic for more information about all system values.

**Century (QCENTURY) system value:** The century (QCENTURY) system value is used to specify the century. It is used with the system values QDATE and QYEAR to determine the specific date currently being used by the server. The possible values are:

- 0 (the years from 1928 to 1999)
- 1 (the years from 2000 to 2053)

**Note:** 1900 to 1927 and 2054 to 2099 are not supported years for system time. Applications can, however, support year date ranges from



0001 to 9999.



You can set the value of QCENTURY with the century indicator, or the system sets the value of QCENTURY based on the following two situations:

- At the time of the first IPL, the system sets the initial value of QCENTURY based on the following rules:
  - If QYEAR is equal to or greater than 40, the system assigns a value of 0 to QCENTURY.
  - If QYEAR is less than 40, the system assigns a value of 1 to QCENTURY.
- When QYEAR or the year in QDATE is changed:
  - QCENTURY is set to 0 if QYEAR is 54 to 99
  - QCENTURY is set to 1 if QYEAR is 00 to 27

For example, if you change QYEAR from 95 to 13, the system changes QCENTURY from 0 to 1, indicating a year of 2013. However, if you change QYEAR from 95 to 45, the system will not change QCENTURY, because both 1945 and 2045 are valid dates.

If you change this value, the change takes effect immediately. Changing this value also affects the system value QDATE.

**Note:** The 21st century begins at 0000 hours, 1 January 2001. However, for purposes of common understanding, we are defining the 20th/21st century boundary to be between 2400 hours, 31 December 1999 and 0000 hours, 1 January 2000. This allows a discussion of the 21st century to include all dates with a 20xx format inclusive of the year 2000.

**Character identifier (QCHRID) system value:** The character identifier (QCHRID) is used to specify the character set and code page CHRID(\*SYSVAL) for the CL commands that create, change, or override display files, display device descriptions, user interface (UIM) menus, panel groups, and printer files. You can change this value if the system QCCSID system value is set to CCSID 65535. You can also change the QCHRID value if the code page portion of the new QCHRID value is the same as the code page portion of the QCCSID value.

**Character identifier (QCHRIDCTL) control system value:** The character identifier control (QCHRIDCTL) system value controls the type of CCSID conversion that occurs for display files, printer files, and panel groups. You must specify the \*CHRIDCTL special value on the CHRID parameter on the create, change, or override command for display files, printer files, and panel groups before this attribute will be used.

Possible values are:

**\*DEVD**

The support provided by the \*DEVD special value on the CHRID parameter for display files, printer files, and panel groups.

**\*JOBCCSID**

The support provided by the \*JOBCCSID special value on the CHRID parameter for display file, printer files, and panel groups.

**Coded character set identifier (QCCSID) system value:** The coded character set identifier (QCCSID) is used to specify the CCSID for OS/400. As shipped, and at installation, the CCSID is set to CCSID 65535.

You can change the coded character set identifier (QCCSID) system value. When you change this value, the default character set and code page system value (QCHRID) is changed to match the character set and code page of the coded character set identifier.

On a system capable of using DBCS data, QCCSID must be set to a mixed-byte coded character set CCSID (such as 05026) or to CCSID 65535. On a system that is not capable of using DBCS data, QCCSID must be set to an SBCS CCSID or to CCSID 65535. The QIGC system value indicates whether a system is capable of using DBCS data.

**Coordinated universal time offset (QUTCOFFSET) system value:** The coordinated universal time offset (QUTCOFFSET) is used to show the number of degrees, in hours and minutes, by which your local system differs from the zero meridian. This value is used by the system when processing alerts that are sent to other systems, as well as by other parts of the system. If systems in a network cross time zones, the QUTCOFFSET value is sent in the alert.

This value is 5 characters long. The first character is a plus (+) sign or minus (-) sign. The next 2 characters specify hours ranging from 00 through 24. The last two characters specify minutes ranging from 00 through 59.

For example, you have a network with one system in Brisbane, Queensland, Australia (Eastern Australia standard time zone) and one system in Caracas, Venezuela. You would set QUTCOFFSET to +1000 for the Brisbane system and to -0400 for the Caracas system.

The Brisbane system value should be changed each time the daylight savings time begins or ends. Caracas, Venezuela does not observe a daylight savings time, and its system value remains constant.

If you change this value, the change takes effect immediately.

