IBM

System i

Programming
i5/OS globalization

Version 6 Release 1

IBM

System i

Programming
i5/OS globalization

Version 6 Release 1

Note

Before using this information and the product it supports, read the information in "Notices," on page 447.

This edition applies to version 6, release 1, modification 0 of IBM i5/OS (product number 5761-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

i5/OS globalization

The i5/OS® operating system is designed to support the culture and languages of many countries around the world. 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.

Globalizing your e-business is no longer a luxury; it is a necessity. As the Internet transcends national and geographical boundaries, the concept of doing business within a single country is quickly giving way to the need to compete in an international marketplace.

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.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

Related information

Globalize your On Demand Business Web site

System i Globalization Web site

What's new for V6R1

Read about new or significantly changed information for the i5/OS globalization topic collection.

1. With V6R1, i5/OS support for the International Components for Unicode (ICU) libraries has been updated to be based on ICU version 3.4. This includes the support for the IUC-based sort support. For more information, see "ICU-based sort support" on page 172.
2. With V6R1, the system service language can now be a different language from the operating system language. For more information, see "Installation preparation and national languages" on page 31.

How to see what's new or changed

To help you see where technical changes have been made, the information center uses:
- The » image to mark where new or changed information begins.
- The « image to mark where new or changed information ends.

In PDF files, you might see revision bars (|) in the left margin of new and changed information.

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

PDF file for i5/OS globalization

You can view and print a PDF file of this information.

To view or download the PDF version of this document, select i5/OS globalization (about 7854 KB).

Saving PDF files

To save a PDF on your workstation for viewing or printing:

1. Right-click the PDF link in your browser.
2. Click the option that saves the PDF locally.
3. Navigate to the directory in which you want to save the PDF.
4. Click **Save**.

Downloading Adobe Reader

You need Adobe® Reader installed on your system to view or print these PDFs. You can download a free copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html) .

Globalization overview

The i5/OS operating system supports many languages, thus working as you expect it to work from both a linguistic and a cultural point of view. You can work in the language of your choice. The i5/OS operating system also ensures that the data you send to and receive from the system appears in the form and order you expect.

The i5/OS operating system 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 system and the program code on a Spanish system 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 a function key for online help information about a U.S. English system, while you see *Ayuda* on a Spanish system. Using the same program code with different sets of textual data allows the i5/OS operating system to support more than one language on a single system.

Multiple language support

The i5/OS operating system provides the tools and functions 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 system to translate, present, and process data in a global environment.

When you install secondary languages on your system, you can set up your system with user interfaces (that is, textual data) for any of the national language versions (NLVs) 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.

National language version

A *national language version (NLV)* is a version of the i5/OS operating system that contains a predefined set of language-dependent values, such as date format, time format, and sort sequence, for a particular language.

When you order an i5/OS 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 might 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. The primary language you designate is the feature code identified when you ordered the operating system. You designate all other national language versions as secondary languages.

You must order some of the licensed programs for your system 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 system, the licensed program might not install correctly. Licensed programs with different feature codes as the primary language can be installed as a secondary language.

Related concepts

"National language version feature codes" on page 247
This table lists the available national language version feature codes on the i5/OS operating system. When you order an i5/OS licensed program, you identify the national language version you want by specifying a language feature code.

"Setting up i5/OS with a national language version" on page 29
The steps to install and configure a national language version on the i5/OS operating system include selecting and installing hardware, installing software, and configuring your environment to run in a globalized setting.

i5/OS translations

i5/OS, or portions of i5/OS, are translated into languages listed in this topic. Not all portions of i5/OS are translated into every language.

- Albanian
- Arabic
- Brazilian Portuguese
- Bulgarian
- Croatian
- Czech
- Danish
- Dutch (this includes Belgian Dutch)
- English (this includes Belgian English)
- Estonian
- Farsi
- Finnish
- French (this includes Belgian and Canadian French)
- German
- Greek
- Hebrew
- Hungarian

- Icelandic
- Italian
- Japanese
- Korean
- Laotian
- Latvian
- Lithuanian
- Macedonian
- Norwegian
- Polish
- Portuguese
- Romanian
- Russian
- Serbian
- Simplified Chinese
- Slovakian
- Slovenian
- Spanish
- Swedish
- Thai
- Traditional Chinese
- Turkish
- Vietnamese

National language design in i5/OS

The national language design in i5/OS defines the functions your application software can use to support national languages.

Character representation

Character representation in the system is controlled by 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 the i5/OS operating system 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.

Encoding scheme:

The Character Data Representation Architecture (CDRA) system of tags uses an encoding scheme to specify many rules.

The rules include:

- 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 that 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 might 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 might 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 might 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 might support an A grave character (À). The target CCSID might 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.

Related concepts

"Recommendations and guidelines for using CCSIDs" on page 126
These recommendations are useful when you write globalized applications.

Related reference

"Default character data conversion that can use substitution" on page 441
The default CCSID conversions use substitution because the character sets within the CCSIDs are different. The table shows which CCSIDs (From CCSID column) can be substituted by other CCSIDs (To CCSID column).

Related information

Character Conversion APIs

Coded character set identifier values:

CDRA defines the range of values for CCSIDs (coded character set identifiers).

The values include:

CCSID value	Purpose or meaning
00000	Use next higher hierarchical CCSID
00001 through 65533	IBM-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 might be in a data structure that is logically associated with the data object (explicit tagging), or it might be inherited from the tag field associated with the other objects within the operating system (implicit tagging).

Related concepts

"Set job attributes (QSETJOBATR) system value" on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

Related information

iSeries CCSID information

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 i5/OS 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 the i5/OS operating system.
- Multiple systems 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.

Related concepts

"CCSID reference information" on page 333
Coded character set identifier (CCSID) is a 16-bit number that includes a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and other information that uniquely identifies the coded graphic-character representation.

Character processing

Character processing on the i5/OS operating system is controlled by specific coding rules and guidelines that ensure consistent processing of character data.

The 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

Related concepts

"Developing globalized applications" on page 50
Globalized applications are applications that have national language support. National language support allows users to enter, store, process, retrieve, print, and display data in their chosen language. It also allows users to see and enter commands, prompts, messages, and documentation in their chosen language, in formats matching their cultural expectations.

Character presentation

Character presentation on the i5/OS operating system 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 "Developing globalized applications" on page 50.

Globalization hardware support

Hardware, in this context, means the physical keyboards, displays, printers, and controllers that make up a System i product. The extent to which this hardware supports national languages might 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 the user interface subset of these rules as guidelines to ensure translation goes smoothly. A subset of these rules is provided in "User interfaces" on page 64.

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. The locale support is provided to supplement the job value options that the i5/OS operating system previously has provided.

Many locales are included with the i5/OS operating system. In addition, locale definition source files are provided for locale customization. A locale definition source file contains one or more categories that describe (or make up) a locale.

Related concepts

"Installing and enabling locales" on page 44
If you are installing a new release, you can request that library QSYSLOCALE be installed on the system at that time.

"Working with locales" on page 200
Locales are used primarily in ILE-based application programs. Additionally, the Retrieve Locale Information (OPM, QLGRTVLC; ILE, QlgRetrieveLocaleInformation) API retrieves one or all categories of a locale.

Linguistic and cultural values

Linguistic and cultural conventions include any system 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 might 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.

Where you can change linguistic and cultural values on i5/OS

Settings of cultural and linguistic conventions are supported at different levels in the i5/OS operating system.

The system is structured in the following way:

System	
Subsystem A	Subsystem B
Job	Job
Job	Job

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.

Related tasks

"Enabling the secondary language" on page 41
You must ensure that secondary languages can be used after they have been installed on the system.

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 the i5/OS operating system when you set job attributes and object attributes for an individual user. Job attributes can also be used as defaults for setting object attributes that are created or changed under the control of that job.

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

- 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.

Related reference

Create User Profile (CRTUSRPRF) command

Change User Profile (CHGUSRPRF) command

Subsystems

A subsystem is a single, predefined operating environment through which the system coordinates the work flow and resource use.

The system can contain several subsystems, all operating independently of each other. Subsystems manage resources. The runtime 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.

Subsystem descriptions for secondary language users

You can create and use a subsystem description 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 system is included with a workstation entry to support DBCS display devices.

Related concepts

Work management

"Packaging and installation process" on page 55
You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

Creating a subsystem description:

IBM-supplied subsystem descriptions have been provided as examples and as backup for user-created subsystem descriptions. Therefore, you should not change 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 either copy and then change an existing subsystem description, or create an entirely new description.

To copy an existing subsystem description, follow these steps:
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, follow these steps:
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 (CRTJOBD).
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. CRTCLS (Create a class).
5. ADDRTGE (Add routing entries to the subsystem description).

Related reference

Create Duplicate Object (CRTDUPOBJ) command

Create Subsystem Description (CRTSBSD) command

Add Work Station Entry (ADDWSE) command

Add Job Queue Entry (ADDJOBQE) command

Add Communications Entry (ADDCMNE) command

Add Autostart Job Entry (ADDAJE) command

Add Prestart Job Entry (ADDPJE) command

Create Class (CRTCLS) command

Add Routing Entry (ADDRTGE) command

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 signon 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:

A workstation entry, which is an entry in the subsystem description, specifies the workstations from which users can sign on to the subsystem or from which interactive jobs can transfer to the subsystem.

Here are the items that you can specify 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. Here is an example of adding a workstation entry:

```
ADDWSE SBSD(USERLIB/ABC) WRKSTN(DSP12)
JOBD(USERLIB/WSE)
```

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

```
CHGWSE SBSD(USERLIB/ABC) WRKSTN(DSP12)
JOBD(USERLIB/NEWJD)
```

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

```
RMVWSE SBSD(USERLIB/ABC) WRKSTN(DSP12)
```

Related reference

Transfer Job (TFRJOB) command

Add Work Station Entry (ADDWSE) command

Change Work Station Entry (CHGWSE) command

Remove Work Station Entry (RMVWSE) command

Starting a subsystem:

After 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
```

Related reference

Start Subsystem (STRSBS) command

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).

Related concepts

"Database management" on page 130
Database management support provides default coded character set identifier (CCSID) values for database files on the system. All database files are assigned a CCSID. At file creation time, the CCSID is either explicitly assigned through DDS, SQL, or IDDU, or implicitly assigned the job default CCSID (DFTCCSID).

Related reference

Submit Job (SBMJOB) command

Change Job (CHGJOB) command

Coded character set identifier job attribute:

When an interactive job is started on the i5/OS operating system, 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 SBMJOB 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 language version is installed on the system.

Job default coded character set identifier:

A job attribute, job default CCSID (DFTCCSID), is created for jobs with a CCSID of 65535. The DFTCCSID value is used by a 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).

When 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.

Related concepts

"Database management" on page 130
Database management support provides default coded character set identifier (CCSID) values for database files on the system. All database files are assigned a CCSID. At file creation time, the CCSID is either explicitly assigned through DDS, SQL, or IDDU, or implicitly assigned the job default CCSID (DFTCCSID).

"Graphic character conversion tables" on page 308
Table (*TBL) objects support non-CCSID conversions from one code page to another. The system-supplied table objects are located in the QUSRSYS library.

"Language identifiers and associated default CCSIDs" on page 358
This table shows the language identifiers and the job default CCSID (DFTCCSID) values associated with those identifiers.

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 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.

Related concepts

"System library list (QSYSLIBL) system value" on page 25
The system library list (QSYSLIBL) system value is used as the first part of the library list associated with a job.

"Packaging and installation process" on page 55
You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

System values

The system values of the primary language on the system are used as system-wide cultural and linguistic defaults. Therefore, if you change the primary language on the system, each varying system value resets to the default system value of 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, QCNTRYID, QCURSYM, QDATFMT, QDATSEP, QDECFMT, QKBDTYPE, QLANGID, and QTIMSEP) are shown in Default system values in the Reference section.

Related concepts

"Default system values for national language versions" on page 255
Jobs and functions on the i5/OS operating system use system values as default values.

System values

"Configuring the primary language" on page 38
A primary language consists of program code, textual data for each licensed program ordered, and default national language cultural values.

Related reference

Work with System Value (WRKSYSVAL) command

Century (QCENTURY) system value:

The century (QCENTURY) system value specifies the century. It is used with the system values QDATE and QYEAR to determine the specific date currently used by the system.

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, the 20th/21st century boundary is defined 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.

Related concepts

"System date (QDATE) system value" on page 18
The system date (QDATE) system value indicates the year, the month, and the day on the system.

"Year (QYEAR) system value" on page 26
The year (QYEAR) system value specifies the last 2 digits of the year on the system.

Character identifier (QCHRID) system value:

The character identifier (QCHRID) system value specifies 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.

Related concepts

"Object-level coded character set identifier 65535" on page 137
CCSID 65535 is the default object-level CCSID for message files and message queues.

Character identifier control (QCHRIDCTL) 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 of the create, change, or override command for display files, printer files, and panel groups before this attribute can 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.

Related concepts

"Display files" on page 132
When a display file object is created, it is tagged with the coded character set identifier (CCSID) of the source file.

Coded character set identifier (QCCSID) system value:

The coded character set identifier (QCCSID) system value specifies the CCSID for the i5/OS operating system.

As shipped, the CCSID is set to CCSID 65535. CCSID 65535 means that all character data tagging support on the system is turned off, which is not generally recommended.

If you use Java™ or WebSphere®, or if you plan to transfer data between the i5/OS operating system and another client, then this value (or the corresponding value on the user profile) should be set to match the CCSID of your data. If the value is 65 535, then the encoding of the data on the system is unknown.

If you leave this value at 65535, then you cannot get the results you expect when working from a client on the i5/OS operating system, or the connection might not work at all.

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.

If a job is started with a single byte CCSID, (from either this value or the user profile value) then that job will not support double-byte language (DBCS) users.

Related concepts

"Country or region identifier (QCNTRYID) system value" on page 17
The country or region identifier (QCNTRYID) system value indicates the default country or region identifier for the system.

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

"Database file attributes" on page 79
Database attributes, such as coded character set identifier (CCSID), sort sequence (SRTSEQ), and language identifier (LANGID), are cultural dependent.

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Coordinated universal time offset (QUTCOFFSET) system value:

The coordinated universal time offset (QUTCOFFSET) system value specifies 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 can 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 saving time begins or ends. Caracas, Venezuela does not observe a daylight saving time, and its system value remains constant.

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

Country or region identifier (QCNTRYID) system value:

The country or region identifier (QCNTRYID) system value indicates 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 QCNTRYID system value and the QCCSID system value.

Related concepts

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

"Coded character set identifier (QCCSID) system value" on page 16
The coded character set identifier (QCCSID) system value specifies the CCSID for the i5/OS operating system.

"Country and region identifiers" on page 249
This table lists the country and region identifiers.

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

"Coding globalized applications with high-level languages" on page 97
Your major goal must be to have only one general set of running code that is common for all language versions and to make your programs table-driven as much as possible.

Currency symbol (QCURSYM) system value:

The currency symbol (QCURSYM) system value verifies 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.

Related concepts

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Related reference

EDTWRD (Edit Word) keyword for display files

EDTCDE (Edit Code) keyword for display files

System date (QDATE) system value:

The system date (QDATE) system value indicates 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 might affect the system values for QCENTURY, QYEAR, QMONTH, QDAY, and QDAYOFWEEK. Any change you make to QDATE takes effect immediately.

Related concepts

"Century (QCENTURY) system value" on page 14
The century (QCENTURY) system value specifies the century. It is used with the system values QDATE and QYEAR to determine the specific date currently used by the system.

"Year (QYEAR) system value" on page 26
The year (QYEAR) system value specifies the last 2 digits of the year on the system.

"Month of the year (QMONTH) system value" on page 23
The month of the year (QMONTH) system value indicates the month of the year on the system.

"Day of the month (QDAY) system value" on page 19
The day of the month (QDAY) system value indicates the day of the month on the system. This value must be a valid day of the month or of the year if you are using the Julian date format.

"Day of week (QDAYOFWEEK) system value" on page 19
The day of week (QDAYOFWEEK) system value specifies the day of the week on the system.

Date format (QDATFMT) system value:

The date format (QDATFMT) system value is used for the default value for the DATFMT job attribute. This system value also determines the format in which a date can be specified on the initial program load (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.

Related concepts

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG

message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Date separator (QDATSEP) system value:

The date separator (QDATSEP) system value is used as the date separator for the default value of the DATSEP job attribute. It is also used as the date separator you can specify on the initial program load (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.

Related concepts

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Day of the month (QDAY) system value:

The day of the month (QDAY) system value indicates the day of the month on the system. This value must be a valid day of the month or of the year if you are using the 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.

Related concepts

"System date (QDATE) system value" on page 18
The system date (QDATE) system value indicates the year, the month, and the day on the system.

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 cannot be set correctly if your system is not using the Gregorian calendar.

Related concepts

"System date (QDATE) system value" on page 18
The system date (QDATE) system value indicates the year, the month, and the day on the system.

"Leap year adjustment (QLEAPADJ) system value" on page 23
The Leap year adjustment (QLEAPADJ) system value adjusts the system algorithms for the leap year in different calendar systems.

DBCS system indicator (QIGC) system value:

The DBCS system indicator (QIGC) system value specifies whether a double-byte character set (DBCS) national language version (NLV) 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.

Beginning with i5/OS V5R3, any NLV can support DBCS. Therefore, QIGC is always set to 1 (or on). If you have applications that check this value, update them to use the job level DBCS indicator. You can use the Retrieve Job Information (QUSRJOBI) API to get the job's IGC value.

You cannot change this value.

Related concepts

"Recommendations and guidelines for using CCSIDs" on page 126
These recommendations are useful when you write globalized applications.

Related information

Retrieve Job Information (QUSRJOBI) API

DBCS font name (QIGCCDEFNT) system value:

The DBCS font name (QIGCCDEFNT) system value is used when the system transforms SNA character string (SCS) data with shift in/shift out (SI/SO) characters into a spooled file that is composed of Advanced Function Presentation™ data stream (AFPDS).

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:

The decimal format (QDECFMT) system value determines the type of zero suppression and decimal point character used by DDS edit codes 1 through 4 and A through M. It also determines the decimal point character for decimal input fields in the interface.

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 effect immediately.

Related concepts

"Set job attributes (QSETJOBATR) system value" on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

"Decimal formats" on page 86
You can change the decimal format with the QDECFMT system value to reflect the way decimals are presented for your country or location.

Language identifier (QLANGID) system value:

The language identifier (QLANGID) system value specifies the default language identifier for the system.

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.

Related concepts

"Country or region identifier (QCNTRYID) system value" on page 17
The country or region identifier (QCNTRYID) system value indicates the default country or region identifier for the system.

"Sort sequence (QSRTSEQ) system value" on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

"Coded character set identifier (QCCSID) system value" on page 16
The coded character set identifier (QCCSID) system value specifies the CCSID for the i5/OS operating system.

"Database file attributes" on page 79
Database attributes, such as coded character set identifier (CCSID), sort sequence (SRTSEQ), and language identifier (LANGID), are cultural dependent.

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

"Coding globalized applications with high-level languages" on page 97
Your major goal must be to have only one general set of running code that is common for all language versions and to make your programs table-driven as much as possible.

"Session manager" on page 99
For all applications that use a session manager, you must ensure that the output data stream has no X'3F' values in it. The i5/OS operating system uses X'3F' values to make a screen blank.

"ILE RPG sort sequence" on page 100
The ILE RPG feature, an option of the IBM® WebSphere Development Studio for System i licensed program, provides the possibility for a user to specify a sort sequence table and to use the table in comparison operations that are performed with nonnumeric data.

"DB2 and SQL sort sequence" on page 103
For Interactive SQL, the SRTSEQ and LANGID parameters can be specified on the STRSQL command. You can change these parameters by using the session services for interactive displays.

"System i Access sort sequence" on page 104
You can specify the sort sequence in System i Access functions. When performing queries on the system databases and SQL tables, you can specify the system-supplied or user-supplied sort sequence tables.

Language indicator for keyboard type (QKBDTYPE) system value:

The language indicator for the keyboard type (QKBDTYPE) system value specifies the language character set for the keyboard.

This value is used as the default keyboard type when you create a display device description.

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

Related concepts

“National language keyboard types and SBCS code pages” on page 303
This table lists the keyboard types and code pages for each national language supported by the i5/OS operating system. The Create Device Display (CRTDEVDSP) command uses the KBDTYPE parameter.

Leap year adjustment (QLEAPADJ) system value:

The Leap year adjustment (QLEAPADJ) system value adjusts the system algorithms for the leap year in different calendar systems.

This system value is set by the operating system. If your system observes the Gregorian calendar, this system value should be zero. You cannot edit this system value.

For more information about the Leap year adjustment (QLEAPADJ) system value, see Date and time system values: Leap year adjustment.

Related concepts

“Day of week (QDAYOFWEEK) system value” on page 19
The day of week (QDAYOFWEEK) system value specifies the day of the week on the system.

“Information in message CPX8416” on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Locale (QLOCALE) system value:

The locale (QLOCALE) system value specifies 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.
path-name	The path name of the locale to be used.

Related concepts

“Set job attributes (QSETJOBATR) system value” on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

Month of the year (QMONTH) system value:

The month of the year (QMONTH) system value indicates the month of the year on the system.

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 system 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.

Related concepts

"System date (QDATE) system value" on page 18
The system date (QDATE) system value indicates the year, the month, and the day on the system.

Set job attributes (QSETJOBATR) system value:

The set job attributes (QSETJOBATR) system value sets 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.

Related concepts

"Coded character set identifier values" on page 6
CDRA defines the range of values for CCSIDs (coded character set identifiers).

"Date formats" on page 83
There is no worldwide standard for the presentation of dates. Therefore, the date format should always be stored externally as part of the textual data.

"Date separators" on page 83
The date separator for presentation should always be stored externally as part of the textual data.

"Decimal format (QDECFMT) system value" on page 21
The decimal format (QDECFMT) system value determines the type of zero suppression and decimal point character used by DDS edit codes 1 through 4 and A through M. It also determines the decimal point character for decimal input fields in the interface.

"Sort sequences" on page 86
The i5/OS operating system supports sort sequence. By using one of the listed options, you can order your data according to cultural-dependent requirements for specific applications.

"Time separators" on page 85
The i5/OS operating system allows several valid time separators.

"Locale (QLOCALE) system value" on page 23
The locale (QLOCALE) system value specifies 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.

Sort sequence (QSRTSEQ) system value:

The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines 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. **Note:** 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 can 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. This sort sequence table can be used to sort Unicode and SBCS data.

Related concepts

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

"Sort sequence tables" on page 156
A sort sequence table is an object that contains the weight of each single-byte graphic character within a specified coded character set identifier (CCSID). The system-recognized identifier for the sort sequence table object type is *TBL.

"Database file attributes" on page 79
Database attributes, such as coded character set identifier (CCSID), sort sequence (SRTSEQ), and language identifier (LANGID), are cultural dependent.

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"ILE RPG sort sequence" on page 100
The ILE RPG feature, an option of the IBM WebSphere Development Studio for System i licensed program, provides the possibility for a user to specify a sort sequence table and to use the table in comparison operations that are performed with nonnumeric data.

"DB2 and SQL sort sequence" on page 103
For Interactive SQL, the SRTSEQ and LANGID parameters can be specified on the STRSQL command. You can change these parameters by using the session services for interactive displays.

"System i Access sort sequence" on page 104
You can specify the sort sequence in System i Access functions. When performing queries on the system databases and SQL tables, you can specify the system-supplied or user-supplied sort sequence tables.

"Sort sequence support in work management" on page 168
Work management involves the assigning of the SRTSEQ value at the job level, the user profile level, and the system value level.

System library list (QSYSLIBL) system value:

The system library list (QSYSLIBL) system value 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.

Related concepts

"Job library list" on page 14
The language used for textual data (displays, messages, printed output, and online help information) is controlled by the library list for the job.

Time separator (QTIMSEP) system value:

The time separator (QTIMSEP) system value specifies 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.

Related concepts

"Job attributes" on page 80
Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Year (QYEAR) system value:

The year (QYEAR) system value specifies the last 2 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.

Related concepts

"Century (QCENTURY) system value" on page 14
The century (QCENTURY) system value specifies the century. It is used with the system values QDATE and QYEAR to determine the specific date currently used by the system.

"System date (QDATE) system value" on page 18
The system date (QDATE) system value indicates the year, the month, and the day on the system.

Device descriptions

These control language (CL) command parameters can be used 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) (CRTDEVDSP)
  - The Change Device Description (Display) (CHGDEVDSP)
  - 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 CHGDEVDSP 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 CRTDEVDSP 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.

Related concepts

"Packaging and installation process" on page 55
You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

Related reference

Create Device Description (Display) (CRTDEVDSP) command

Change Device Description (Display) (CHGDEVDSP) command

Create Device Description (Printer) (CRTDEVPRT) command

Change Device Description (Printer) (CHGDEVPRT) command

Display and printer files

These keywords and command parameters can be used 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 (*DEVD)
- 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.

Related reference

Create Display File (CRTDSPF) command

Change Display File (CHGDSPF) command

Create Printer File (CRTPRTF) command

Change Printer File (CHGPRTF) command

Override with Printer File (OVRPRTF) command

Database files

These command parameters and the DDS keywords can be used to change language-dependent values for database files.

You can use the following command parameters:

- 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

You can use the following DDS keywords 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.

  Related reference

  Create Physical File (CRTPF) command

  Change Physical File (CHGPF) command

  Copy File (CPYF) command

  Create Logical File (CRTLF) command

DDS keywords and parameters

UIM menus and panel groups

You can use the CHRID parameter on the Create Menu (CRTMNU) or Create Panel Group (CRTPNLGRP) command 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.

Related reference

Create Menu (CRTMNU) command

Create Panel Group (CRTPNLGRP) command

Setting up i5/OS with a national language version

The steps to install and configure a national language version on the i5/OS operating system include 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 systems, and you can apply the principles when you develop applications for customers who are installing their own national language version on i5/OS.

The feature code identified when you order an i5/OS operating system is the language of your textual data and is called the primary language of the system. Any other language versions that you have ordered are called secondary languages. For secondary languages, the national language version consists of only the textual data for all licensed programs ordered. The program code is not contained in the secondary language version.

The primary language is the language in which the system is serviced and from which all language-dependent or cultural-dependent system values are initialized. In addition, other system objects and functions assume attributes based on the primary language. For example, messages appearing in the history log always appear in the primary language.

Related concepts

"National language version" on page 3
A *national language version (NLV)* is a version of the i5/OS operating system that contains a predefined set of language-dependent values, such as date format, time format, and sort sequence, for a particular language.

"National language version feature codes" on page 247
This table lists the available national language version feature codes on the i5/OS operating system. When you order an i5/OS licensed program, you identify the national language version you want by specifying a language feature code.

"Configuring the primary language" on page 38
A primary language consists of program code, textual data for each licensed program ordered, and default national language cultural values.

"Configuring secondary languages" on page 39
A secondary language consists of textual data for all licensed programs supported for a national language version.

Related information

Installing, upgrading, or deleting i5/OS and related software

How a language is displayed for i5/OS functions

If you want information presented in a language other than the primary language of the system, you must first have a secondary language loaded. When a secondary language is loaded, you have three ways to display information in that language.

Method 1: Placing the language you want at the top of your library list

One way to display information in a secondary language is to change the system part of your library list so the library of the national language you want is positioned before all other libraries in the system library list that contain national language information.

For example, to present the French version of textual data, you can enter the following command to place French information at the top of the library list:

```
CHGSYSLIBL  LIB(QSYS2928)  OPTION(*ADD)
```

To remove a library from the library list enter:

```
CHGSYSLIBL  LIB(QSYS2928)  OPTION(*REMOVE)
```

Note: The authority included with the CHGSYSLIBL command does not allow all users to run the command. As included, you must have *ALLOBJ and *SECADM special authority to use the Change System Library List (CHGSYSLIBL) command.

Method 2: Creating a subsystem for the language you want

A second way to present information in a different language is to follow these steps:

1. Create a subsystem for the secondary language.
2. Define the subsystem system part of the library list entry with the national language version library for the secondary language.

All jobs running in the subsystem use textual data from the secondary language. All jobs that you submit as batch jobs have the national language version library as the first library on the system part of the library list.

Method 3: Changing the library list for your job so that the national language version library for the secondary language is the first library on the system part of the library list

A third way to present information in a different language is to change the library list for your job so that the national language version library for the secondary language is the first library on the system part of the library list. All jobs running in the subsystem use textual data from the secondary language. All jobs that you submit as batch jobs have the national language version library as the first library on the system part of the library list.

How a language of your choice is displayed for licensed programs

Libraries for other licensed programs are either added automatically, or must be added by the user, when needed. If you want to add libraries for other licensed programs to your library list, use the CHGLIBL command.

Related concepts

"Configuring secondary languages" on page 39
A secondary language consists of textual data for all licensed programs supported for a national language version.

Related reference

Change System Library List (CHGSYSLIBL) command

Change Library List (CHGLIBL) command

Installation preparation and national languages

IBM periodically creates program temporary fixes (PTF) to correct existing problems or potential problems within a particular IBM licensed program.

PTFs are designed to fully replace one or more objects in the licensed program. Primary and secondary languages can have language-sensitive online information PTFs.

If the primary language of your system is changed at any time for reasons other than a new release update, the cumulative PTF package of the new primary language should be at the same level as the previous primary national language. PTFs that were associated with the primary language and any secondary language must be applied again. In addition, primary language and secondary language PTFs for the online information need to be ordered by the customer.

| For systems running i5/OS V6R1, or later, you can set the system service language (the dedicated service
| tools (DST) language) to a language different from the operating system language. For more information,
| see Installing, upgrading, or deleting i5/OS and related software.

Related concepts

"Configuring the primary language" on page 38
A primary language consists of program code, textual data for each licensed program ordered, and default national language cultural values.

"Configuring secondary languages" on page 39
A secondary language consists of textual data for all licensed programs supported for a national language version.

Checklist: Globalization planning

When planning to install a multilingual i5/OS operating system, start by completing the Globalization planning checklist for globalization and multilingual support. The checklist consists of two parts, which should be completed sequentially.

Globalization checklist: Part 1

Before you work with a national language, answer the questions in the following table. After you have answered the questions in this table, you can then use "Globalization checklist: Part 2" on page 32 for planning for multilingual support.

Check off	Question	Response
	What national language version for the primary language are you going to install? (Refer to "Setting up i5/OS with a national language version" on page 29.)	
	What program library can it be ordered from? (Refer to "National language version feature codes" on page 247.)	
	Are you going to use a DBCS national language version as a secondary language? (Refer to "Notes on secondary languages when you require English as the primary language" on page 40.)	
	Are you aware that the latest 5250 PC emulation is necessary to support graphics data format (GDF) type?	

Check off	Question	Response
	What national language version for the secondary language are you going to install, if any? (Refer to “Notes on secondary languages when you require English as the primary language” on page 40.)	
	Do you want to change your subsystem to change the language of your initial sign-on display? (Refer to “Notes on secondary languages when you require English as the primary language” on page 40.)	
	What release level of the national language version for the primary language are you ordering? (Refer to “Configuring the primary language” on page 38.)	
	Are the release levels of the national language version for the secondary language the same as the primary language you ordering? (Refer to “Notes on secondary languages when you require English as the primary language” on page 40.)	

Globalization checklist: Part 2

When you have completed “Globalization checklist: Part 1” on page 31 of the checklist, answer the additional questions in part 2.

Check off	Question	Response
	What local workstation controllers support your language? (Refer to “Workstation controllers reference” on page 271.)	
	What display stations and keyboards are required to support your language? (To determine the display stations and keyboards, refer to “Workstation controllers reference” on page 271.)	
	What printers support your language?	
	What keyboard ID are you using for your local devices? (Refer to “Default system values for national language versions” on page 255.)	
	What remote workstation controllers support your language? (Refer to “Workstation controllers reference” on page 271.)	
	What display stations and keyboards support your language from a remote location? (Refer to “Workstation controllers reference” on page 271.)	
	What printers support your language from a remote location?	
	What keyboard ID are you using for your remote devices? (Refer to “Default system values for national language versions” on page 255.)	
	Are you considering the workstation customization function for workstations? (See the Workstation Customization Programming PDF.)	

Check off	Question	Response
	What applications support your languages on the local system? (Contact your marketing support representative in your country.)	
	What applications support your languages on the remote system? (Contact your marketing support representative in your country.)	
	Do you want all your database files with the CCSID of the primary language? (Refer to "Database management" on page 130.)	
	Do you want to work with sort sequence tables in your applications? (Refer to "Sort sequence support in programs" on page 168.)	
	When creating user profiles (user IDs) only certain characters are allowed. You can use any of the following characters in the user profile name: • Any letter (A through Z) • Any number (0 through 9) • These special characters: pound (#), dollar ($), underscore (_), at (@). However, these characters should be avoided for globalized application systems. See "User profile name considerations" on page 36 for more information. See User profiles for more detailed information about user profile considerations.	

Hardware installation and national languages

When installing or changing a device on your system, you must make sure that the device is configured correctly to reflect the keyboard ID that matches the character set and code page of the job CCSID.

Changing the keyboard configuration of a device results in different behavior, similar to adding a new display or printer to the system.

Panels, menus, and messages used by the installation process do not support right-to-left presentation of data. Therefore, online information for the installation appears left to right, in English, for bidirectional languages (such as Arabic and Hebrew).

Related concepts

"Configuring secondary languages" on page 39
A secondary language consists of textual data for all licensed programs supported for a national language version.

Console device

You should make sure that your console device is configured to support the default code page of the primary language you are going to install on the i5/OS operating system.

If the console device supports the code page of the new primary language, panels, messages, and online help will display properly after you change the primary language.

You must change the console device to one that supports the code page of the new primary language before doing the IPL that activates the new primary language. Make sure that autoconfig is on before doing this IPL.

Scenario: Console configured as a single-byte device

Your system has a primary language of English Uppercase DBCS (feature 2938). You decide to change the primary language to Japanese DBCS (feature 2962).

The existing console device on your system is configured as a single-byte-only English device using a code page of 00037. While a single-byte English device supports the installation of all other single-byte national language versions, it does not support the installation of double-byte national language versions like Japanese or Chinese. You must change the console device to one that supports the Japanese DBCS code page before doing the activation IPL.

If you do not change the console device to one that supports the Japanese DBCS code page, the IPL cannot complete.

Scenario: Console configured with an F-type keyboard

Assume that your system has a primary language of English (feature 2924) and you decide to change your primary language to Czech, Farsi, Hungarian, Russian, Polish, Slovakian, or Thai. Also assume that your system console has an F type keyboard (a relatively old keyboard type).

While F type keyboards are supported for many national language versions, they are not supported for Czech, Farsi, Hungarian, Russian, Polish, Slovakian, or Thai. You must change the system console to a device that does not have an F type keyboard.

If you do not change the console device and keyboard, an error results because there is no keyboard mapping table supported for the F type keyboard in your new primary language (Czech, Farsi, Hungarian, Russian, Polish, Slovakian, or Thai). The IPL that occurs when changing the primary language cannot complete.

Workstation considerations

In a multilingual environment, different workstations support different languages on the same system.

Any data that is not tagged with CCSIDs should be stored in separate objects, unless the CCSID for each language is the same. Data that is tagged with CCSIDs (such as message files and database files) do not have to be stored in separate objects.

To correctly retrieve, process, and display data that is not tagged with CCSIDs, the application being used needs to be aware of the language differences, and how they relate to the following items:

- Programmable workstations through System i Access programs
- Nonprogrammable workstations

  Note: The 3486, 3487, 3488 model V, and 3489 displays support all languages (except Thai) listed in 3486, 3487, 3488 Model V, and 3489 Keyboard and Display Part Numbers by Language.
- Keyboards
- Telnet or pass-through implications

  The characters shown on your workstation depend on the keyboard type defined on your source system. If you pass through to the target system and use a virtual device with a different keyboard type, you might not see the same characters as if you were directly attached to the target system, because the target system uses another language.

  Related concepts

“SBCS keyboard and display part numbers by language” on page 291
These tables list the part numbers of the SBCS displays and keyboards that should be used for each language or country supported by IBM System i products.

“Keyboard layouts” on page 278
These keyboard layout samples are provided for your information. The special-character keyboard set is available only with the enhanced keyboard.

Considerations for changing printers

When changing printers, consider the areas of data interchange, data stream, fonts, and host printer emulation.

- Interchange (z/OS® operating systems sending Advanced Function Presentation (AFP™) data for DBCS to i5/OS.)

  AFP data containing DBCS data can be generated on the i5/OS operating system. In addition, the system can receive AFP-generated data from the z/OS system containing DBCS data and print the data on Intelligent Printer Data Stream™ (IPDS™) printers attached to the System i platform. The IPDS printers must be configured with *YES specified for the AFP parameter.
- Data stream

  Printers consist of SNA character string (SCS) and IPDS printers.

  SNA character string (SCS) is a data stream composed of EBCDIC controls, optionally intermixed with end-user data, which is carried within a request/response unit. Host-attached SCS printers can be configured by the systems engineer or by the customer, using a diskette or selection of keys on the printer. The appropriate printer operator’s guide should be used to determine how to configure the SCS printer for the language you are using.

  One of the strengths of IPDS is that independent applications can create source data. The source data from independent applications is merged at the printer to create an integrated mixed data page. For example, text data can be produced on an editor, image data can be the output of a scanner stored in a folder, and graphics data be produced by the Business Graphics Utility program. IPDS makes it possible to integrate application output rather than requiring the use of integrated applications.
- Fonts

  Font types for IPDS printers can be configured through the use of the Create Device Description (Printer) or Change Device Description (Printer) (CRTDEVPRT or CHGDEVPRT) commands. Fonts can be downloaded from the host or can be saved in printer storage.

  For a list of the character identifier (CHRID values) supported by the various printers and languages, see the Printer Device Programming PDF.
- System i Access printer to emulate host printer

  The System i Access programs support multiple languages on a single system. A System i Access user (except for host emulation) can use any single language of choice that is installed on the attached System i platform. If a System i Access user has a host emulation session with five different systems, the user can view a different language on each session. However, the same personal computer ASCII code page must be on all the systems.

  See System i Access of your environment for information about installing and configuring attached PC printers.

  Related concepts

  Advanced Function Presentation

  Related reference

  Create Device Desc (printer) (CRTDEVPRT) command

  Change Device Desc (printer) (CHGDEVPRT) command

Software installation and national languages

If your system communicates with systems using different languages, you need to be careful when specifying configuration names that are exchanged with the remote system.

Do not use characters that might not be available on the keyboard used by the remote system; for example, characters such as a dollar sign ($), pound sign (#), and an at sign (@). For an illustration of the characters that you can use in configuration names, see “Invariant character set” on page 324.

You should limit support of configuration names that use characters outside of the invariant character set to those already in use on existing systems.

Configuration names that might be exchanged with remote systems include:
- Network identifiers
- Location names
- Control point names
- Mode description names
- Class-of-service description names
- User IDs (from the directory entry)

For more information about software installation, see the appropriate software product books.

Related information

Installing, upgrading, or deleting i5/OS and related software

Configuring a national language version

You must configure the national language version on your system before the system can meet your business needs in the multilanguage environment.

User profile name considerations

The user profile name identifies the user to the system. This user profile name is also known as the user ID. It is the name that the user types in the *User* prompt on the Signon display.

The user profile name can be a maximum of 10 characters. The characters can be:
- Any letter (A through Z)
- Any number (0 through 9)
- In addition to these characters, three special codepoints are allowed (x’5B’, x’7B’, x’7C’). For many CCSIDs, including 37, these code points are interpreted as $, #, and @. For other CCSIDs, however, these code points represent other characters. Although these code points are allowed, you should avoid using them because of the potential misinterpretation when multiple CCSIDs are used on a single system. For example, a Spanish-speaking person using CCSID 284 might create a user profile with the name `ESPAÑOL`, but an English-speaking person using CCSID 37 might see this name as `ESPA#OL`.

The user profile name cannot begin with a number.

Note: You can create a user profile such that when a user signs on, the user ID is only numerals. To create a profile like this, specify a Q as the first character, such as Q12345. A user can then sign on by entering 12345 or Q12345 for the *User* prompt on the Signon display.

Related concepts

“Packaging and installation process” on page 55
You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

Related information

User profiles

Service tools

Panels, messages, and online help information for service tools are typically shown in the primary language of the system. Therefore, the workstation from which the system is being serviced must be configured to support the primary language, and the keyboard for the primary language must be attached to that workstation.

Panels, menus, and messages used by the service tools do not support right-to-left presentation of data. Therefore, online information for the service tools appears left to right, in English, for bidirectional languages (such as Arabic and Hebrew).

System and user interfaces

The system interfaces and user interfaces are presented through a workstation or printer.

The workstation controller interprets keystrokes on keyboards according to the mapping determined by the KBDTYPE parameter in the device description. The display presents the data to the user, depending on the code page mapping located in the workstation controller. This code page mapping in the workstation controller is determined by the CHRID parameter in the device description. Each supported keyboard type has a character identifier assigned to it, and the default setting of CHRID in the device description (*KBDTYPE) refers to that character identifier. Ensure the code page of the emulator is set to match the language of the system. For more information, see the help provided by the emulator.

Automatic device configuration:

Automatic configuration defines the local devices and some remote devices to the system.

This means that the devices attached to your system are available for use when the system is running and has a powered-on display. You do not have to use manual configuration to create configuration descriptions for the devices before you can use them. For devices that are able to send configuration information to the workstation controller, the KBDTYPE parameter is set according to the keyboard attached. If the device cannot send KBDTYPE information to the system, the QKBDTYPE keyboard system value is used.

Note: If you use manual configuration to set up a device with a different keyboard type than the hardware reports, automatic configuration changes the device description to match the keyboard attached. To avoid this, each time the device is powered on; you can switch automatic configuration off by setting QAUTOCFG system value to 0 (Off).

Related information

Local Device Configuration PDF

Automatic character set and code page conversion:

The i5/OS operating system provides automatic conversion between character set and code pages for all applications that are enabled for national language support.

This automatic conversion can be controlled in the display, menu, or panel source, or through the CHRID parameter on the control language (CL) commands that create these displays. The character set and code page of the device used by the user is determined by the CHRID parameter in the device description. The CHRID value is normally set to *KBDTYPE.

When the data to be presented is in a character set and code page different from the language of the user, automatic data conversion might occur.

Related concepts

"Working with CCSIDs" on page 126
Using the system implementation of Character Data Representation Architecture (CDRA), you can

achieve consistent representation, processing, and interchange of coded characters (data) on the i5/OS operating system and across IBM Systems. The primary implementation of CDRA on the i5/OS operating system is through coded character set identifier (CCSID) support.

"CCSID reference information" on page 333
Coded character set identifier (CCSID) is a 16-bit number that includes a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and other information that uniquely identifies the coded graphic-character representation.

Printer file conversion:

The printer provides printed output to the user. i5/OS printer support does not do any conversion between the different character sets. For the data to be printed, the user must make sure that the proper character set and code page are specified in the printer and the fonts are in the printer.

If the CHRID value of the printer file is set to *JOBCCSID, the printer joins the CHRID value of the job CCSID to the data to be printed. For externally described printer files, constants within your DDS (data description specification) are converted from the DDS source file CCSID to the character identifier of the job CCSID value.

Configuring the primary language

A primary language consists of program code, textual data for each licensed program ordered, and default national language cultural values.

The primary language is the language in which the system is serviced and from which all language-dependent or cultural-dependent system values are initialized. In addition, other system objects and functions assume attributes based on the primary language. For example, messages appearing in the history log always appear in the primary language.

For each licensed program installed on the system, the national language version for the primary language is in the product library. For example, the i5/OS operating system ordered in Spanish is installed in library QSYS as the primary language.

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.

Selecting and changing the primary language

Choosing your primary language is important. The i5/OS operating system allows you to change your primary language to accommodate your business needs based on the country in which you are operating. Keep in mind, however, that changing the primary language can take several hours or longer to accomplish.

To change a primary language on your system, you can order a different primary language from IBM. If you have a secondary language tape for the language you want as your new primary language, you can change the primary language from that tape. For example, if you have a primary language of U.S. English, and a secondary language of Canadian French, you can use the Canadian French secondary language tape to change your primary language to Canadian French.