**Country or region identifier (QCNTYID) system value:** The country or region identifier (QCNTYID) is used to indicate the default country or region identifier for the system. Setting this system value, along with the QLANGID system value, allows you to choose the correct language dictionary, encoding of data, and advanced linguistics for successful document indexing. There is no validity checking between the QCNTYID system value and the QCCSID system value.

See Language and country/region identifiers for a list of country and region identifiers.

**Currency symbol (QCURSYM) system value:** The currency symbol (QCURSYM) is used to verify the currency symbols specified in the DDS keywords Edit Word (EDTWRD) and Edit Code (EDTCDE). You can change the currency symbol to correctly reflect the monetary symbol used in your country or location. If you change this system value, the change takes effect immediately.

**System date (QDATE) system value:** The system date (QDATE) is used to indicate the year, the month, and the day on the system. This value is made up of the QYEAR, QMONTH, and QDAY system values. The format in which QDATE appears is specified by the QDATFMT system value. You can change the system date. If you change QDATE, the change may affect the system values for QCENTURY, QYEAR, QMONTH, QDAY, and QDAYOFWEEK. Any change you make to QDATE takes effect immediately.

**Date format (QDATFMT) system value:** The date format (QDATFMT) is used for the default value for the DATFMT job attribute. QDATFMT is also used to determine the format in which a date can be specified on the IPL options prompt.

This system value can be:

- YMD (year, month, day)
- MDY (month, day, year)
- DMY (day, month, year)
- JUL (Julian format, which is year, day of year)

You can change the date format to reflect the format in which months, days, and years are represented in your country or location. If you change this system value, the change takes effect for new jobs that enter the system after you make the change.

**Date separator (QDATSEP) system value:** The date separator (QDATSEP) is used as the date separator for the default value of the DATSEP job attribute. QDATSEP is also used as the date separator you can specify on the IPL options prompt.

You can change the date separator to reflect the character used to separate days, months, and years for your country or location. You can change the date separator to any one of the following values:

- A slash (/) as a date separator
- A hyphen (-) as a date separator
- A period (.) as a date separator
- A comma (,) as a date separator
- A blank ( ) as a date separator

If you change this value, the change takes effect for new jobs that enter the system after you make the change.

**Day of the month (QDAY) system value:** The day of the month (QDAY) is used to indicate the day of the month on the system. This value must be a valid day of month or a valid day of year (if you are using Julian date format).

You can change the day of the month to reflect the current day of the month in your country or location. If you change QDAY, you also change the value for QDATE. A change to this value takes place immediately.

**Day of week (QDAYOFWEEK) system value:** The day of week (QDAYOFWEEK) system value specifies the day of the week on the system. This value can be:

- \*SUN (Sunday)
- \*MON (Monday)
- \*TUE (Tuesday)
- \*WED (Wednesday)
- \*THU (Thursday)
- \*FRI (Friday)
- \*SAT (Saturday)

This value cannot be changed. It is set by the system. The value of QDATE determines the value of QDAYOFWEEK.

This value may not be set correctly if your system is not using the Gregorian calendar.

See QLEAPADJ for additional information that could affect the QDAYOFWEEK system value.

**DBCS system indicator (QIGC) system value:** The DBCS system indicator (QIGC) is used to specify whether a DBCS national language version is installed. This value is set when the primary national language version is installed.

If QIGC is set to 0, no DBCS national language version is installed on the system. When QIGC is set to 0, the coded character set system identifier (QCCSID) must be set to an SBCS coded character set identifier.

If QIGC is set to 1, a DBCS national language version is installed as the primary language on the system. When QIGC is set to 1, the coded character set system identifier (QCCSID) system value should be set to a mixed CCSID (such as 05026) or to CCSID 65535.

You cannot change this value.

**DBCS font name (QIGCCDEFNT) system value:** The DBCS font name (QIGCCDEFNT) is used when transforming SNA character string (SCS) data into an Advanced Function Printing<sup>(R)</sup> data stream (AFPDS) spooled file with shift in/shift out (SI/SO) characters present in the data.

QIGCCDEFNT is a 20-character list of up to 2 values. The first 10 characters contain the font name. The last 10 characters contain the library name. The font name can be only 8 characters. The possible values for the DBCS font name are:

**\*NONE**

No font is identified to the system.

**Coded font name**

The name of the DBCS font.

The possible values for the library are:

**\*LIBL** The library list is used to locate the font.

**\*CURLIB**

The current library is used to locate the font. If no library is specified, library QGPL is used.

**Library name**

The library containing the font.

**Decimal format (QDECFMT) system value:** This decimal format (QDECFMT) is used to do the following:

- Determines the type of zero suppression and decimal point character used by DDS edit codes 1 through 4 and A through M
- Determines the decimal point character for decimal input fields on displays

You can change the decimal format to reflect the way decimals are formatted for your country or location. You can change the decimal format to any one of the following values:

**(blank)**

If you specify a blank, the system uses a period for a decimal point, a comma for a 3-digit grouping character, and zero suppression to the left of the decimal point. For example,

One thousand is formatted as 1,000  
and  
Four one-hundredths is formatted as .04

**J**

If you specify a J, the system uses a comma for a decimal point, a period for a 3-digit grouping character, and zero suppression at the second character to the left of the decimal point. For example,

One thousand is formatted as 1.000  
and  
Four one-hundredths is formatted as 0,04

**I**

If you specify an I, the system uses a comma for a decimal point, a period for a 3-digit grouping character, and zero suppression to the left of the decimal point. For example,

One thousand is formatted as 1.000  
and  
Four one-hundredths is formatted as ,04

A change to this value takes place immediately.

**Language identifier (QLANGID) system value:** The language identifier (QLANGID) is used to specify the default language identifier for the server. This value also determines the sort sequence table to be used for sorting character data when the QSRTSEQ system value is set to \*LANGIDSHR or \*LANGIDUNQ.

**Note:** This value is not used to determine the sort sequence table when QSRTSEQ is set either to \*HEX or to a user-specified table.

You can change this system value to reflect the default language identifier for your country or location.



There is no validity checking between the QLANGID system value and the QCCSID system value.

**Language indicator for keyboards (QKBDTYPE) system value:** The language indicator for keyboards (QKBDTYPE) is used to specify the language character set for the keyboard. This value is used as the default keyboard type when you create a display device description. See National language keyboard types and SBCS code pages for a list of language indicators for keyboards.

You can change this value to reflect the language of your keyboard.

**Leap-year adjustment (QLEAPADJ) system value:** The leap-year adjustment (QLEAPADJ) is used to adjust the system algorithms for the leap year in different calendar systems. If your calendar year agrees with what is used in the Gregorian calendar, this system value should be zero. If your calendar year differs from the Gregorian, you may need to adjust the system calendar algorithm to account for the leap year you are using. To make the adjustment, divide the leap year in your calendar system by 4. Then set QLEAPADJ to the value of the remainder.

For example, the Gregorian calendar year of 1988 was the year 77 in the Taiwan calendar. Because 77 was a leap year for Taiwan, you need to divide 77 by 4. This leaves a remainder of 1. Therefore, to adjust the system calendar algorithm for Taiwan, specify a 1 for the QLEAPADJ value.

If you change QLEAPADJ, you do not change the system clock or the job dates of active jobs. Changing QLEAPADJ may result, however, in an implicit change to the QDATE system value.

**Note:** Not all system functions, such as DB date and timestamp fields, support QLEAPADJ when working with leap years.

**Locale (QLOCALE) system value:** The locale (QLOCALE) system value is used to specify a locale object that can determine how data is processed, printed, and displayed. Locales can define the language used by the system, cultural data of that language, and the type of characters displayed or printed.

The locale path name must be a path name that specifies a locale. A locale is made up of the language, territory, and code set combination used to identify a set of language conventions. The maximum path length allowed for the locale path name on the Change System Value (CHGSYSVAL) command is 1024 bytes.

The allowed values are:

| Value     | Indication                                       |
|-----------|--|
| *NONE:    | There is no locale for the QLOCALE system value. |
| *C:       | The C locale is to be used.                      |
| *POSIX:   | The POSIX locale is to be used.                  |
| ▶▶        | The path name of the locale to be used.          |
| path-name | ◀◀   |

**Month of the year (QMONTH) system value:** The month of the year (QMONTH) is used to indicate the month of the year on the server. This value must be a number from 1 (January) through 12 (December) if your system date format uses the Gregorian calendar. This value cannot be displayed or changed if your server date format uses the Julian format (year, day of year).

You can change the month to reflect the current month in your country or location. If you change QMONTH, you also change the value for QDATE. A change to this value takes place immediately.

**Set job attributes (QSETJOBATR) system value:** The set job attributes (QSETJOBATR) is used to set job attributes at job startup time. This system value has the following attributes that can be assigned values:

- Coded character set identifier (CCSID)
- Date format (DATFMT)
- Date separator (DATSEP)
- Decimal format (DECFMT)
- Sort sequence (SRTSEQ)
- Time separator (TIMSEP)

The system sets the initial values for these attributes from the locale (QLOCALE) system value.