When you change a primary or secondary language, and want to continue receiving software and documentation updates for future releases of licensed programs that you are currently using, contact your IBM representative.

Selecting and changing a primary language affects the following operational characteristics of your system:

- Cultural values of the user.

- Language used to communicate with the system through user interfaces presented through a workstation or printer. See the figure in "Example: How locales work" on page 221.
- Implied character identifier (CHRID) of the character data stored in objects other than database files, message files, and message queues on the system.

All user-created database files have an implicit CCSID and are tagged with the job default CCSID (DFTCCSID) unless you provide a CCSID at creation time.

- If you change the primary language and the CCSID for the data remains the same, there is no effect on your system. An example is to change the primary language from the German MNCS to the Italian MNCS, of which both use CCSID 00500. The multinational character set refers to character set 00697 and code page 00500.
- If changing the primary language includes changing the CCSID value, the character data in objects other than database files might not be presented properly through the system and user interfaces. The database manager automatically converts character data unless conversion is suppressed by the application that processes the file. Data in objects other than database files are displayed correctly if the CHRID value of the display file, panel group, or menu is *JOBCCSID.

Because some of the system values are set based on the installed primary language, you should record your current system value settings before you change the primary language of your system. Then, after you change the primary language, you can compare the current system values with the previous system value settings.

When you change the primary language of your system, the CCSID of the text fields in the system-supplied output files might also change. This is because the CCSID is dependent on the installed primary language.

Related concepts

"Setting up i5/OS with a national language version" on page 29
The steps to install and configure a national language version on the i5/OS operating system include selecting and installing hardware, installing software, and configuring your environment to run in a globalized setting.

"Installation preparation and national languages" on page 31
IBM periodically creates program temporary fixes (PTF) to correct existing problems or potential problems within a particular IBM licensed program.

"Default system values for national language versions" on page 255
Jobs and functions on the i5/OS operating system use system values as default values.

"System values" on page 14
The system values of the primary language on the system are used as system-wide cultural and linguistic defaults. Therefore, if you change the primary language on the system, each varying system value resets to the default system value of the new primary language.

Related information

Changing the primary language of your system or logical partition

Configuring secondary languages

A secondary language consists of textual data for all licensed programs supported for a national language version.

When you install a secondary language, the textual data for licensed programs installed on your system is copied into the secondary language library. See the chapter called "Installing a Secondary Language" in the Software Installation PDF for instructions on installing secondary languages.

The program code is not included in the secondary language version.

Secondary language environments

Some multilingual environments have more than one national language version installed. To have a single system support multiple languages, you must have the associated hardware installed. You must also have sufficient disk storage space available to contain all of the system and application textual data for the secondary languages. The amount of disk storage space that is required varies by language and application, but it is typically somewhere in the range of 50 to 300 MB.

The languages currently supported on i5/OS as either primary or secondary languages can be found in National Language Version (NLV) feature codes. Listed are the national language versions, their feature codes, and the program libraries from which they are available.

Each of the national language versions available from the program library (primary or secondary) include cultural- and language-dependent system values for that particular language. Date format, date and time separators, code page and character set, and keyboard types are examples. The system values are initially set to the cultural values of the primary language. By setting up a subsystem, however, you can ensure that the cultural values for the secondary languages are set properly for users of the secondary languages.

Applications can use language values that are available in message CPX8416, in file QCPFMSG, accessed using the library list. Message CPX8416 gives the correct values for the primary or secondary language, depending on the library list.

Except for logical partitioning (LPAR), when you use a multilingual environment, the primary language version and any secondary languages must be at the same release level. You must also order and install the correct devices (workstation controllers, display stations, and printers) to support your languages.

Related concepts

"Setting up i5/OS with a national language version" on page 29
The steps to install and configure a national language version on the i5/OS operating system include selecting and installing hardware, installing software, and configuring your environment to run in a globalized setting.

"How a language is displayed for i5/OS functions" on page 30
If you want information presented in a language other than the primary language of the system, you must first have a secondary language loaded. When a secondary language is loaded, you have three ways to display information in that language.

"Installation preparation and national languages" on page 31
IBM periodically creates program temporary fixes (PTF) to correct existing problems or potential problems within a particular IBM licensed program.

"Hardware installation and national languages" on page 33
When installing or changing a device on your system, you must make sure that the device is configured correctly to reflect the keyboard ID that matches the character set and code page of the job CCSID.

"National language version feature codes" on page 247
This table lists the available national language version feature codes on the i5/OS operating system. When you order an i5/OS licensed program, you identify the national language version you want by specifying a language feature code.

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

Notes on secondary languages when you require English as the primary language:

These considerations are important when you require English as the primary language and want to install DBCS secondary languages.

If you install Japanese DBCS (feature number 5762) as a secondary language and you require English as a primary language

If you install Japanese DBCS (feature number 5762) as a secondary language and you require English as a primary language, you must install English uppercase DBCS (2938) as the primary language. English uppercase DBCS (2938) should be installed because all of its user interface text is in uppercase English letters. Some older Japanese workstation displays do not support lowercase English letters. English Uppercase DBCS allows users to view English text from these Japanese displays, without loss of data. If you install English uppercase DBCS as the primary language, you must respond to all messages in uppercase rather than in lowercase. If you respond in lowercase, you receive an error message.

Set the QKBDTYPE system value to JUB (Japanese English) when using Japanese DBCS (5762) as a secondary language with English Uppercase and Lowercase Support for DBCS (2984). Use JUB as QKBDTYPE because this allows the primary language users to enter English characters, but not Katakana characters.

If Simplified Chinese, Traditional Chinese, or Korean is used as a secondary language and English is required as the primary language

If Simplified Chinese, Traditional Chinese, or Korean is used as a secondary language and English is required as the primary language, use English Uppercase and Lowercase (2984) as a primary language. You should use English Uppercase and Lowercase because English uppercase and lowercase characters can be displayed on devices supporting these DBCS languages.

In other cases

In cases other than those described in this topic, use English (2924) as a primary language.

Enabling the secondary language:

You must ensure that secondary languages can be used after they have been installed on the system.

To enable the secondary language, follow these steps:

1. Add the secondary language library to the beginning of the user's system part of the library list. To do this, use one of the following ways:
   - Use the Change System Library List (CHGSYSLIBL) command to add the national language library you want to the top of the library list.

     The command can be in an initial program specified in the user profile so that the user does not have to enter the command at every sign-on.

     The authority included with the CHGSYSLIBL command does not allow all users to run the command. To enable a user to run the CHGSYSLIBL command without granting the user rights to the command, you can write a CL program containing the command. The program is owned by the security officer and adopts the security officer's authority when the program is created. Any user with authority to run the program can use it to change the system part of the library list in the user's job.
   - Use a separate subsystem for a secondary language. To do this, follow these steps:
     a. Create a subsystem description for secondary language users (for example, QGPL/DANISH).
     b. Specify the secondary language library for the Subsystem library (SYSLIBLE) attribute (for example, QSYS2926).
     c. Specify the sign-on display file from the secondary language library for the Sign-on display file (SGNDSPF) attribute (for example, QSYS2926/QDSIGNON).

d. Use the Remove Work Station Entry (RMVWSE) command to remove the appropriate display devices from the interactive subsystem, and then use the Add Work Station Entry (ADDWSE) command to add these devices to the secondary language subsystem.

   When you use these commands, no one can be signed on to the devices that you are removing.

e. If you want to use separate job queues (JOBQ) and output queues (OUTQ) for a secondary language, you can create these queues in the secondary language library (for example, QSYS2926). Attach the job queue to the secondary language subsystem (for example, QGPL/DANISH).

- You might have licensed programs that have secondary language libraries and that are not on the i5/OS secondary language tape. You should add those secondary language libraries to the library list before the primary language product libraries. Use the Change System Library List (CHGSYSLIBL) command to add the secondary language libraries to the library list if the product libraries are in the system part of the library list.

2. Specify the keyboard ID for the secondary language in the device description for the display station.
   a. Turn off your device.
   b. Use the Change Device Description Display (CHGDEVDSP) command to specify the keyboard ID for the secondary language in the device description.
   c. Use the Vary Configuration (VRYCFG) command to turn on the device.
3. Change the date format to reflect the date format of your language. The date format, date separator, and time separator can only be changed using the CHGJOB command for secondary language users. If you use the CHGSYSVAL command to change these values, all primary language users and all secondary language users have this information changed. The following table illustrates this and shows the ways the date and other NLS-related job attributes should be specified for secondary language users.

	CHGJOB	**CRTJOBD**	**CHGJOBD**	**CRTUSRPRF**	**CHGUSRPRF**
Date	X	X	X		
Date format	X				
Date separator	X				
Time separator	X				
Character set identifier	X			X	X
Language identifier	X			X	X
Sort sequence	X			X	X
Country or region identifier	X			X	X

Note: The following commands are used in this table:
- Change Job (CHGJOB) command
- Create Job Description (CRTJOBD) command
- Change Job Description (CHGJOBD) command
- Create User Profile (CRTUSRPRF) command
- Change User Profile (CHGUSRPRF) command

4. Change the CCSID value to reflect the CCSID of the secondary language that you want to use. You can set the CCSID value for all jobs to run under your user profile by using the Change User Profile (CHGUSRPRF) command. This change takes effect for any jobs that enter the system using your profile after you have made the change.

You can set the CCSID value for a batch job to be run using the CCSID parameter on the Submit Job (SBMJOB) command. You can change the CCSID of a job that is running by using the Change Job (CHGJOB) command.

5. Ensure that your data in objects other than database files and message files prints correctly. To do this, you might want to direct all of your printed output to a print queue that contains printer output only for the character identifier of your language.
   a. Use the Create Output Queue (CRTOUTQ) command to create a printer queue.
   b. Use the OUTQ parameter of the Change Job (CHGJOB) command to change your job output queue.

      ```
      CHGJOB OUTQ(output_queue)
      ```

      Note: You can use the Change User Profile (CHGUSRPRF) command instead to make a more permanent change to the OUTQ parameter. Then, each time you sign on to the system, the correct output queue is used.

      If the printer supports changing the code page, you can use the *JOBCCSID value in the printer file.
6. Change other cultural- and language-dependent values to the secondary language you want to use if you do not want to use the system values. Use the Change Job (CHGJOB) command to change the cultural- and language-dependent values.

   Related concepts

   "Working with CCSIDs" on page 126
   Using the system implementation of Character Data Representation Architecture (CDRA), you can achieve consistent representation, processing, and interchange of coded characters (data) on the i5/OS operating system and across IBM Systems. The primary implementation of CDRA on the i5/OS operating system is through coded character set identifier (CCSID) support.

   "Linguistic and cultural values" on page 8
   Linguistic and cultural conventions include any system values, attributes, or settings that can be altered to suit a country or language.

   Related reference

   Change System Library List (CHGSYSLIBL) command

   Create Output Queue (CRTOUTQ) command

   Change Job (CHGJOB) command

Multilingual support:

Multilingual support on the i5/OS operating system is the support that includes more than one language on one system.

A system that works in multiple languages must be able to handle a variety of cultural and linguistic characteristics such as the following:

- Graphic characters, such as an e accent grave (è)
- Currency symbols, such as the Pound Sterling symbol
- Date formats, such as 24.06.93
- Time formats, such as 23:59
- Sort sequences, such as a, b, c....

The system must also handle differences, such as the direction in which text prints and displays. For example, all text of Latin-based languages, such as French and Spanish, displays from left to right across a display. However, the general direction of Arabic and Hebrew text is from right to left across a display. The system displays text, prints text, and allows data entry left to right for some languages and right to left for other languages.

Printing and displaying text left to right for some languages and right to left for others is not enough, though. Numbers and Latin character phrases that are included in Arabic and Hebrew text display and print from left to right. For example, Hebrew text generally flows from right to left across a display. When Hebrew text includes a street address, the street name flows right to left, but the address number flows left to right. Similarly, if Hebrew text includes a Latin name, such as John Smith, the Latin name flows from left to right. Because this text flows both right to left and left to right (bidirectionally), the system displays and prints text bidirectionally.

Multilingual network

Two or more systems, each using a different primary language, can interchange data. Because data is flowing between systems with different primary languages, the data must have a CCSID assigned. When data has a CCSID assigned, data integrity is maintained. Thus, character data is correctly displayed for the receiving user.

Installing and enabling locales

If you are installing a new release, you can request that library QSYSLOCALE be installed on the system at that time.

To install library QSYSLOCALE at a later time, type GO LICPGM and press the Enter key. Scroll until you find *Extended NLS Support*. Select option 1 to install *Extended NLS Support*.

Locales can be enabled on the system by using system values or user profiles.

Enabling locales with system values

Two system values are related to locales:

QLOCALE
: The system value specifying the locale object. The default is *NONE. Other possible values are:

- *C

  The C locale is assigned for this user (same result as using *POSIX)
- *POSIX

  The POSIX (Portable Operating System Interface for Computer Environments) locale path name is assigned for this user.
- locale path name

  The path name of the locale to be assigned for this user.

QSETJOBATR
: A system value that sets job attributes at job start time. The default is *NONE. The following values indicate the job attributes that are to be set from the locale object specified by QLOCALE:

- *CCSID (Coded character set identifier)

  The CCSID associated with a locale when the locale object is created.
- *DATFMT (Date format)

  The date format is determined from the locale object.
- *DATSEP (Date separator)

  The date separator is determined from the locale object.
- *SRTSEQ (Sort sequence)

  The sort sequence is determined from the locale object
- *TIMSEP (Time separator)

  The time separator is determined from the locale object.
- *DECFMT (Decimal format)

  The decimal format is determined from the locale object.

Enabling locales with user profiles

Two parameters on the user profile are related to locales:

LOCALE
The parameter value specifying the locale object to use for the LANG environment variable. The default is *NONE. Other possible values are:

- *SYSVAL

  The system value QLOCALE is used to determine the locale path name to be assigned for this user.
- *C

  The C locale is assigned for this user (same result as using *POSIX).
- *POSIX

  The POSIX locale path name is assigned for this user.
- locale path name

  The path name of the locale to be assigned for this user.

SETJOBATR
The parameter value that sets job attributes at job start time. The default is *NONE. If *SYSVAL is specified, then the attributes are set from the QSETJOBATR value. The same attributes (*CCSID, *TIMSEP, *DATFMT, *DATSEP, *DECFMT, *SRTSEQ) that can be specified on the system value QSETJOBATR can be specified on the SETJOBATR parameter of the user profile.

If you want all users on the system to use locales, setting system values accomplishes this. Alternatively, the user profile is an ideal mechanism if you want to provide locale function to a limited or specific group of users.

Related concepts

"Locales" on page 8
A *locale* is an object that can determine how data is processed, printed, and displayed.

"System-supplied locales and recommended CCSIDs" on page 360
The system-supplied locale source definition file members are in the optionally installable library QSYSLOCALE in the QLOCALESRC source file. The source file members are encoded in CCSID 37 and are read only.

"Working with locales" on page 200
Locales are used primarily in ILE-based application programs. Additionally, the Retrieve Locale Information (OPM, QLGRTVLC; ILE, QlgRetrieveLocaleInformation) API retrieves one or all categories of a locale.

Scenarios: Setting up i5/OS with a national language version

These scenarios demonstrate how you can enable multilingual support on the i5/OS operating system.

Note: For more information about the details described in the scenarios, see the following topics:

- For Unicode database information relating to DDS, see DDS for physical and logical files.
- For Unicode display information with DDS, see DDS for display files.
- For Unicode printing information with DDS, see DDS for printer files.
- For information about using the subsystem description, see Enabling the secondary language.

Related concepts

"Working with Unicode" on page 111
Unicode is a standard that precisely defines a character set as well as a small number of encodings for it. It enables you to handle text in any language efficiently. It allows a single application to work for a global audience.

Scenario: A single system supporting Spanish

In this scenario, a single system supports Spanish users and applications.

The primary language of the system is Spanish (NLV 2931). Because 2931 is the primary language, the default system settings and i5/OS localization preference are set to Spanish.

The user has also created a database file where the fields of interest are defined to contain Unicode, because they plan to use this same database file for both 5250 applications and Java applications.

The following example shows the SQL statement used to create a database containing a Unicode field named PART_NAME and a non-Unicode field named STOCK_NUMBER:

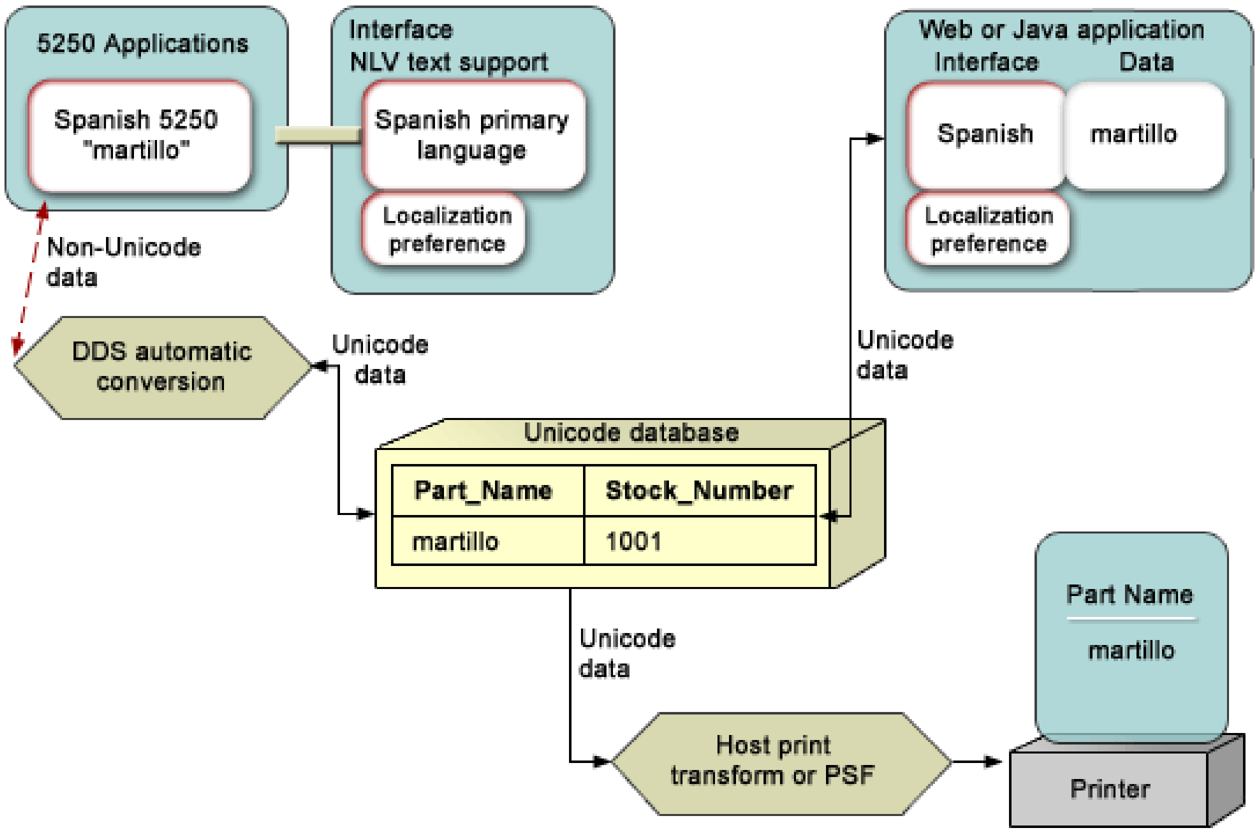
```
CREATE TABLE SAMPLE (PART_NAME GRAPHIC (10) CCSID 1200
   NOT NULL WITH DEFAULT, STOCK_NUMBER INT NOT NULL WITH DEFAULT 0)
```

If the user wants to display this data with a web service or Unicode enabled application, then Unicode is the natural encoding for web use, and no conversion is needed. To get the correct localization preference for the Java application, the user sets the Java locale to `sp_SP` for Spanish in Spain.

If the user wants to display this data with a 5250 session, then the Unicode field must be converted to the CCSID of the display device. The user only has to set the user profiles's CCSID value to 284 to tell the system that this user is on a Spanish display. This service is provided automatically by the system if requested with the CCSID keyword and the *CONVERT parameter in DDS.

To print the Unicode data, the user specifies the *NOCONVERT parameter of the CCSID keyword, and uses the FONTNAME keyword to specify a TrueType font. The unconverted Unicode data can be printed with PSF or with Host Print Transform.

The following figure illustrates this scenario.



Scenario: A single system supporting Spanish and an existing EBCDIC database

In this scenario, a single system supports Spanish users and applications and an existing EBCDIC database.

The primary language of the system is Spanish (NLV 2931). Because 2931 is the primary language, the default system settings and i5/OS localization preference is set to Spanish.

The user has also created a database file where the fields of interest are defined to contain Unicode, because they plan to use this same database file for both 5250 applications and Java applications. They also have an existing database in which the fields are defined in EBCDIC.

The following example shows the SQL statement used to create the EBCDIC database:

```
CREATE TABLE SAMPLE (PART_NAME CHAR (10) CCSID 284 NOT NULL WITH DEFAULT,
   STOCK_NUMBER INT NOT NULL WITH DEFAULT 0)
```

The following example shows the SQL statement used to create a database containing a Unicode field named PART_NAME and a non-Unicode field named STOCK_NUMBER:

```
CREATE TABLE SAMPLE (PART_NAME GRAPHIC (10) CCSID 1200 NOT NULL WITH DEFAULT,
   STOCK_NUMBER INT NOT NULL WITH DEFAULT 0)
```

When using the Unicode file

If the user wants to display this data with a web service or Unicode enabled application, then Unicode is the natural encoding for web use, and no conversion is needed. To get the correct localization preference for the Java application, the user sets the Java locale to sp_SP for Spanish in Spain.

If the user wants to display this data with a 5250 session, then the Unicode field must be converted to the CCSID of the display device. The user only has to set the user profiles's CCSID value to 284 to tell the system that this user is on a Spanish display. This service is provided automatically by the system if requested with the CCSID keyword and the *CONVERT parameter in DDS.

To print the Unicode data, the user specifies the *NOCONVERT parameter of the CCSID keyword and uses the FONTNAME keyword to specify a TrueType font. The unconverted Unicode data can be printed with PSF or with Host Print Transform.

When using the EBCDIC file

If the user wants to display this data with a web service, then the file first must be converted to Unicode. This can be done with the JDBC connector. To get the correct localization preference for the Java application, the user sets the Java locale to `sp_SP` for Spanish in Spain.

If the user wants to display this data with a 5250 session, EBCDIC is the natural encoding for the 5250 device and no conversion is needed. To print the EBCDIC data, the user sends the data to the printer. Because EBCDIC is the default encoding for the printer, no conversion is needed.

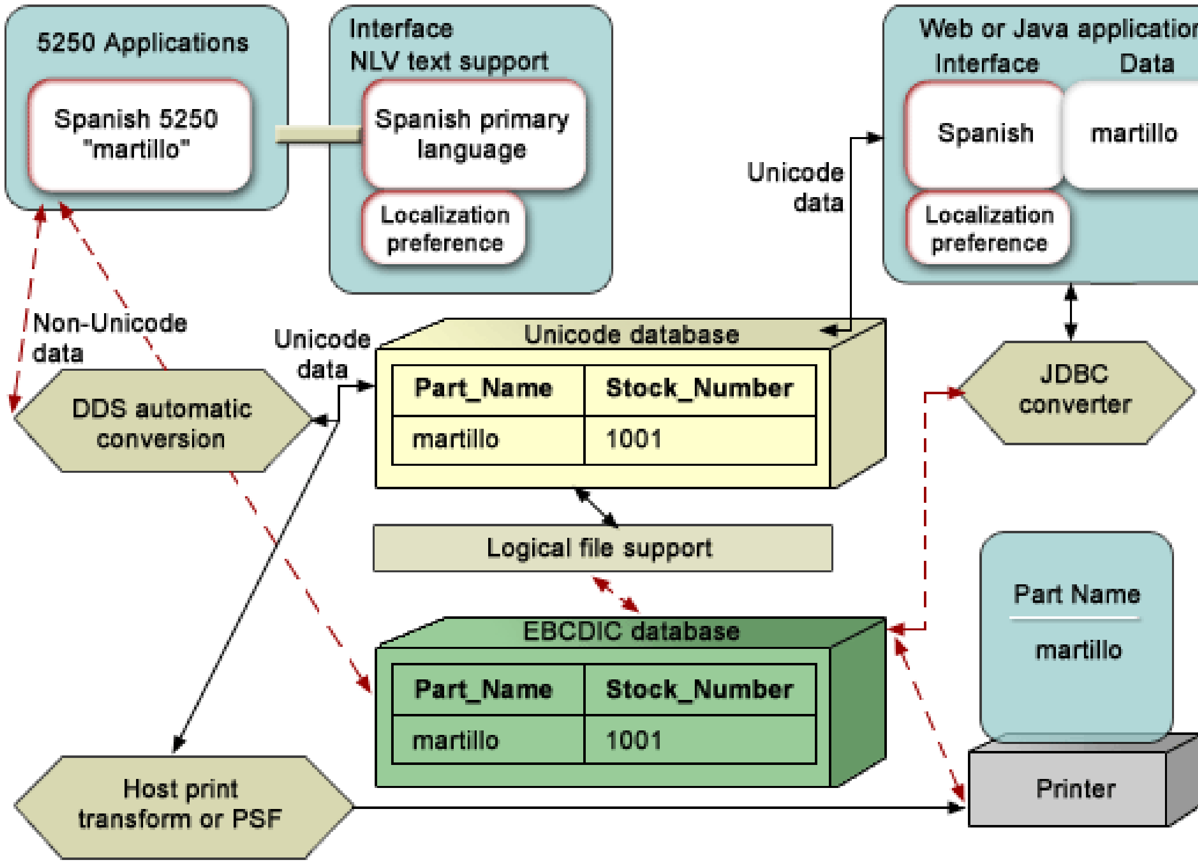
Logical file support

One of the unique features of i5/OS is the ability to use the system's logical file support to have either the EBCDIC file appear to the application as a Unicode file, or to have a Unicode file appear to the application as an EBCDIC file. This might be of use if you want to move your database to Unicode, but do not want to update your existing applications.

If the majority of your application's use of the database involves Unicode, you can have the data stored as Unicode, and create a logical view of the file in EBCDIC. You can then have your EBCDIC programs access this logical file and they do not need to be updated to handle Unicode.

If the majority of your application's use of database involves EBCDIC, you can have the data stored as EBCDIC, and create a logical view of the file in Unicode. You can then have your Unicode programs access this logical file and they do not need to be updated to handle EBCDIC. However, because EBCDIC encodes a smaller set of characters than Unicode does, some character loss might occur.

The following figure illustrates this scenario.



Scenario: A single system supporting English, Japanese, and German

In this scenario, a single system supports English, Japanese, and German users and applications.

The primary language of the system is English (NLV 2924). The system has also been loaded with secondary languages of Japanese (NLV 2962) and German (2929). Because 2924 is the primary language, the default system settings and i5/OS localization preference is set to English. Because these three NLVs are installed, each user can work with the system in English, German, or Japanese.

The users see their language of choice and i5/OS localization preference from the initial sign-on screen by the use of a subsystem description for each secondary language.

The user has also created a database file in which the fields of interest are defined to contain Unicode. Because Unicode provides a unique number for every character on any platform, in any program,and in any language, one field can contain English, German, and Japanese.

The following example shows the SQL statement used to create a database containing a Unicode field named "PART_NAME" and a non-Unicode field named "STOCK_NUMBER":

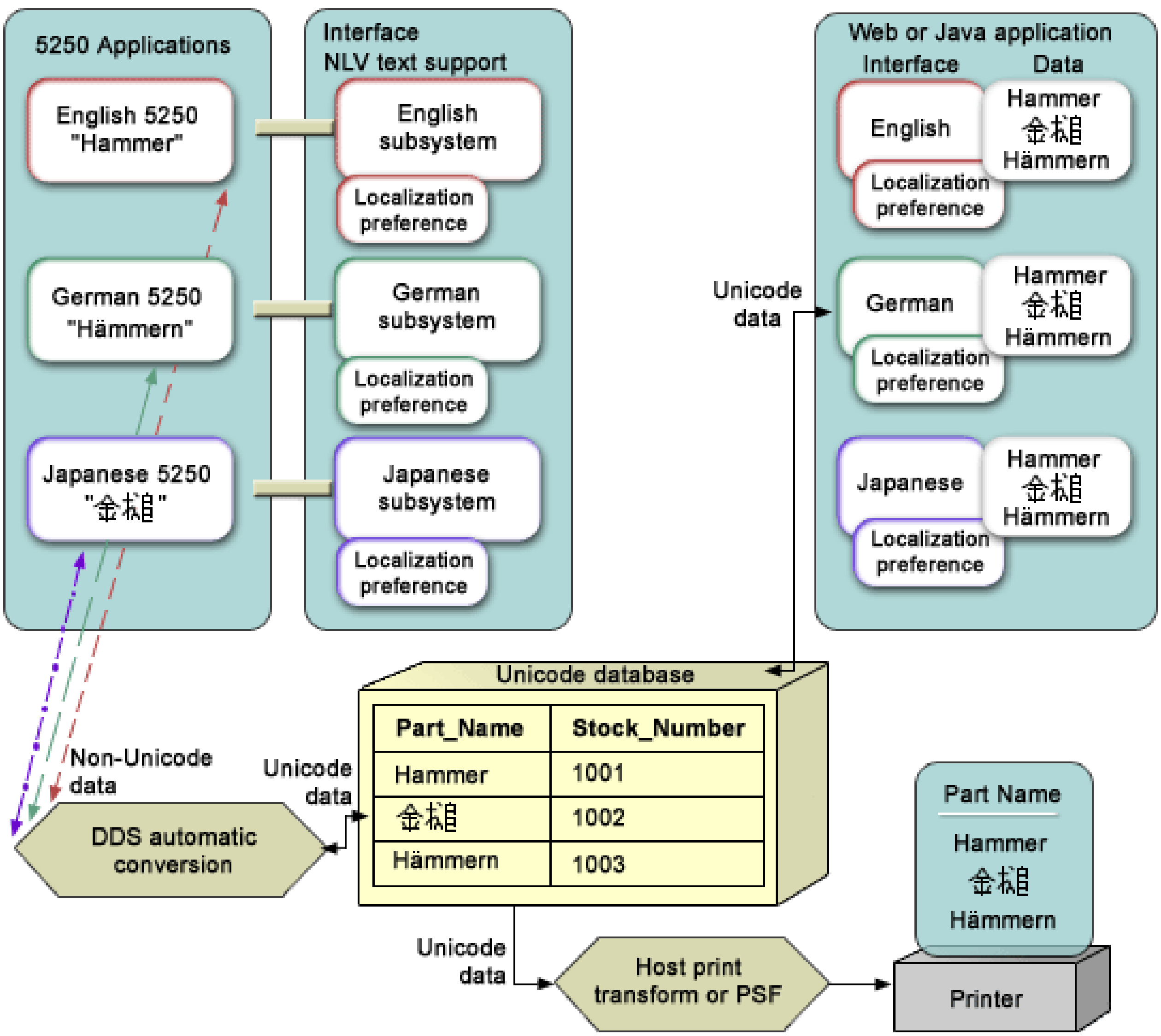
```
CREATE TABLE SAMPLE (PART_NAME GRAPHIC (10) CCSID 1200
   NOT NULL WITH DEFAULT, STOCK_NUMBER INT NOT NULL WITH DEFAULT 0)
```

If the user wants to display this data with a web service or Unicode enabled application, then Unicode is the natural encoding for web use and no conversion is needed. To get the correct localization and interface preference for the Java application, the user needs to set the Java locale to the correct value: `en_US` for English, `Jp_JA` for Japanese, and `de_DE` for German.

If the user wants to display this data with a 5250 session, then the Unicode field must be converted to the CCSID of the display device. The user only has to set the user profile's CCSID value to the correct value (37 for English, 1399 for Japanese, and 278 for German) to tell the system what the user's preference is for the display. This service is provided automatically by the system if requested with the CCSID keyword and the *CONVERT parameter in DDS.

To print the Unicode data, the user specifies the *NOCONVERT parameter of the CCSID keyword and uses the FONTNAME keyword to specify a TrueType font. The unconverted Unicode data can be printed with PSF or with Host Print Transform.

The following figure illustrates this scenario.



Developing globalized applications

Globalized applications are applications that have national language support. National language support allows users to enter, store, process, retrieve, print, and display data in their chosen language. It also allows users to see and enter commands, prompts, messages, and documentation in their chosen language, in formats matching their cultural expectations.

Here are some guidelines for designing, developing, and delivering globalized applications:
- Designing functions that are sensitive to national languages
- Supporting various types of hardware
- Translating the textual data in your application
- Making your application available worldwide

Although your reasons might differ, most internationalized applications are created because:
- The market demands globalized software products that have a local feel
- The application is used in a community that represents multiple cultures
- Revenue opportunities are expanded

Related concepts

"Character processing" on page 7
Character processing on the i5/OS operating system is controlled by specific coding rules and guidelines that ensure consistent processing of character data.

"Handling data in globalized applications" on page 111
The i5/OS operating system enables you to handle data in a globalized environment. This topic collection describes Unicode and Unicode 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.

Goals and processes

Before you invest your time and money in the development of globalized applications, you should set up a planning process to consider how to serve your users well.

Globalization development goals

This information assumes certain goals and provides you with recommendations for developing globalized applications.

The recommendations in this topic assume that your basic goals are:
- To create an application efficiently.
- To create an application at minimal expense. You can retrofit existing applications for globalization and create new applications designed for globalization. Designing an application for globalization, however, is typically less expensive than retrofitting an existing application.
- To ensure that the application design does not interfere with the current or planned design of other internationalized applications.
- When creating an application with national language support, you must plan for or put into effect the following tasks:
  - 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.

Globalization development planning processes

A globalized application should be well planned at every stage in order to save time, effort, and money. You should not have to recompile programs nor repackage data objects. Your product might, however, be required to use a different data object based on the language version you are using. You should have one set of program code and different sets of cultural- and text-dependent code, as needed.

Consider these processes when planning for a globalized application.

Market research process

In the market research process, you must determine for whom you are designing and developing globalized applications. To find the answer, you can ask yourself and your potential customers these types of questions.

What are my target markets for today and tomorrow?

The answer to this question makes a significant difference if you define your market place in different countries or only in the area of your own language, or if you decide to include countries speaking other languages. For example, if you are coding an application from a Latin-based language, application complexity increases when you decide to include countries using non-Latin languages such as Hebrew, Chinese, or Japanese. The application complexity increases because you need to deal with incompatible characters sets and more complex input methods.

Along with the language problem, there are other areas to consider. You need to understand the culture, habits, ways of doing business, and laws of the target markets. You need to understand the customers' ways of life for you to be accepted as a business partner, to be able to get into the market, and to support them in their countries.

These factors can affect:
- The skills that you need (technical, cultural, language, laws)
- The environments to consider
- Your company structure and support organization
- Your relationship to other companies
- The resources that you need (people, time, and money)

Who are the users of my application?

You must understand the requirements that future users of your application will have. For example, do they want to:
- Work with separate databases for different languages?
- Work with a shared database for all languages?
- Exchange or consolidate data?
- Work with different languages dependent on the user, the company, or the company's customers?
- Use end-user database tools to do their own inquiries on the application database?

All these factors affect the design you choose, the way your application is able to switch from one language environment to another, and how data presentation and conversion take place.

How much globalization support is needed?

After you understand the requirements for your customers and their end users, you can decide what kind of cultural-sensitive information you need to store and maintain, the type of data presentation, which parts you need to translate, and how your application must be able to be integrated in the different environments.

What is the cost of the effort?

To estimate the expected revenue, analyze the places you have chosen as your target market. After you know the requirements, you should be able to determine the effort and costs. This amount allows you to compare the costs against the expected revenue.

Which costs more, enabling or retrofitting an application?

The initial cost of enabling an application for national language support might be higher. But consider that the enabling steps are based more on normal modular and data-driven design techniques, which improve the quality of your application even without NLS enabling. Because a good design helps people to understand and describe the application system, you will receive a certain return on the investment. A good design helps to improve productivity of development and maintenance. You have the additional effort of designing and implementing the application only once, even for many different language versions. Compared to retrofitting an existing application, it is much less expensive to plan and design it from the very beginning.

Development process

Before you start to develop NLS-enabled applications, you need to consider initial education and the implementation of internationalized applications.

Education for developing internationalized applications

When you intend to develop NLS-enabled applications, you need to consider additional initial education. The following topics are important to learn about:

- General globalization concepts
- Available globalization support on the i5/OS operating system
- Available globalization support on other systems and applications with which your application operates
- Isolation of different parts of an application
- Data presentation corresponding to cultural conventions
- Design and coding for textual data parts
- Translation process
- Product and system integration
- Packaging, installation, and setup
- Product support and maintenance

Based on the globalization enabling guidelines, first prepare a prototype application and test the chosen way of implementing the application for your specific environment. Afterward include the globalization enabling guidelines in your general application development processes, guidelines, and standards.

Implementing internationalized applications

When implementing an internationalized application, the most important objective is to produce only one set of running code. You must differentiate consistently between running code and textual data. It is essential that you standardize the chosen approach throughout the whole application. Work with unique and clearly defined naming conventions. To understand and to maintain this information in the application, handle parameters called from a program in a consistent way.

Documentation process

Documentation should provide information for the end users of the application system in their own language. The documentation should also include installation, setup, and customization information for the user, the system operator, and the application system manager.

The user documentation should be textual data that can be easily translated. Whenever possible, combine the online help information and user documentation to reduce the volume of text to translate. Any example displays or print layouts should be produced by the application and included in the documentation.

Translation process

Translating the textual data is a time-consuming process. The textual data should be available to translators early at the development stage, even before the code is stable. When planning for translation, you need to consider translation tools, education, guidelines, instructions, and the glossary as well as physical equipment.

Physical equipment

Each translator should have equipment compatible with the language being translated. The display stations and keyboards should have all the characters needed to translate, and the printers should be able to print the translated text.

Translation tools

Provide the translators with tools that increase productivity and that prevent translation of non-textual application data. When purchasing or developing a translation tool, the following features should be included:

- An editor that provides the ability to show displays that can be seen by the user, and the ability to translate the textual data on the system but still protect the parts of the application that are not textual data. The editor should also include functions such as scan and replace, find, copy, move, and delete.
- A dictionary function to provide consistency of words and phrases throughout the product.
- A validation process to check translation errors that might cause the application to malfunction.
- A merge function that provides the ability to merge the translated text into a new version of the original text. This merge function allows for translating only new text, and saves time and effort.
- A print function for validation purposes.

Translation education

It is important that translators are familiar with the product they are translating and also with the tools they are using. The translation process is not the replacement of one word with another, but the formation of concepts in another language. Knowledge of the product being translated provides more understandable products to the user. Time and resources for educating translators should be planned well in advance.

Translation guidelines and instructions

Translation guidelines and instructions should be provided to ensure correct translation. For example, to translate an error message properly, it is important to know in what context this message is displayed. A note to translators telling them what error caused the message to be displayed also helps.

Translation glossary

To ensure accurate translation, use terminology based on definitions in standard, widely available, dictionaries. If your application uses terms not found in standard dictionaries or terms that are used differently from standard definitions, provide a glossary of non-standard terms to the translators. Avoid using abbreviations and acronyms in your application. If you must use abbreviations or acronyms in your application, define them in the glossary. Remember, abbreviations and acronyms that are obvious in your language might not be obvious in another language.

Testing process

The testing of a globalization-enabled product involves testing the running code, checking the textual data, and integrating the running code and textual data.

1. Test the running code

The running code should be tested in a globalization support environment in order to check all the possible language-dependent combinations. Translators should not test the product functionality.

2. Check the textual data

   The textual data should be tested to check correct translation and consistency throughout the product.

3. Integrate the running code and textual data

   After the textual data and the code have been tested separately, an integration test should be performed to test if the application has taken into account all the globalization-related processing, and that the translation of the textual data has not caused a malfunction in the product.

   If your application will also run on a multinational or multilingual system, a separate test that includes more than one set of textual data should be planned.

Packaging and installation process

You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

- Store the running code and textual data separately.
- Package the textual data so that customers receive only the textual data in the languages that are ordered. (If the textual data for all languages is sent to all customers, it will waste system resources and lead to maintenance problems.)
- Provide comprehensive installation documents (translated to the language of the person installing the product) to avoid unnecessary operator-related problems and also to avoid the wrong impression right at the beginning that the application is not reliable.

  Installation documentation should cover the following topics:

  – What is needed to install and run the application, such as hardware and software requirements.
  – How to install the application, and how to recover when things go wrong.
  – What changes need to be made regarding:
    - Subsystem definitions
    - Device descriptions
    - User profiles
    - System values
    - Library lists
  – What are the application limitations?

Related concepts

"Subsystems" on page 9
A subsystem is a single, predefined operating environment through which the system coordinates the work flow and resource use.

"Device descriptions" on page 27
These control language (CL) command parameters can be used to change cultural and linguistic conventions for some display and printer devices.

"User profile name considerations" on page 36
The user profile name identifies the user to the system. This user profile name is also known as the user ID. It is the name that the user types in the *User* prompt on the Signon display.

"Default system values for national language versions" on page 255
Jobs and functions on the i5/OS operating system use system values as default values.

"Job library list" on page 14
The language used for textual data (displays, messages, printed output, and online help information) is controlled by the library list for the job.

"Delivering globalized applications" on page 110
As you prepare to deliver your globalized application, you should consider how globalization issues might affect the ways that your customers install and use your application.

Application maintenance process

Consider these points when planning for the maintenance of a multilingual application.

- The running code must be maintained separately from the textual data. These separate components must be fully synchronized. A redesign in one component might cause a redesign to be made in another.
- Whenever textual data is changed, be sure that it is incorporated in all the languages to which your textual data was translated. In this way, you can ensure a single maintenance level for the complete product.
- Be sure to test the running code for each textual data change that you distribute.

Designing globalized applications

Your goal in designing international application components is to create components that support national languages independently.

The support of one language should not interfere with the support of another language. The support of one language should not force any reduction in the function of the product for another language.

Your application should be able to support multiple languages simultaneously. For example, support for a double-byte coded character set (DBCS) language should not exclude support for single-byte coded character set (SBCS) languages. When you set up your libraries, consider using multiple textual data libraries, which can be dynamically allocated for testing, packaging, and delivery.

As you develop a globalized application for the i5/OS operating system, you must consider these and other unique design considerations that will affect the way you build and code your application.

Related concepts

"Developing applications that process DBCS data" on page 177
You should design your application programs for processing double-byte data in the same way you design application programs for processing alphanumeric data.

Checklist: Application design

The checklist provides some guidelines that you can follow when you create an application with national language support.

Complies	Not applicable	Rule
		The existence of a specific character set within a system or its components must not be assumed.
		Converting character case must be definable for each language and code page.
		Folding must be definable for each language and code page. Folding is the process in which characters that can be printed or displayed are substituted for those that cannot be printed or displayed on a particular device.
		The use of a graphic character for software control purposes must not preclude the use of the same character in the text of messages, menus, prompts, input fields, or output fields.
		The set of characters allowed for use in the entry of data must be definable by the system operator, a user, or an application.
		Graphic symbols and icons must be translatable.
		All characters on the active code page must be accessible.
		Language-dependent parts of a product must be isolated from non-language-dependent parts for easy modification.

Complies	Not applicable	Rule
		The design of a product must allow for the national language support of the various components of the product to be independent of each other.
		National language exits must be provided at strategic points.
		Diagnostics must be enabled.
		Logical layouts different from a given physical keyboard layout must be available to the user.
		All user interface text and presentation control information must be isolated from the running code.
		Functions dependent on display field length and display field position, or display field position alone, must not be designed in such a way that they are affected by user-interface text expansion.
		A method must be provided to allow for the identification and tracking of panels and messages during the translation process.
		Variables must be permitted to assume any location and order within a display field.
		Messages and other displayed words or phrases must be complete entities and must not be constructed from individual words or phrases.
		Entry of end-user commands, keywords, or responses must be possible without regard to uppercase or lowercase characters.
		A product with national language-dependent functions must be designed to facilitate the addition of other countries or national languages.
		Lowercase alphabets should not be assumed to be invariant.
		Character sets should be definable by the operator, a user, or an application.
		Special characters, including punctuation marks, should be definable and not program dependent.
		User-interface text modules should be packaged separately from the running code.

Globalization and localization

National language support enables users to interact with the i5/OS operating system in the language of their choice, with results that are culturally acceptable. National language support consists of two parts: globalization and localization.

The i5/OS operating system controls the operation of programs and provides services such as controlling resources, scheduling jobs, controlling input and output, and managing data. It is designed to complement and extend the capabilities of the system to provide fully integrated support for interactive and batch applications.

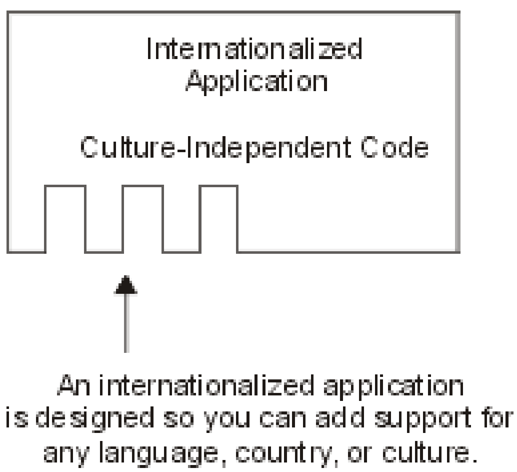
Many functions of the operating system apply directly to interactive data processing. Some of the functions are listed as follows:

- Database support to make up-to-date business data available for rapid retrieval from any workstation
- Work management support to schedule the processing of requests from all work station users
- Application development support that allows online development and testing of new application programs to run at the same time as normal production activities
- System operation support that allows the user responsible for system operations to perform work from the display station using a single control language, complete with prompting and help for all commands

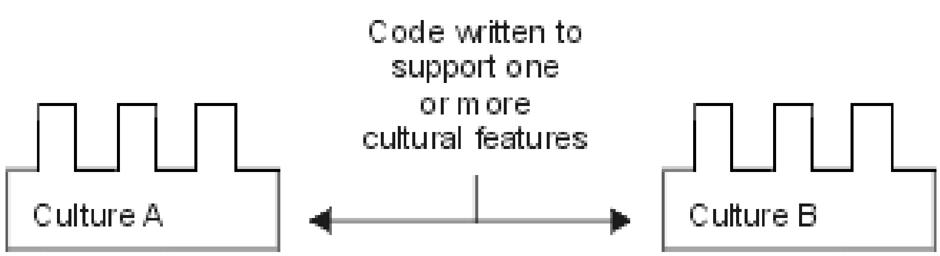
- Help and index search support that allows users to request online information about a wide variety of topics
- Message handling support that allows communication among the system, the user responsible for systems operations, workstation users, and programs running in the system
- Security support to protect data and other system resources from unauthorized access

In addition to these functions, the operating system provides national language support. National language support allows users to interact with the system in the language of their choice, with results that are culturally acceptable. National language support consists of two parts: globalization and localization.

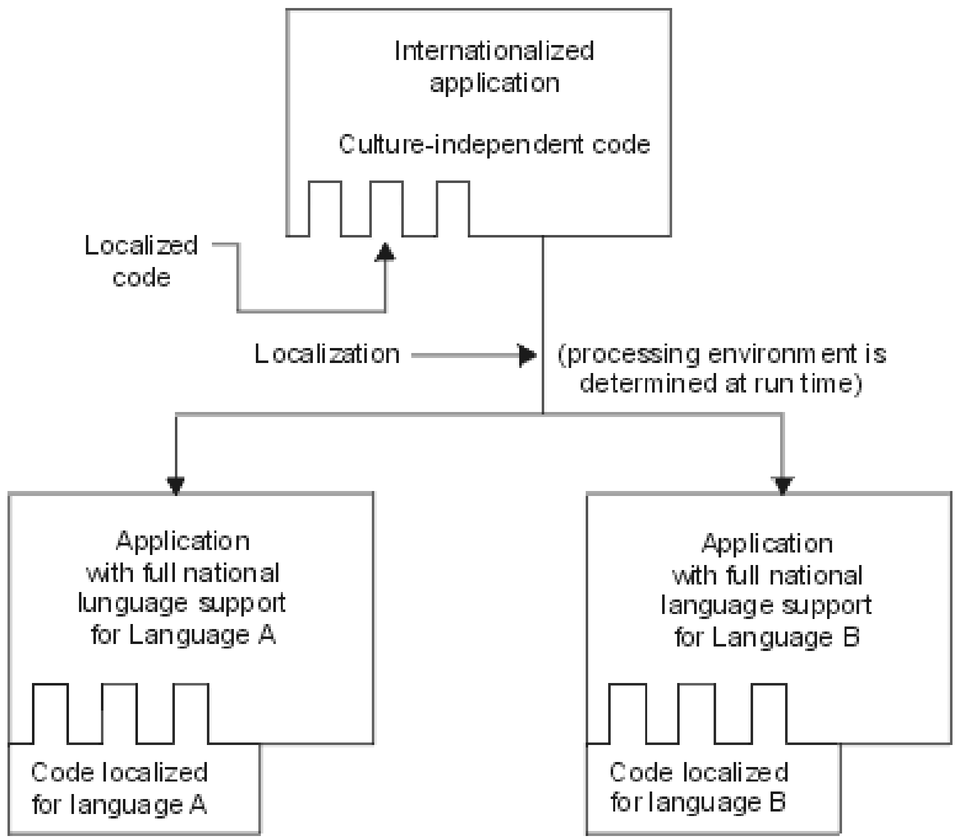
Globalization allows an application to operate in all language environments without any change to the application. This type of design is also known as enabling an application for national language support. A globalized application, shown in the following figure, is culturally neutral.



By contrast, localization allows an application to operate in a specific language, country, or culture. Localization of an application goes a step beyond globalization of the application, as shown in the following figure.



When localized code is integrated with globalized code at run time, the resulting application appears to the user with full national language support. The processing environment defines which localization code is combined with the globalized code at run time, as shown in the following figure.



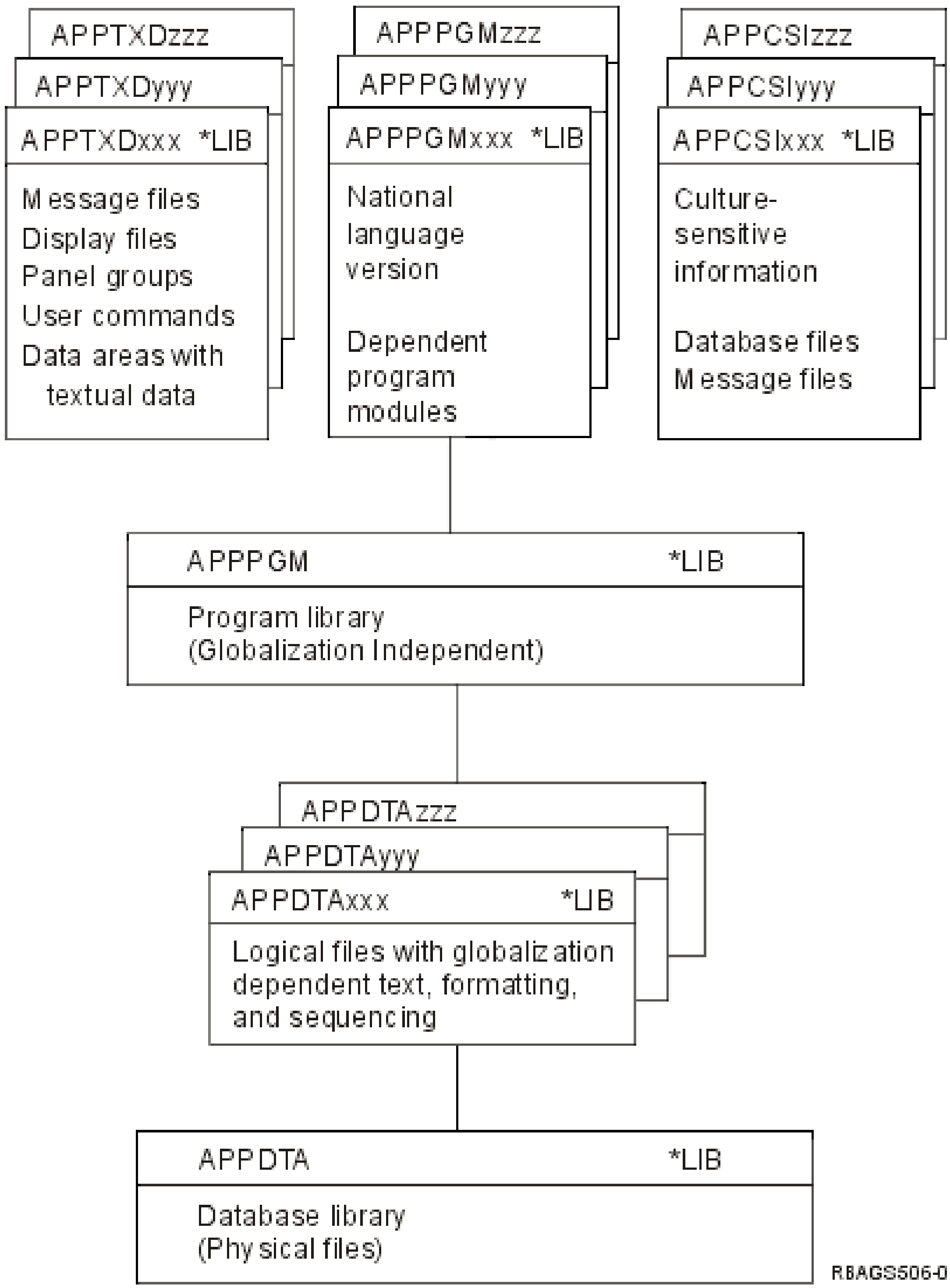
Application arrangement and architecture

When you design an international application, consider organizing and structuring your application in ways that enable it to be used in an international environment.

In particular, consider the following strategies:

- Separate program modules at appropriate places
- Name application parts appropriately for a multilingual environment
- Refer to specifications whenever possible
- Provide multiple sets of logical files in separate libraries when working with database definitions

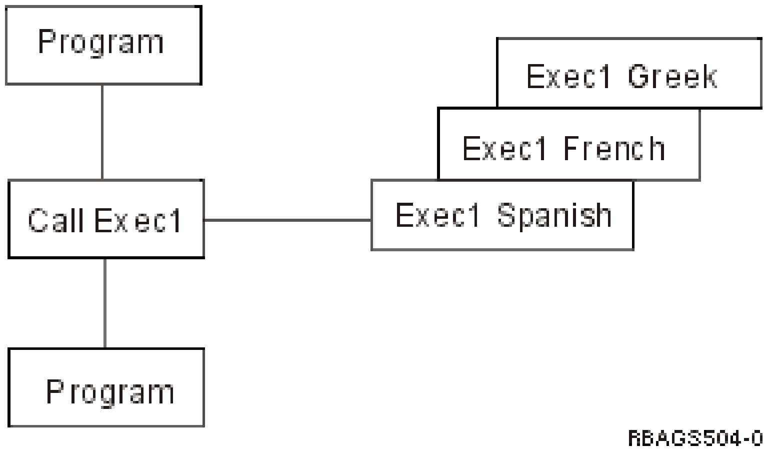
The following figure shows you the recommended way to organize the parts of your application.



Program module separation:

You can separate cultural-dependent parts from your running code and set up cultural-dependent environments. You can do this using system values, user profile attributes, job attributes, and object attributes.

When it is impossible to separate national language and cultural-dependent parts from the running code, you must provide national language exits or calls at all points where functions dependent on national language support are required. The following figure shows a national language exit.



Application part names:

When you want to enable your application for different languages and countries, consider the environments of the target systems in your naming conventions.

Use characters that are available, can be displayed, and can be printed in all the target environments. Use only characters of the invariant character set whenever you specify names for:

- Libraries
- Database files
- Device files (display or printer)
- Help panels
- Message files
- User commands
- Programs
- Record formats
- Fields

All other characters either vary their meaning or might not be available on the keyboard.

To create an internationalized application, you need to divide your application objects into related parts that are textual data and nontextual data. Your naming conventions should be able to distinguish between these parts. You should also be able to distinguish between the textual data of different languages. You can do this by separating the objects into different libraries.

Scenario: Library naming convention

Your library naming convention can look like this:

```
AAATTTLL
```

where: **AAA** is the application identification; **TTT** is the type of objects; and **LLL** is the language code.

This naming convention allows you to have all libraries that belong to an application grouped together because you have a unique identifier (AAA) at the beginning.

The second part (TTT) allows you to distinguish between different types of objects:

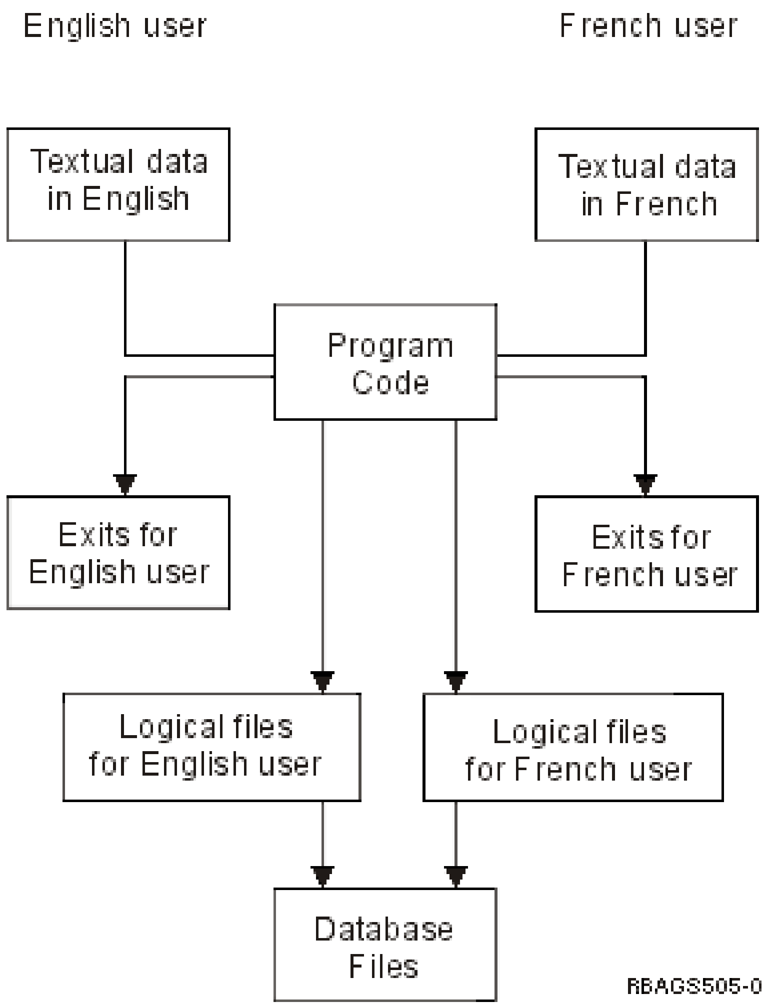
Textual data
- Display files
- Printer files
- Message files
- Help panels
- User command
- Cultural values
- Database files with NLS-sensitive information and specifications
- NLS-dependent program modules

Nontextual data
Programs

Data Database files

The third part (LLL) allows you to specify the national language version for all the textual data parts. This allows you to use the same names for objects of the different national language versions within the different libraries. Your program is able to use different objects by just rearranging the library list accordingly when the job is run.

The initial library list can be taken from the job description. You can build a new library list by specifying the library list in the INLLIBL parameter of the Create Job Description (CRTJOBD) command for a new job description, or of the Change Job Description (CHGJOBD) command for an existing job description. The following figure shows an example of this.



Related concepts

"Invariant character set" on page 324
An *invariant character set* is a character set, such as the syntactic character set, whose code point assignments do not change from code page to code page. The table illustrates the invariant character set (character set 00640) on the i5/OS operating system.

"Database definitions" on page 64
You can define a file to specify certain facts. The specifications are then used in database files.

Specification references:

You should define all your fields first in the field reference file of your application and refer to them wherever you can, in the database specifications, in device file specifications, and in the high-level language programs. This technique helps you to define the field specifications once and use them again.

If you need to distinguish between the same field of different sources, you can rename or qualify them. Whenever you need to change the definition of a specific field, you just need to change the attributes of that field in the field reference file and create the objects again. Then the changes take place automatically in all the different places where the field is used.

For example:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
     A                                      REF(field-ref-file-name)
     A          R record
     A field    R line  pos
or
     A field    R line  pos                 REFFLD(ref-field-name)
```

Database definitions:

You can define a file to specify certain facts. The specifications are then used in database files.

Here are some examples of such specifications:
- The object description text of the file
- The explanation text (TEXT keyword) on record formats and field descriptions
- The column headings (COLHDG keyword) on field descriptions
- Date and time formats and separators
- Sort sequence
- Language identifier

The object description text is shown by many database tools, such as DB2® for i5/OS SQL, System i Access, and data file utility (DFU), on the file selection display.

The column headings are shown by the database tools on the output field definition display. Column headings are also used on screen design aid (SDA) and report layout utility (RLU) as the proposed field-prompting text or heading.

Data management handles date- and time-type fields in the format specified at file-creation time, unless your application or database tool does a conversion to present it according to your request or job demand.

When you want to present all this information according to the language and culture of the user, you need to provide multiple sets of logical files in separate libraries. Along with the translated text, you can specify different date and time formats or different sort sequence and let data management perform the conversion. A similar technique can also be used for numeric-type date fields (unless they are packed), using the substring (SST) function. The user can access the data only through the designated logical views. When you are defining logical files with different sort sequences, avoid using a unique index with a shared-weight table. Although this is possible, a unique index prevents using keys that differ only in characters with the same weight.