**Sort sequence (QSRTSEQ) system value:** The sort sequence (QSRTSEQ), along with QLANGID, is used to determine the sort sequence table to be used for sorting character data. You can change QSRTSEQ to any one of the following values:

| Value                              | Meaning   |
|------------------------------------|---|
| *HEX                               | No sort sequence table is used. The hexadecimal values of the graphic characters are used to determine the sort sequence (a binary sort). This is the only sort sequence available for DBCS data.<br><br><b>Note:</b> When you specify values other than *HEX for mixed-byte character data, SBCS character data is sorted according to the sort sequence specified. DBCS character data is sorted by hexadecimal values (binary sort). |
| *LANGIDSHR                         | The sort sequence table may use the same weight for multiple graphic characters. The shared-weight sort table associated with the language specified in the LANGID parameter is used. This sort applies only to SBCS data.  |
| *LANGIDUNQ                         | The sort sequence table contains uniquely weighted graphic characters. The unique-weight sort table associated with the language specified in the LANGID parameter is used. This sort applies only to SBCS data.  |
| Qualified sort sequence table name | The name and library of the sort sequence table to be used. This value allows you to specify a sort sequence table other than those associated with the language specified in the LANGID parameter.<br><br>➤<br><br>This sort sequence table can be used to sort Unicode and SBCS data.<br><br>⬅  |

See Sort sequence tables for more information about sort sequence tables.

**System library list (QSYSLIBL) system value:** The system library list (QSYSLIBL) is used as the first part of the library list associated with a job. The libraries in the system part of the library list of a job are searched before any other libraries in the library list of a job. The list can contain as many as 15 names. You cannot delete or rename a library specified as part of the system library list, because libraries in this library list are locked.

You can change the system library list (QSYSLIBL). If you change QSYSLIBL, the change takes place immediately for new jobs entering the system. The change does not affect running jobs, unless the application in the job accesses the system library list directly.

**Time separator (QTIMSEP) system value:** The time separator (QTIMSEP) is used to specify the character separator for time. This value is used as the time separator for the default value of the TIMSEP job attribute. This value is also used as the time separator that you can specify on the IPL options prompt.

You can change the time separator to reflect the character used to separate hours and minutes for your country or location. You can change the time separator to any one of the following values:

- A colon (:) as a time separator
- A period (.) as a time separator
- A comma (,) as a time separator
- A blank ( ) as a time separator

If you change this value, the change takes effect for new jobs that enter the system after you make the change.

**Year (QYEAR) system value:** The year (QYEAR) is used to specify the last two digits of the year on the system. This value ranges from 0 through 99. The system assigns the first two digits for the year based on the current setting for the QCENTURY system value. If the calculated year falls outside the range of dates supported by the system (1928 to 2053), the QCENTURY system value is changed so that the calculated year is within the supported range.

If you change this system value:

- QCENTURY is set to 0 if QYEAR is 54 to 99
- QCENTURY is set to 1 if QYEAR is 00 to 27

For example, if you change QYEAR from 95 to 13, the system changes QCENTURY from 0 to 1, indicating a year of 2013. However, if you change QYEAR from 95 to 45, the system will not change QCENTURY, because both 1945 and 2045 are valid dates.

If you change this value, the change takes effect immediately. Changing this value also affects the system value QDATE.

## Device descriptions

Following are the control language (CL) command parameters that you can use to change cultural and linguistic conventions for some display and printer devices.

**Note:** Some printer device descriptions do not allow you to specify a CHRID.

- Character identifier (CHRID) parameter. You can change the character identifier when you create or change device descriptions for printers and displays. Change the character identifier for a printer or display device using one of the following commands:
  - The Create Device Description (Display) (CRTDEVDSPP)
  - The Change Device Description (Display) (CHGDEVDSPP)
  - The Create Device Description (Printer) (CRTDEVPRT)
  - The Change Device Description (Printer) (CHGDEVPRT)
- Keyboard type (KBDTYPE) parameter. You can set the keyboard language type for a keyboard when you create a device description. Set the keyboard language type using the CHGDEVDSPP command.
- Workstation customization (WSCST) parameter. You can set the workstation customization parameter when creating a device to specify the use of a customized keyboard layout. To set this parameter, the display device must be varied off. You can specify the WSCST parameter when using the CRTDEVDSPP command.
- Language type (LNGTYPE) parameter. When you create an ASCII printer using the CRTDEVPRT command, the LNGTYPE parameter describes the default country or region keyboard language identifier for the printer. When you specify the \*SYSVAL value, the QKBDTYPE system value is used.

## Display and printer files

Following are the keywords and command parameters that you can use to change cultural and linguistic values for display files and printer files.

- The Create Display File (CRTDSPF), Change Display File (CHGDSPF), Create Printer File (CRTPRTF), Change Printer file (CHGPRTF), and Override Printer File (OVRPRTF) commands. You can specify a character identifier explicitly:
  - As the QCHRID system value (\*SYSVAL)
  - As a device description or a device default of the output device (\*DEVDD)
  - With the \*JOBCCSID value
  - As using the \*CHRIDCTL system value (\*SYSVAL)
- Character identifier (CHRID) keyword in DDS. Use this field-level keyword to identify fields that should be converted to the character identifier (CHRID) of the device. Use this keyword in conjunction with the CHRID parameter on the CRTDSPF, CHGDSPF, CRTPRTF, CHGPRTF, and OVRPRTF commands. This keyword is ignored, however, when the CHRID parameter of these commands is set to \*JOBCCSID.
- The SRTSEQ parameter and LANGID parameter on the CRTDSPF command. These parameters can be used to specify a sort sequence and a language identifier for a display file.

### Note:



If \*JOBCCSID is not specified for the CHRID parameter of a display file (either directly or indirectly with CHRIDCTL), the CHRID parameter of the display file must be compatible with the job CCSID. Otherwise, unpredictable results might occur when data is displayed or when data is stored in a database file.



## Database files

Following are the command parameters that you can use to change language-dependent values for database files:

- The SRTSEQ, LANGID,



and CCSID parameters



on the Create Physical File (CRTPF) command

•



The SRTSEQ, LANGID, and CCSID parameters on the Change Physical File (CHGPF) command



•



The parameters on the Copy File (CPYF) command



- The SRTSEQ parameter and LANGID parameter on the Create Logical File (CRTLF) command

These parameters can be used to specify a sort sequence and language for a database file.

## DDS keywords for database files

Following are the DDS keywords that you can use to change language-dependent values for database files:

- The CCSID keyword. This keyword can be used to tag character data stored in a database. By default, the CCSID value is taken from the job creating the database file.
- DATFMT, DATSEP, TIMFMT, and TIMSEP keywords in DDS.

The format of the data type Time (T) field is described by DDS with the TIMFMT keyword that can have \*JOB specified for a value. Similarly, the format of the data type Date (L) is described by DDS with the DATFMT keyword that can have \*JOB specified for a value.

Use the TIMSEP and DATSEP keywords to specify date and time separators.

### UIM menus and panel groups

You can use commands to specify a \*JOBCCSID for a menu or a panel group.

The CHRID parameter on the Create Menu (CRTMNU) command for creating menus can be used to specify a \*JOBCCSID value for a menu. Conversion is automatically done between the CHRID parameter of the device and the CCSID value of the menu.

The CHRID parameter on the Create Panel Group (CRTPNLGRP) command for creating panel groups can be used to specify a \*JOBCCSID value for panel groups. Conversion is automatically done between the CHRID of the device and the CCSID of the panel group and the CCSID of the job.

---

## Globalization checklists

This topic provides links to the numerous checklists that are presented within the globalization topics. They are useful tools for insuring that you have considered key issues that will be important in each particular aspect of your globalization planning and implementation. The following table provides the link to each checklist and shows where you can go to read related topics.

| Checklist               | Where you can go to read related topics  |
|-------------------------|--|
| Globalization planning  | This comprehensive two-part checklist, which you can find in the Set up an iSeries server with a national language version topic, provides a good summary of issues relating to hardware installation, software installation, and system configuration.  |
| Application design      | Use this checklist as you begin planning the development of a globalized application. You can find this checklist in the Design global applications topic.   |
| User interface design   | Use this checklist to insure that the user interfaces you create for your applications anticipate unique requirements from multiple linguistic and cultural environments. You can find this checklist in the User interfaces topic.  |
| Bidirectional support   | Languages such as Arabic and Hebrew are displayed in a right-to-left direction. Because of this, you must take care that your applications handle bidirectional data properly, and that your interfaces can accommodate this presentation of text and data. You can find this checklist in the Work with bidirectional data topic. |
| DBCS application design | You need to take numerous issues under consideration when you develop applications that use double-byte character set (DBCS) support. You can find this checklist in the Work with DBCS data topic.  |







Printed in U.S.A.