The information about Application part names includes a scenario that uses different sets of logical files for different users.

Related concepts

"Application part names" on page 61
When you want to enable your application for different languages and countries, consider the environments of the target systems in your naming conventions.

User interfaces

A user interface is the part of a software product that your customer actually sees.

A user interface may include the layout of display screens or printed output, displayed or printed text, commands, online help, and messages. A user interface is also the part of a software product that you must either translate or make cultural changes to for users in other countries or cultures.

The operating system provides specific software functions to help you organize text from your user interface and store that text in a library for easy translation. The operating system also provides you with a user interface manager that provides a consistent user interface. The user interface manager provides comprehensive support for defining and running panels such as displays and online help.

This section provides guidelines that you can follow when designing a user interface for an international application. You should apply these guidelines early in the design process.

Related concepts

"Delivering globalized applications" on page 110
As you prepare to deliver your globalized application, you should consider how globalization issues might affect the ways that your customers install and use your application.

Checklist: User interface design:

When creating a user interface with globalized support, you should follow some rules and guidelines.

The rules and guidelines are shown in the following table:

Complies	Not applicable	Rule
		The use of a graphic character for software control purposes must not preclude the use of the same character in the text of messages, menus, prompts, input fields, or output fields.
		Graphic symbols and icons must be translatable.
		Language-dependent parts of a product must be isolated from nonlanguage-dependent parts for easy modification.
		All user interface text and presentation control information must be isolated from the running code.
		Sufficient space must be available for user-interface text expansion caused by translation.
		Functions dependent on display field length and display field position, or display field position alone, must not be designed in such a way that they are affected by user-interface text expansion.
		A method must be provided to allow for the identification and tracking of panels and messages during the translation process.
		Variables must be permitted to assume any location and order within a display field.
		Messages and other displayed words or phrases must be complete entities and must not be constructed from individual words or phrases.
		Entry of end-user commands, keywords, or responses must be possible without regard to uppercase or lowercase characters.
		Date and time formats must be selectable.
		Numeric punctuation must be selectable.
		Number rounding and mathematical formats must be selectable.
		Monetary format must be definable.
		The default currency symbol and its abbreviations must be selectable.
		The currency symbol position must be selectable.
		Field sizes for monetary values must be selectable.
		The measurement system must be selectable.
		Lowercase alphabets should not be assumed to be invariant.
		Special characters, including punctuation marks, should be definable and not program dependent.

Complies	Not applicable	Rule
		User-interface text modules should be packaged separately from the running code.
		User-interface text modules for single-byte coded character set systems should be loaded separately from the running code.
		A consistent convention should be used throughout the product for denoting variables and input fields.
		Words should not be used in place of numbers.
		The terminology in user interface text should be consistent throughout a product.
		Abbreviations should be avoided.
		Slang, jargon, and humor should not be used.
		Trademarks should be identified and explained.
		Ambiguous words should not be used.
		Proper style and sentence structure should be used in user interface text.
		Negative questions should be avoided.

Text translation design:

These general tips help simplify the translation of your textual material.

Isolating textual data from running code

To allow easier translation and to avoid translating the running code, you should separate all textual data from the running code. Only one set of running code is needed, but many translations of the textual data can be done.

Providing expansion space

The space needed to translate text from one language to another varies by language. To ensure that the translated version preserves the concept and keeps usability, allow sufficient presentation space for the textual data expansion. The following table shows recommended expansion space for user interfaces designed using U.S. English.

Number of characters in text	Additional space required
Up to 10	100 to 200%
11 to 20	80 to 100%
21 to 30	60 to 80%
31 to 50	40 to 60%
51 to 70	31 to 40%
Over 70	30%

Variable placement of an object on the display

Because the position of one display element often is influenced by the position and size of others, some of the elements on the translated version of a display might need to be relocated. The program must continue to respond properly, despite this relocation.

Flexible order of variables

In order to contain dynamic information, messages typically employ substitution variables. However, each spoken language has its own syntax (order of arrangement of parts of speech). When a message is translated into another language, the position and order of substitution variables might need to change to meet the syntax requirements in the translated language.

Complete textual data entities

If the final form of the constant text relies on the composition of various parts, it might be untranslatable. This is because the translator might not know which form of the word to use or because there is no combination of parts that work for a different language.

For example, you should define column headings for display screens as complete entities. You should not combine words or parts of words to define column headings. Assume that you are writing an application for scheduling jobs between Monday and Friday. You are creating your application in French. You decide to create column headings for reports and screen displays by combining the first part of the name of the day with the constant DI. Throughout the application, the column and report headings are assembled like this:

```
First Part of the
Name of the Day:    Combine With:      Result:

LUN                  DI           LUNDI
MAR                  DI           MARDI
MERCRE               DI           MERCREDI
JEU                  DI           JEUDI
VENDRE               DI           VENDREDI
```

When you translate your application from French to German, you cannot combine two parts to create the names of the days: MONTAG, DIENSTAG, MITTWOCH, DONNERSTAG, and FREITAG.

Treating commands, responses, and keywords like textual data

Commands, responses, and keywords should be translated into the language normally spoken by the user. For example, an English application has been translated into German. If the response is still in English as Yes and No, the German users might feel unfamiliar and uncomfortable in using the program because the responses they are familiar with are Ja and Nein.

Expressing all text as simply and clearly as possible

- Use simple phrases and sentences and avoid compound phrases. Simple words allow easy translation.
- Make terminology consistent throughout the product.

  If consistent terminology is not being adopted throughout the product, translators will waste time trying to determine the appropriate word to be used in translation.
- Include notes to translators in your information for correct word use to prevent any misunderstandings.
- Avoid abbreviations.

  Rules for abbreviations vary from language to language. Abbreviations of words can lead to misunderstandings by the translator and by the user.
- Avoid slang, jargon, and humor.

  Slang, jargon, and humor are specific for a particular language and cannot be easily translated into another language.
- Avoid negative questions.

  Negative questions are often misunderstood by the user. When asking questions, ask them in a positive way.

Textual data code design:

You can use different techniques to specify, store, and use constant text. You can use each technique for specific types of textual data components. Each technique has its advantages and disadvantages.

Application displays, printer file specifications, and user-created commands typically contain a large amount of constant text. Application displays, printer file specifications, and user-created commands also contain input and output fields such as headings, field prompts, instruction lines, and function key descriptions.

Related concepts

"Constant text strings" on page 88
When designing your panels, keep in mind that different languages have different space needs for the same expression.

"Printer file design and translation" on page 92
Program-described printer files and externally described printer files are two types of printer files. When you design printer files to be translated into a national language version, you should follow some specific guidelines.

Early message binding:

Text can be stored externally from the source code in a separate message file but is bound into the object when it is created.

This technique can be used for:

Display files
Constants such as titles, instruction lines, option definitions, headings, field prompts, command key descriptions

Printer files
Constants such as titles, headings, total line descriptions

User commands
Prompt descriptions on the command definition statements

For device files (display and printer), the message is referred to by the Message Constant (MSGCON) keyword in the DDS source specifications.

For example:

```
A         line pos   MSGCON(length message-ID[*libl/]message-file-name)
                                  ^
                            includes expansion space
```

For user commands, the message identifier *xxxnnnn* is specified on the PROMPT keyword instead of a literal. The message file is referred to on the Create Command (CRTCMD) command.

For example:

```
CMD      PROMPT(xxxnnnn)
```

The message file name *message-file-name* is in a source file referred to by the following command.

```
CRTCMD CMD(command-name)  PGM(library-name/program-name)  +
     PMTFILE([*libl/]message-file-name)
```

Before the object can be created, you must enter the message description into the specified message file. Enter the message description using the Add Message Description (ADDMSGD) command.

For example:

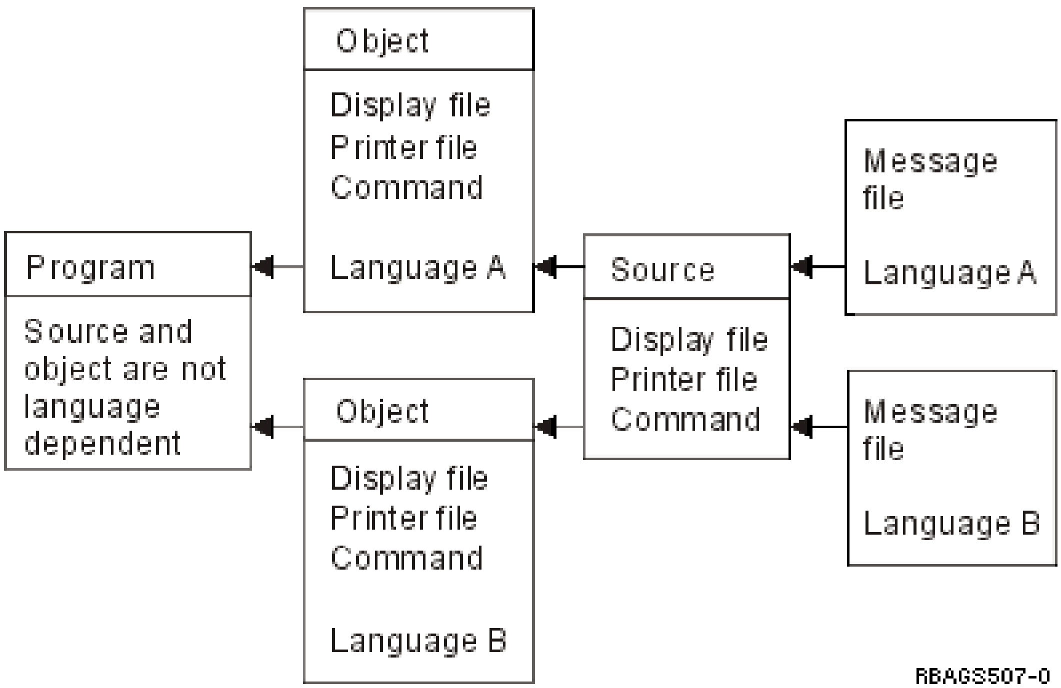
```
ADDMSGD    MSGID(xxxnnnn)  MSGF(library-name/message-file-name) +
MSG('Text                                    ')
```

where *xxxnnnn* is the message identifier.

This technique allows you to create any number of objects in different languages and to put them into different libraries using the same source code by just assigning another message file at object creation time.

The message file is needed only during the creation of the object. Consider specifying the appropriate length for different languages on the MSGCON keyword. Then make the length information available to the translator.

The following figure shows how early message binding works:



At file creation time, you can choose the appropriate textual data of the language version you want to work with by setting up the library list with the specific library containing the textual data and the program library.

Related concepts

MSGCON (Message Constant) keyword for display files

MSGCON (Message Constant) keyword in printer files

Related reference

Create Command (CRTCMD) command

Add Message Description (ADDMSGD) command

Late message binding:

Text can be stored externally from the DDS source code in a message description and is bound only to the display format at run time.

This technique can be used for:

Display files only

Constants such as titles, instruction lines, option definitions, headings, field prompts, command key descriptions (MSGID keyword)

Default values on input fields (MSGID keyword)

Field validation specifications (CHKMSGID keyword)

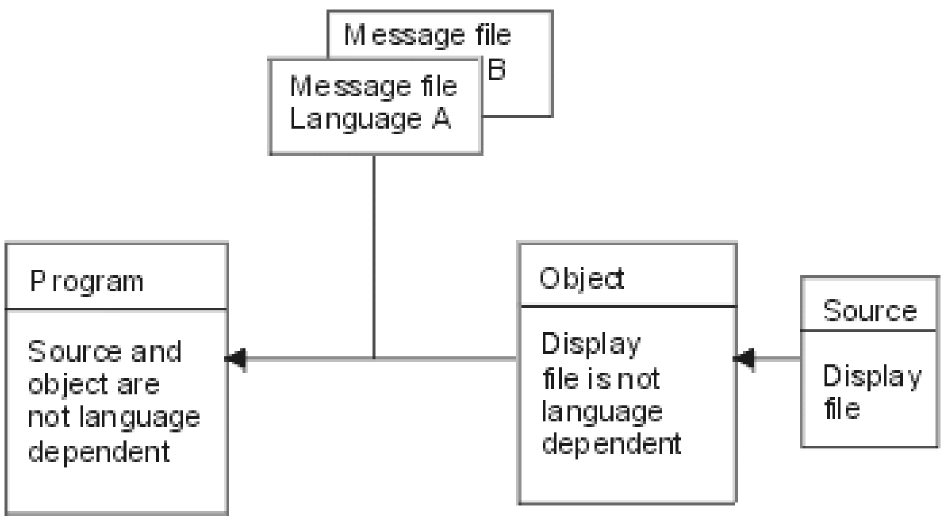
Error messages (ERRMSGID and SFLMSGID keywords)

In the DDS for the display file, the message is specified through the MSGID (Message Identifier) keyword. The message has to be entered into the specified message file using the ADDMSGD (Add Message Description) command.

For example:

```
A    FLD-name  length  line pos    MSGID(message-ID [*libl/]message-filename)
                  ^
           includes expansion space
ADDMSGD   MSGID(xxxnnnn) MSGF(library-name/message-file-name)  +
MSG('Text                                    ')
```

This technique allows you to create any number of message files in different languages and different libraries, with one DDS source code and display file object. During run time, you assign another message file by setting the library list accordingly. The following figure is an example.



Note: This technique requires the application to perform all editing based on the cultural convention.

Direct coding as an unnamed output field:

The most common way to define constant text is to specify the text directly in the source code as a literal. While this method is the most common way to define constant text, it is the most difficult to translate. Avoid using this method whenever coding an application, even if the application is not planned for translation.

If you are coding an application that will not be translated, you might want to use this technique for:

Display files

Constants such as titles, instruction lines, option definitions, headings, field prompts, command key descriptions

Default values on input fields (DFT keyword)

Error messages (ERRMSG/SFLMSG keyword)

Printer files
Constants such as titles, headings, total line descriptions

User commands
Prompt descriptions on the command definition statements.

For device files, specify the text as an unnamed field, indicating the starting line and column and the constant text itself.

For example:

```
A               line pos        'Text . . . . . . . . . . . . . . :
'
```

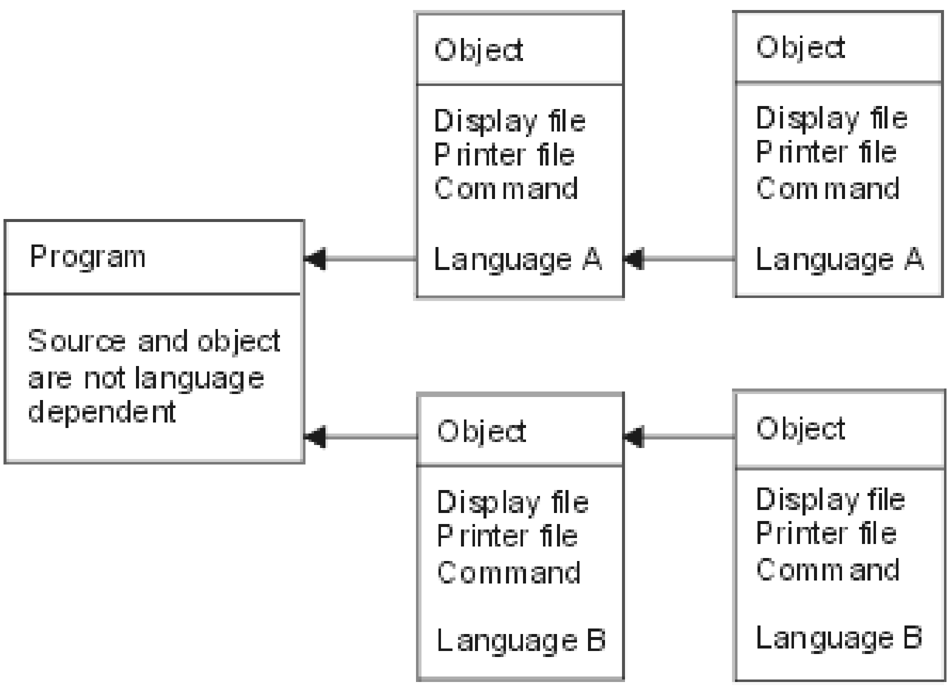
A similar rule applies to user-created commands. Define the text directly on the keywords of your command source statements.

For example:

```
CMD          PROMPT('    Command description       ')
```

When defining the text directly on the keywords, standardize the sizes of the different elements in a large literal, rather than specifying many small single ones as single words. This makes the source code more readable and more flexible for translation.

Consider that the space needed for explanation text can vary from language to language. To have enough room after translation, remember to reserve space initially. The source members need to be translated and the objects need to be created for different languages as shown in the following figure:



Each national language version has one set of programs, but can have multiple sets of source members and data objects. When the application is run, you can choose the appropriate textual data of the language version that you want to work with. This can be done if you set up the system part of the library list with the specific library that contains both the textual data and the program library.

Text stored in database files:

Text can be stored externally from the source code in a database file, retrieved by the application program, and then moved to the display or print format at run time.

Instead of coding constants on the DDS, you can specify output fields that can be filled by the program. Consider specifying the appropriate length for different languages on the output fields and making that available to the translator.

This technique can be used for:

Display files
: All constant text
: Default values on input fields
: Error messages

Printer files
: All constant text

Programs
: All constants like compare values, scan characters, and tables.

This technique allows you to create any number of database files in different languages and different libraries, with only one DDS source code and display file object. During run time, you assign the corresponding database file by setting the library list accordingly.

Note: This technique requires the application to perform all editing based on the cultural convention.

User interface manager:

The i5/OS user interface manager (UIM) is a part of the system that allows you to define panels and dialogs for your application.

UIM provides the following support:
- A tag-based language for describing data and panels.
- A compiler to create panel group objects and menu objects by using the tag-based language.
- A set of application programming interfaces (APIs) to use as panel group objects to display and print panels.

The UIM also provides the following functions:
- Dialog commands for screen management
- Contextual online help
- Pop-up windows
- Menu bars
- Command line for entering CL commands
- Tailoring of the contents of a panel for different users or environments
- Fast paths through menu networks
- Double-byte character set (DBCS) languages
- Bidirectional (BIDI) language support

UIM supports common panel types, such as menus, information displays, list displays, and entry displays. When all display types and interfaces are consistent, users adapt more quickly to new applications.

UIM applications can coexist with and share the requester display device with other open display files that are not under UIM control. However, a UIM panel and a DDS-defined record format cannot appear on the display at the same time. When a UIM panel either replaces a DDS panel or vice versa, the system suspends operations of one file or panel group and restores the display as needed.

Online help design:

You can define online help by using panel groups or records. By using panel groups, you can define online help as objects into which user interface manager (UIM) source is entered. By using records, you can define online help as a set of DDS keywords contained in a source file member.

If the user interface manager is used for defining online help, the panel groups are defined either in place of DDS or in the display file. In either case, the encoding of the data to be displayed must be indicated by the CHRID value in the display file or the panel group. A panel group is an object that can be used to contain help information. The operating system uses *PNLGRP as an identifier for the object type that contains a collection of help information.

Guidelines: Online help

When defining online help information to be translated into national language versions, keep in mind the following considerations about panel groups and records:
- Records do not have word processing available (functions such as spell check and word wrap though system APIs exist to provide spell checking).
- Various i5/OS messages and panel groups determine the national language conventions and translations. Not all countries have a national language version available for the operating system. Not all national language versions are completely translated, with many parts still in English. The messages and panel groups that are not translated do not reflect the national language cultural conventions. The command design information includes an example of a translated panel in which part of the panel has remained in English because not all parts of the NLV were translated.
- Allow for translation expansion.

Guidelines: DDS online help design

When multiple languages are installed on one system, the help documents are stored in different folders. The DDS source file needs to be copied, changed, and compiled again for each language on the system.

Related concepts

"Command design" on page 78
The i5/OS operating system allows users to define and create their own commands.

Index search tags:

Help panel groups can contain index search modules. Index search supplements the help information that is provided for each display.

To use the information in help panel groups for the index search function, you need to add the appropriate UIM tags to your help modules.

Users can access the index search function from any display help that specifies that the index search function is available.

The ISCH tag

The ISCH tag defines the title of a topic in the index and specifies the root words that serve as the link between the topic and the search words (synonyms) entered by the user. The tag appears immediately after the HELP tag to which it refers. There can only be one ISCH tag within a single help module.

For each ISCH tag, there can be several lines of root words, provided that the total number of root words is no more than 50. If more than one line of root words is used, ROOTS= must be repeated at the beginning of the second line and subsequent lines.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
:PNLGRP.
:HELP name=entry1.
:ISCH ROOTS='root1 root2 root3 root4 root5'
 ROOTS='root6 root7 root8 root9 root10'
 ROOTS='root11 root12 root13 ... root50'.
Title of First Topic

This is the first index search module in this panel group.
:EHELP.
:EPNLGRP.
```

The root words on all lines must be enclosed in apostrophes and a period must be placed only at the end of the last line of root words. The topic title follows the period on the ISCH tag and can be placed on the line immediately following the period.

The ISCHSYN tag

The ISCHSYN tag defines the words (synonyms) that, if entered by a user, match a specific root word. If a word that is entered by a user is a synonym for a root word, then a match is found for each topic whose ISCH tag contains that root.

If you want a word that is used as a root word to be used as a synonym as well, you must include the word as a synonym on the ISCHSYN tag. For example:

```
:ISCHSYN ROOT='ocean'.ocean water sea
```

The synonyms for the ISCHSYN tag must be entered on one line, and at least one ISCHSYN tag must exist for each root word. If more than one line is needed, more ISCHSYN tags can be entered for the same root word.

UIM does not differentiate between synonyms entered in uppercase, lowercase, or mixed case. For this reason, it is not necessary to repeat synonyms to cover all the different cases.

You can use alphabetic or numeric characters for synonyms; however, the following characters (including their hexadecimal equivalents) are not allowed to be used as a synonym or part of a synonym:
- . (period)
- ((left parenthesis)
-) (right parenthesis)
- ; (semicolon)
- , (comma)
- ? (question mark)
- : (colon)

The ISCHSYN tags can be placed anywhere in the panel group, but to make maintenance and translation easier, place them all in one area (such as at the beginning of your panel group or in a panel group object that contains only ISCHSYN tags).

Example: ISCH and ISCHSYN usage

The following example shows some ISCHSYN tags and the ISCH tags that use them:

```
:PNLGRP.
:ISCHSYN ROOT='ocean'.ocean water sea
:ISCHSYN ROOT='lake'.lake water pond
:ISCHSYN ROOT='definition'.definition define description what
:ISCHSYN ROOT='definition'.summary concept information explanation
:HELP name='defocean'.
:ISCH ROOTS='definition ocean'.
Definition of ocean

An ocean is one of the five large bodies of salt water, which
together cover nearly three-fourths of the world.
:EHELP.
:HELP name='deflake'.
:ISCH ROOTS='definition lake'.
Definition of lake

A lake is a body of standing water that is enclosed by land.
:EHELP.
:EPNLGRP.
```

Index search and double-byte character set:

The index search function can be used with either double-byte character set (DBCS) or single-byte character set (SBCS) data. When DBCS data is used, the device from which it is requested must be capable of entering and presenting the data in DBCS.

The object that contains the index search data is marked as containing DBCS data. The system determines if the device is capable of handling the DBCS data.

When the data is being prepared for DBCS format and the index search function is used with that data, consider the following information:

- When the index search data is prepared for a DBCS system, the synonyms entered on the ISCHSYN tag must be in double-byte character mode. That is, the first byte after the tag must be a shift-out character and the last byte of the data must be a shift-in character. The system does not convert data on the ISCHSYN tag to double-byte character data.
- Words must be separated by a single-byte blank. From 1 to 19 double-byte characters can be combined to form a word. Intervening shift-out and shift-in characters are allowed, but are ignored by index search.
- The words that are used to link the ISCH and ISCHSYN tags (the ROOTS attribute of the ISCH tag and the ROOT attribute of ISCHSYN tag) must be identical and should not be entered in DBCS.
- Search words can be entered in either single-byte mode or double-byte mode. Single-byte blanks can be entered to separate the words.

When the search words are shown on the screen, the double-byte character representation (the character that was actually used in the search) is shown. Special processing takes place so that index search is not case sensitive. The search words from the ISCHSYN tag are converted to uppercase using a conversion table for the code page that is specified with the TXTCHRID attribute of the PNLGRP tag. If the search words are DBCS, they are not converted to uppercase. Shift-out and shift-in characters are treated as blanks during parsing; leading and trailing blanks are removed. All SBCS words are converted to uppercase using a conversion table for the code page of the device description.

Program message design:

A message can be predefined or immediate.

Consider the following information when you design and code:

- Do not use immediate messages. They are created by the sender or program at the time they are sent and are not stored in a message file. Therefore, they cannot be translated by the translator.

- Use predefined message descriptions that can both:
  - Exist outside of the program that uses them.
  - Be stored in a message file.
- Do not specify the maximum size for a message file. When the message file becomes full, you cannot change the size of the message file. You need to create another message file and add the message description again.

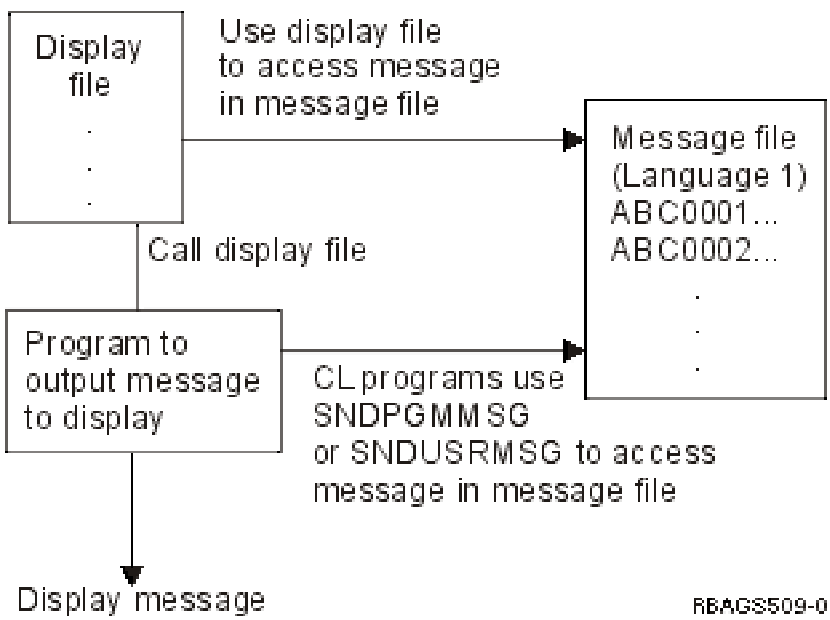
  Use the Create Message File (CRTMSGF) command to create a message file to hold the predefined message description. The contents of the predefined message description can be put into a message file by the Add Message Description (ADDMSGD) command.
- Use substitution variables with care. Different languages have different orders for substitution variables. For example, in the English message:

  ```
  File &1 in Library &2 not found.
  ```

  &1; and &2; are the substitution variables. Those substitution variables can appear in different positions for different languages.
- Make your design and coding able to understand a reply code for different languages. For example,

  ```
  English     Y = Yes
  Danish      J = Ja (means Yes)
  ```

The following figure shows the creation of different NLV messages from message files.



A program can directly access the message file for program messages, or it can indirectly access the message file through display files for program messages.

Related concepts

"CCSID support for messages" on page 136
You can use CCSID support for handling messages and message catalogs on the i5/OS operating system by using commands and application programming interfaces. You can send messages tagged with one CCSID to users with a different CCSID.

Related reference

Create Message File (CRTMSGF) command

Add Message Description (ADDMSGD) command

Control language

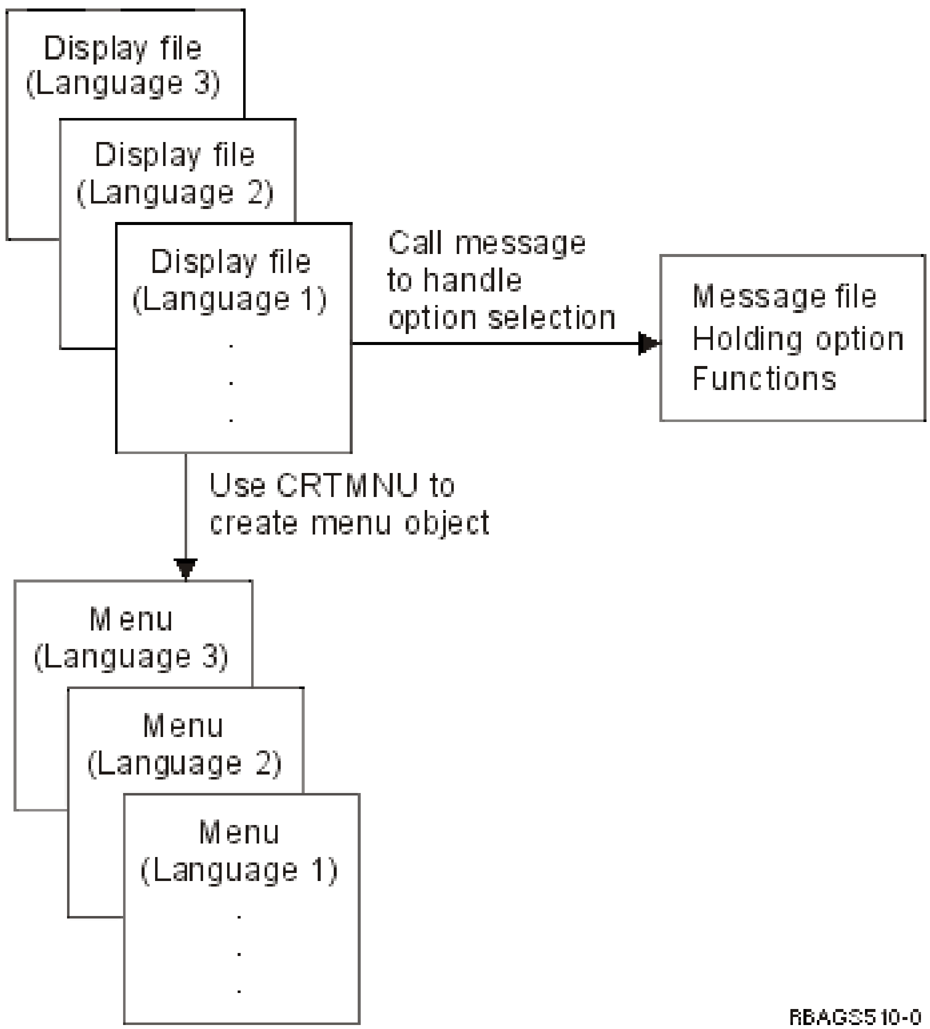
Menu design:

You can define your own menus on the system. The types of user-defined menus include display file menus, UIM (reference) menus, and program menus.

To use an application system, users need to deal with a lot of menus and displays. When an application is being translated from one language to another, a large portion of the literal text to be translated comes from menus.

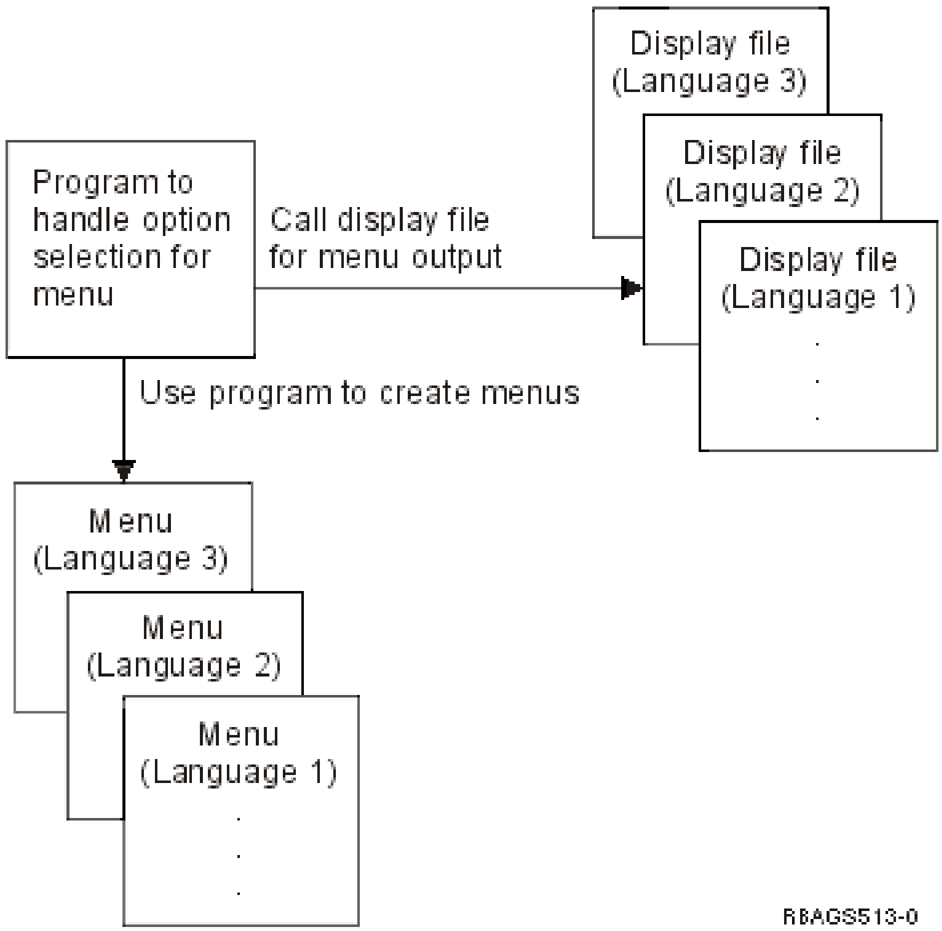
Display file menu

A display file menu uses a display defined by DDS to present a menu format. The menu functions are controlled by a menu object that contains the commands used to run each of the menu options. The following figure shows how display file menus are created for different national language versions.



Program menu

A program menu uses programs to present the menu format (defined by DDS) and to provide functions necessary to run the menu options. The following figure shows how program menus are created in different national language versions.



Menu translation

To allow for easy translation into national language versions of your menus:

- Keep the literal text of menus external by holding the constant text as externally defined message descriptions in a message file and by incorporating the text into a menu file when the program is run.
- Be aware of the expansion space needed when a menu is translated from one language to the next. Leave space for translation expansion when you design your menus.
- Be aware of cultural conventions when date, time, or edited fields are displayed on the menu.
- Use numerals 0 through 9, instead of uppercase and lowercase English letters (A through Z), as the option fields for selection. Numeric characters are more standard among different languages.

Command design:

The i5/OS operating system allows users to define and create their own commands.

To create a command, you must first define the command through command definition statements. Then use the Create Command (CRTCMD) command to process the command definition statements to create the command definition object.

When you define and create a command, take into consideration the following information:

- Use help panel groups to provide online help information for the command.

- Use message identifiers instead of literal text for the PROMPT keyword on the CL CMD, PARM, ELEM, and QUAL command definition statements.
- Translate the text that is displayed to the right of the prompt line of each parameter on the prompt display. This text is specified by the CHOICE parameter of the PARM command definition statements, so the appearance of the command prompt display maintains its coherency.
- Compile command-prompt text into separate command definition object versions for each national language. Use the Change System Library List (CHGSYSLIBL) command before creating the command to get the national language version prompt text from the correct national language version library.
- The function keys of the command prompt display are provided by the operating system. If the NLV of the operating system is different from the NLV of the command, two different languages appear on the command prompt display. For example, when translating an English display into German, both the English and German appear on the command prompt display.

  The Control language information includes additional information about creating and defining commands.

Related concepts

"Online help design" on page 73
You can define online help by using panel groups or records. By using panel groups, you can define online help as objects into which user interface manager (UIM) source is entered. By using records, you can define online help as a set of DDS keywords contained in a source file member.

Related reference

Create Command (CRTCMD) command

Change System Library List (CHGSYSLIBL) command

Control language

Cultural-dependent design:

Different countries might have different standards, which you must consider when developing an NLS-enabled application. This culturally sensitive information must be placed outside the program the same way as the textual data is handled.

Many languages have characters (such as common-usage vowels essential to the correct spelling of a word) outside of the A-Z alphabet that must be considered for collating purposes.

Through system values, the system supplies linguistic support, cultural support, and the ordering of data.

Related concepts

"Default system values for national language versions" on page 255
Jobs and functions on the i5/OS operating system use system values as default values.

"Field editing specifications" on page 89
For the edit specification of your numeric, date, and time fields, you must consider the different cultural conventions of the users.

Database file attributes:

Database attributes, such as coded character set identifier (CCSID), sort sequence (SRTSEQ), and language identifier (LANGID), are cultural dependent.

The CCSID attribute applies only to physical files. The SRTSEQ and LANGID attributes can be used with both physical files and logical files. A logical file can have a CCSID value only when it has taken the CCSID from the physical file. The database attributes are stored with the data. They are static in the sense that they cannot be dynamically altered by the process of accessing the data.

Related concepts

“Coded character set identifier (QCCSID) system value” on page 16
The coded character set identifier (QCCSID) system value specifies the CCSID for the i5/OS operating system.

“Sort sequence (QSRTSEQ) system value” on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

“Language identifier (QLANGID) system value” on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

Job attributes:

Some job attributes are cultural dependent. Through cultural-dependent attributes, the system provides linguistic support, cultural support, and the ordering of data.

- Coded character set identifier (CCSID)
- Sort sequence (SRTSEQ)
- Language identifier (LANGID)
- Country or region identifier (CNTRYID)
- Date format (DATFMT)
- Date separator (DATSEP)
- Decimal format (DECFMT)
- Time separator (TIMSEP)

The default values for CCSID, SRTSEQ, LANGID, and CNTRYID attributes are set from the user profile when the job starts. The values for CCSID, DATFMT, DATSEP, DECFMT, SRTSEQ, and TIMESEP can be set from the LOCALE and SETJOBATR attributes associated with the user profile. When you use the Change Job (CHGJOB) command, you can override the values specified for any of the listed job attributes.

Related concepts

“Coded character set identifier (QCCSID) system value” on page 16
The coded character set identifier (QCCSID) system value specifies the CCSID for the i5/OS operating system.

“Sort sequence (QSRTSEQ) system value” on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

“Language identifier (QLANGID) system value” on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

“Country or region identifier (QCNTRYID) system value” on page 17
The country or region identifier (QCNTRYID) system value indicates the default country or region identifier for the system.

“Date format (QDATFMT) system value” on page 18
The date format (QDATFMT) system value is used for the default value for the DATFMT job attribute. This system value also determines the format in which a date can be specified on the initial program load (IPL) options prompt.

“Date separator (QDATSEP) system value” on page 19
The date separator (QDATSEP) system value is used as the date separator for the default value of the DATSEP job attribute. It is also used as the date separator you can specify on the initial program load (IPL) options prompt.

“Decimal format (QDECFMT) system value” on page 21
The decimal format (QDECFMT) system value determines the type of zero suppression and decimal

point character used by DDS edit codes 1 through 4 and A through M. It also determines the decimal point character for decimal input fields in the interface.

“Time separator (QTIMSEP) system value” on page 26
The time separator (QTIMSEP) system value specifies the character separator for time.

Related reference

Change Job (CHGJOB) command

Program attributes:

The SRTSEQ and LANGID parameters can be specified as program attributes belonging to a *PGM object type.

The LANGID parameter is used together with the SRTSEQ parameter only when the SRTSEQ value is set to *LANGIDUNQ or *LANGIDSHR. Otherwise, the LANGID parameter is not used.

If a program explicitly refers to a sort sequence or a language identifier, then those attributes stored in the program object take effect. The *JOBRUN value for these parameters is used to refer to the attributes of the job running the program. *JOBRUN makes it possible to use a single set of programs processing data according to different sort sequences. The *JOBRUN value affects only the processing of data, however, not the retrieval sequence of data. The retrieval sequence is determined by the database attributes. To retrieve data in a sort sequence different than what is defined in the database, use logical files that are built separately.

Information in message CPX8416:

If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

The system message contains these values:
- Code page and character set
- Currency symbol
- Date format
- Date separator
- Decimal format
- Leap year adjustment
- Coded character set identifier
- Time separator
- Language identifier
- Country or region identifier

Cultural-dependent fields in the panel or display should not contain hard-coded values. These fields should be defined with the maximum length permitted for the field on the display.

If your application is to support users in languages other than the primary language, the callable routines should use the CPX8416 message values. A callable routine uses the cultural values for the primary language to determine the contents of the field (for example, date format) and places these values on the display. NLS system values maintained in message CPX8416 determine the format of the cultural values appearing in the cultural-dependent fields.

Your application can use the details from the system message.

The following table shows the layout for message CPX8416. This example shows the values in the text column using the English uppercase and lowercase NLV (feature 2924).

	Field	Start	Length	Justify
Description value	QCHRID 697 37	0001 0012	10 21	L L
Description value	QCURSYM $	0034 0045	10 01	L L
Description value	QDATFMT MDY	0047 0058	10 03	L L
Description value	QDATSEP /	0062 0073	10 01	L L
Description value	QDECFMT	0075 0086	10 01	L L
Description value	QLEAPADJ 0	0088 0099	10 01	L L
Description value	QCCSID 37	0101 0112	10 05	L L
Description value	QTIMSEP :	0118 0129	10 01	L L
Description value	QLANGID ENU	0131 0142	10 03	L L
Description value	QCNTRYID US	0146 0157	10 02	L L
Description value	QIGCCDEFNT *NONE	0160 0171	10 21	L L

Related concepts

"Configuring secondary languages" on page 39
A secondary language consists of textual data for all licensed programs supported for a national language version.

"Currency symbol (QCURSYM) system value" on page 17
The currency symbol (QCURSYM) system value verifies the currency symbols specified in the DDS keywords Edit Word (EDTWRD) and Edit Code (EDTCDE).

"Date format (QDATFMT) system value" on page 18
The date format (QDATFMT) system value is used for the default value for the DATFMT job attribute. This system value also determines the format in which a date can be specified on the initial program load (IPL) options prompt.

"Date separator (QDATSEP) system value" on page 19
The date separator (QDATSEP) system value is used as the date separator for the default value of the DATSEP job attribute. It is also used as the date separator you can specify on the initial program load (IPL) options prompt.

"Decimal format (QDECFMT) system value" on page 21
The decimal format (QDECFMT) system value determines the type of zero suppression and decimal point character used by DDS edit codes 1 through 4 and A through M. It also determines the decimal point character for decimal input fields in the interface.

"Leap year adjustment (QLEAPADJ) system value" on page 23
The Leap year adjustment (QLEAPADJ) system value adjusts the system algorithms for the leap year in different calendar systems.

"Coded character set identifier (QCCSID) system value" on page 16
The coded character set identifier (QCCSID) system value specifies the CCSID for the i5/OS operating system.

“Time separator (QTIMSEP) system value” on page 26
The time separator (QTIMSEP) system value specifies the character separator for time.

“Language identifier (QLANGID) system value” on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

“Country or region identifier (QCNTRYID) system value” on page 17
The country or region identifier (QCNTRYID) system value indicates the default country or region identifier for the system.

“English Uppercase and Lowercase (Feature 2924)” on page 259
The table shows the default system values for the English Uppercase and Lowercase (Feature 2924) national language version.

Date formats:

There is no worldwide standard for the presentation of dates. Therefore, the date format should always be stored externally as part of the textual data.

The valid date formats on the operating system are:
- *MDY (Month, day, year)
- *DMY (Day, month, year)
- *YMD (Year, Month, Day)
- *JUL (yy/ddd)
- *ISO (YYYY-MM-DD)
- *USA (MM/DD/YYYY)
- *EUR (DD.MM.YYYY)
- *JIS (YYYY-MM-DD)

Note: Some operating system functions do not support all of the previous date formats.

In database files, dates can be stored as:
- Normal numeric data fields
- SAA® date data-types

When you store dates as numeric data, your application needs to specify the format in which it is stored and presented.

When you store dates as data type DATE (L), you can specify the format with the DDS keyword DATFMT on the database file. The date is shown in this predefined format as character data, including the date separators.

If date sorting and other processing is needed, the date should be stored in *ISO format (YYYY-MM-DD) and converted to another format during the input and output operations. Write a high-level language routine to convert dates.

Related concepts

“Set job attributes (QSETJOBATR) system value” on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

Related information

DATFMT (Date Format) keyword for physical and logical files

Date separators:

The date separator for presentation should always be stored externally as part of the textual data.

The following list shows valid date separators:
- / (slash)
- - (dash)
- . (period)
- , (comma)
- (blank)

When you use decimal fields for dates, not only must your application specify the format, but it also must handle the date separators during the input operation and presentation.

When you use date-type fields, the date separators are always included in the date. To change the date separator, you can write a high-level language routine to convert dates.

Related concepts

"Set job attributes (QSETJOBATR) system value" on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

Edit date presentation:

You need to handle the presentation of dates in display and printer files differently, depending on how they are stored.

- As a normal decimal data field

  Your application program has responsibility for the way the date is entered, stored, and presented. The application must check to see that the date is entered in the right format, remove any date separators, convert the date to another format when necessary, and edit it on the display file or printer file.

  The DDS keyword DATE is used as an output-only field. DATE uses the job attributes DATE, DATFMT, and DATSEP. You can edit DATE using the edit code keyword, EDTCDE, for 6- and 8-digit date fields.

  Editing with EDTCDE includes the following changes to the appearance of displayed fields, depending on which edit code is specified:
  - Leading zeros are suppressed.
  - Zero values can be displayed as zero or blanks.
  - The field can be further edited using a user-defined edit code.

  For all other types of fields using the EDTCDE Y keyword, the program has to specify the format, and the system uses the date separator of the job that created the device file. The date separator is integrated in the object, and you are not able to change it dynamically at run time.
- As an SAA data type DATE (L) field

  The DDS date format (DATFMT) keyword allows you to specify different date formats, including default date separators, at the database field level. For the *MDY, *DMY, *YMD, and *JUL parameters, the default date separator can be changed with the date separator (DATSEP) keyword. The *ISO, *USA, *EUR, and *JIS values have a fixed separator, and the DATSEP keyword is not allowed with these values. The DATFMT and DATSEP keywords allow you to specify the format and editing characters for storing date fields. The date is shown as a character string, including the separators.

  Any format conversion between the date input and the format the database asks for can be done by:
  - Application program routines
  - Field mapping through logical files that define different date formats and separators

  For example, you can provide a date conversion that is dependent on the actual job attributes by using the following CL program:

```
PGM      PARM(&fromfmt &fromfld  &tofld  );
DCL      VAR(&fromfmt); TYPE(*CHAR)  LEN(4)
DCL      VAR(&fromfld); TYPE(*CHAR)  LEN(10)
```

```
DCL      VAR(&tofld);  TYPE(*CHAR)  LEN(10)
CVTDAT   DATE(&fromfld); TOVAR(&tofld);
FROMFMT(&fromfmt); TOFMT(*JOB)  TOSEP(*JOB)
ENDPGM
```

Your application program has to pass the format of the date you want to convert and the date itself to the CL program. The CL program assumes that the job attributes represent the way the user expects to see date fields edited. It retrieves these values and does the conversion, conforming to these values, and passes back the date in that way. The *ISO, *USA, *EUR, and *JIS values have a fixed separator that cannot be changed. If the TOFMT parameter contains one of these values, the TOSEP value is ignored.

Related information

DATE (Date) keyword for display files

DATFMT (Date Format) keyword for display files

DATSEP (Date Separator) keyword for display files

Time formats:

The i5/OS operating system supports several time formats.
- *HMS (hh:mm:ss)
- *ISO (hh.mm.ss)
- *USA (hh:mm AM or hh:mm PM)
- *EUR (hh.mm.ss)
- *JIS (hh:mm:ss)

The system value QTIME has one format (hhmmss). The time separator value is determined by the QTIMSEP system value.

The time format for presentation should always be stored externally as part of the textual data.

In database files, times can be stored as:
- Normal numeric data fields
- SAA time data-types

When you store the time as numeric data, your application needs to specify the format in which it is stored and presented.

When you store the time as data type TIME (T), you can specify the format with the DDS keyword TIMFMT on the database file. The time is sorted in this predefined format as character data, including the time separators. To convert time fields from one format to another, write a CL program or high-level language routine to do the conversion.

Time separators:

The i5/OS operating system allows several valid time separators.
- : (colon)
- . (period)
-   (blank)
- , (comma)

The time separator for presentation should always be stored externally as part of the textual data.

When you use decimal-data fields for time fields, your application needs to specify the format and time separators on the input and presentation operations.

When you use time-type fields, the time separators are always included in the time field. To change the time separators, write a CL program or high-level language routine to do the conversion.

Related concepts

"Set job attributes (QSETJOBATR) system value" on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

Edit time presentation:

You need to handle the presentation of times in display files and printer files differently, depending on the way they are stored.

- As a decimal data field

  Your application program has responsibility for the way the value is entered, stored, and presented. The program must check for the correct format, eliminate the time separators, convert the time to another format when necessary, and edit it on the display file or printer file.

  The editing can be done by specifying the edit word (EDTWRD) for the field. The TIME keyword is an output-only field. Both the edit word and TIME keyword use the information available at creation time. The time separators are integrated in the device file object.

  Both ways force you to have different copies of the source and objects for different editing requirements.
- As an SAA data type TIME (T) field

  The operating system allows you to specify different time formats and time separators on the database file level. The TIME keywords allow you to specify the format and editing characters for storing time fields. The time type field is shown as a character string that includes the separators.

  As an SAA data type, you can specify such time fields as normal character fields on the display file or printer file. On an input operation, your program has to check entered values for the correct format and separators and move them over to the database field. On an output operation, you just move the character string from the database file field to the device file field, including the separators. Any format conversion between the input and output format and the format that the database asks for can be done by either of the following two ways:
  - Application program routines
  - Field mapping through logical files that define different time format and separators

Decimal formats:

You can change the decimal format with the QDECFMT system value to reflect the way decimals are presented for your country or location.

Related concepts

"Decimal format (QDECFMT) system value" on page 21
The decimal format (QDECFMT) system value determines the type of zero suppression and decimal point character used by DDS edit codes 1 through 4 and A through M. It also determines the decimal point character for decimal input fields in the interface.

Sort sequences:

The i5/OS operating system supports sort sequence. By using one of the listed options, you can order your data according to cultural-dependent requirements for specific applications.

- Hexadecimal sorting (sort sequence tables not used). This is the default.
- A user-supplied or system-supplied shared-weight sort sequence table or unique-weight sort sequence table, determined by the SRTSEQ parameter.

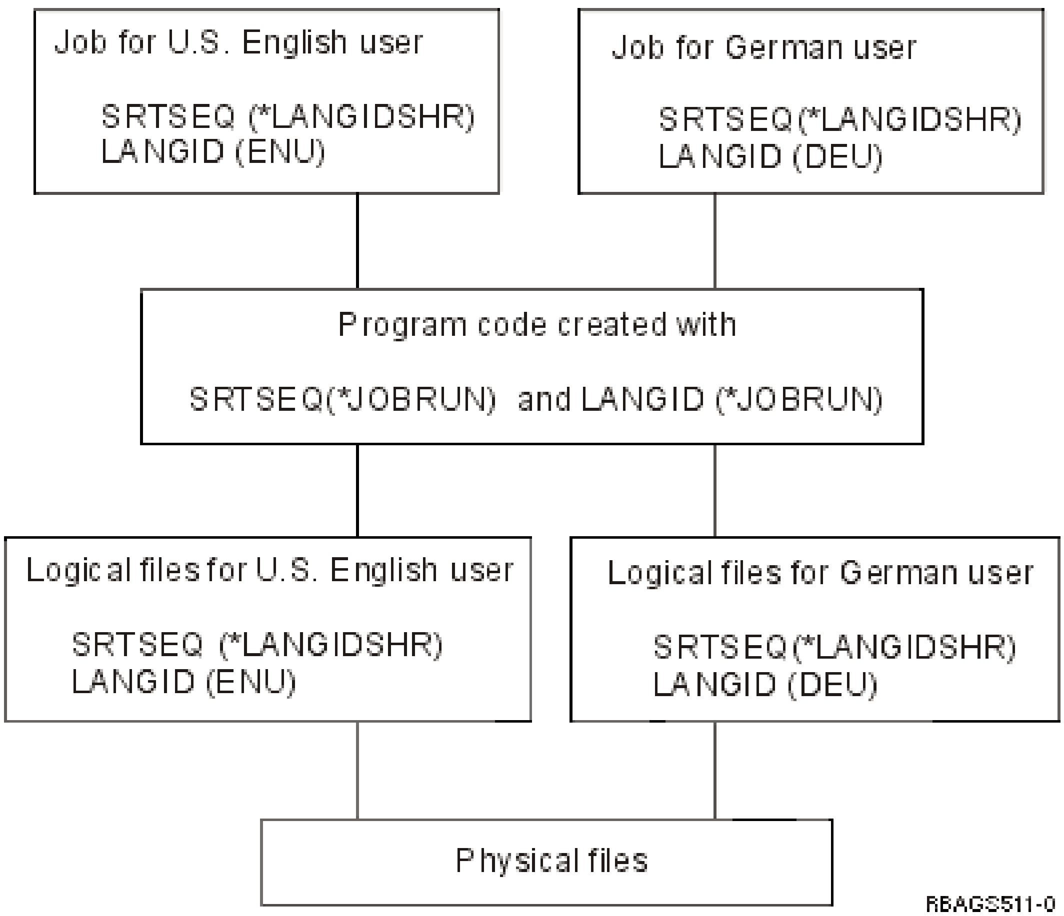
The following example shows how to use one DDS source file to create database files with different sort sequences. The following steps can be performed:

```
CRTxF     FILE(*CURLIB/NAME)
          SRTSEQ(*JOB)
          LANGID(*JOB)
```

You can then change the job attributes to create files with different sort sequences.

The CL program and high-level language programs can be created by specifying either early binding or late binding of a sort sequence. With early binding of a sort sequence, the sort sequence table to be used is determined at compile time. With late binding of a sort sequence, the sort sequence table to be used is determined at run time.

Late binding makes it possible to use one set of programs in different national language environments. The following figure illustrates using different sort sequences for different jobs with one set of physical files and program code. The sort sequence table defined for the job and used by the program should be the same as (or compatible with) the sort sequence table assigned to the logical files accessed through the library list.



Designing for running with different sort sequences:

If your program is expected to run with different sort sequences, consider the following conditions:
- Presenting the data in different order.
- Processing different records.

  Specifying selection criteria such as less than or greater than can result in selecting different records. The selection criteria equal to can result in selecting a different number of records when the shared-weight sort sequence table is used.
- Processing of a conditional branch may be different.

Note: System lists (such as the output from the WRKOBJ command) are not affected by sort sequence support.

You can use the DDS file-level keyword alternate sequence (ALTSEQ) to specify the sequencing table and the library in which it is contained. The system-supplied sort sequence tables with shared and unique weight can be used for defining the alternative collating sequence.

The alternative collating sequence table is inserted into the file at compile time and is not needed at run time. You can have different files containing different collating sequences using one set of DDS.

Note: The alternative collating sequence defined in your database files must also be defined in your application programs; otherwise, you might get unexpected results.

The DDS ALTSEQ keyword provides limited support for sequencing. It has no effect on select/omit logic. The ALTSEQ keyword can only be used with the SRTSEQ(*SRC) parameter on the Create Physical File (CRTPF) and Create Logical File (CRTLF) commands.

Related concepts

“Set job attributes (QSETJOBATR) system value” on page 24
The set job attributes (QSETJOBATR) system value sets job attributes at job startup time.

ALTSEQ (Alternate Collating Sequence) keyword

“Character sorting” on page 155
Traditionally, information is displayed in sorted order to enable users to easily find the items they are looking for. However, users of different languages might have very different expectations of what a sorted list should look like.

Related reference

Create Physical File (CRTPF) command

Create Logical File (CRTLF) command

Display file design:

Application panels typically consist of major elements, such as constant text strings, input and output fields, and cursor positioning specifications.

Note: You can handle these either as a program-described or an externally described file using DDS. The information found in this topic is based on the externally described technique using DDS.

Constant text strings:

When designing your panels, keep in mind that different languages have different space needs for the same expression.

Do not place many fields on the same line, except for a list panel that has column headings instead of field prompts. Do not overload the panels with information. Choose one of the techniques described under Textual data code design to make your panels.

Related concepts

“Textual data code design” on page 68
You can use different techniques to specify, store, and use constant text. You can use each technique for specific types of textual data components. Each technique has its advantages and disadvantages.

Input and output fields:

You must define fields according to the needs of the different languages, countries, cultures, currencies, and laws that you want to address with your application.

For example, assume that you want to store the British pound and the Japanese yen in the same field as the United States dollar. You must set the field size to accommodate the higher number of digits needed for the British pound.

Field editing specifications:

For the edit specification of your numeric, date, and time fields, you must consider the different cultural conventions of the users.

Do not code the format and editing instructions in your application program in a way that requires program modification when another convention is needed.

Related concepts

"Cultural-dependent design" on page 79
Different countries might have different standards, which you must consider when developing an NLS-enabled application. This culturally sensitive information must be placed outside the program the same way as the textual data is handled.

Cursor positioning specifications:

Do not specify cursor positioning values to fixed locations on the screen, because different languages have different space requirements.

When you work with different display files, you can adjust them with the translation process. When you need to work with field-independent cursor locations, store the positional information outside of your code and retrieve the variable values for the keyword within your program.

For example:

```
A     record-name      CSRLOC(field-name-1 field-name-2)
```

Cursor positioning on the field level is more useful in an NLS environment. For normal records, this is done by specifying the DSPATR(PC) keyword on a specific field. For subfiles, the cursor can be positioned using SFLRCDNBR(CURSOR) keyword on a special positioning field. In addition, the subfile record number must be stored in that field before the format is written.

For example:

```
A     field-name      4S 0B  line pos  SFLRCDNBR(CURSOR)
```

Note: The name of the record and field where the cursor is positioned, the subfile relative record number, and subfile fold/truncate indicator can be returned to your application program. This function is provided by hidden fields on the DDS keywords RTNCSRLOC, SFLCSRRRN, and SFLMODE.

Related information

RTNCSRLOC (Return Cursor Location) keyword for display files

SFLCSRRRN (Subfile Cursor Relative Record Number) keyword for display files

SFLMODE (Subfile Mode) keyword for display files

Input field default values:

You can use these methods to put default values into the input fields of your display. Users can override the default values with their own data.

- Getting information from program

  Never hard code the values as a literal if they are language or cultural-dependent values. Use values you can get from the system-provided information, such as system or job date, or get the values from a data object, such as a database file or data area from outside of the program.
- Using DDS keywords DFT (Default) or DFTVAL (Default Value)

Specify the default input value directly on the DDS after the keyword. The DDS keyword DFT is for input-only (I) fields. For output-only (O) or input-output (B) fields, use the keyword DFTVAL.

For example:

```
A     field-name   length type I   line pos  DFT('default    ')
or
A     field-name   length type O/B line pos  DFTVAL('default value ')
```

- Using DDS keyword MSGID (Message Identification)

  Using the Message Identification (MSGID) keyword allows you to retrieve the content of a specified message description when the program is run and to put that value as a default in your display file field. The field must be input-output capable (B) for you to use this technique.

  For example:

  ```
  A   field-name  length type B line pos  MSGID(message-id [*libl/message-file)
  ```

  This allows you to use different message files for each national language version by setting the library list accordingly when the program is run.

  Related information

  DFT (Default) keyword for display files

  DFTVAL (Default Value) keyword for display files

  MSGID (Message Identifier) keyword for display files

Field validation specifications:

Some DDS keywords provide validation checks on input-capable fields on your display.

- RANGE (Range checking)
- VALUES (Values checking)
- CMP and COMP (Comparison)
- CHECK (Check validity, keyboard control and cursor control)

Using the DDS keywords with any hard-coded values that are language, country, or cultural-dependent makes duplication and modification of the DDS and the application program necessary.

Example: Validation checks

An example of field validation checks on input-capable fields on your display using the DDS keywords VALUES, COMP, and CHECK follows:

```
A     field-name   length type usage line pos  VALUES('Y' 'N')
or
A     field-name   length type usage line pos  COMP(EQ 'US$')
or
A     field-name   length type usage line pos  CHECK(M10 or M11)
(Modulus checking)
or
A     field-name   length type usage line pos  CHECK(RL)
(Right-to-left support)
```

Validation checks are provided according to the sort sequence defined for the display file at creation time. You can use the same DDS source file to create objects for different languages. For example, the following command creates a display object tagged with the Latin 1 sort sequence table:

```
CRTDSPF FILE(name) SRTSEQ(*LANGIDSHR) LANGID(DEU)
```

The following specification:

```
A     field-name   length type usage line pos  COMP(EQ 'a')
```

accepts all lowercase, uppercase, and accented characters, as defined by the shared-weight in the Latin 1 sort sequence.

In addition, note that all the checks specified using those DDS keywords are done by the data management function of the operating system. Any error message caused by wrong input or handling by the user appears in the language of the operating system. This can be the primary language or a secondary language, depending how the library list of the job is set up.

You can override this when you use the additional DDS keyword CHKMSGID (Check Message Identifier). This keyword allows you to specify your own customized messages and message file to be used by the checking routines of the operating system.

For example:

```
A     field-name   length type usage RANGE(1 999)
A                                    CHKMSGID(USR1234 [*libl/]APPMSGF [&MSGFLD1])
A     MSGFLD1      length type   P   TEXT('Message data field')

and

ADDMSGD   MSGID(USR1234) MSGF(APPTXDENU/APPMSGF)
          MSG('Value &1; is out of range 1 to 999')
and

ADDMSGD   MSGID(USR1234) MSGF(APPTXDDEU/APPMSGF)


MSG('Wert &1; ist ausserhalb des gültigen Bereichs 1 bis 999')
```

To use different message files of different library names, do not specify a fixed library name. You can use a message file for different languages by setting the library list when you run the program.

Related information

RANGE (Range) keyword for display files

VALUES (Values) keyword for display files

CMP (Comparison) keyword for display files

CHECK (Check) keyword for display files

CHKMSGID (Check Message Identifier) keyword for display files

Error messages:

You can provide error messages in a display file by specifying text as constant, or by using predefined messages.

- Specifying text as constant on ERRMSG or SFLMSG keywords

  Specify the text directly as a constant on the DDS keyword. When you want to have more than one language, you must duplicate the DDS source code and translate constants within the DDS specifications. You can then create a separate display file object for each language.
- Using predefined messages on ERRMSGID or SFLMSGID keyword

  When using predefined messages instead of constants, you need to have multiple display files.

  Instead of using different display files, exchange only the used message file by setting the library according to the language that you want to use.

  For example:

  ```
  A     field-name   length type usage EDTCDE(x)
  A 61                                 ERRMSGID(USR3456 [*libl/]APPMSGF [&MSGFLD2])
  A     MSGFLD2      length type   P   TEXT('Message data field')

  and

  ADDMSGD   MSGID(USR3456) MSGF(APPTXDENU/APPMSGF)
            MSG('Delivery date &1; is earlier than production end date &2')
  and
  ```

```
ADDMSGD   MSGID(USR3456) MSGF(APPTXDDEU/APPMSGF)
          MSG('Lieferdatum &1; ist . . .')
.
.
.
```

Related information

ERRMSG (Error Message) and ERRMSGID (Error Message Identifier) keywords for display files

SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files

Printer file design and translation:

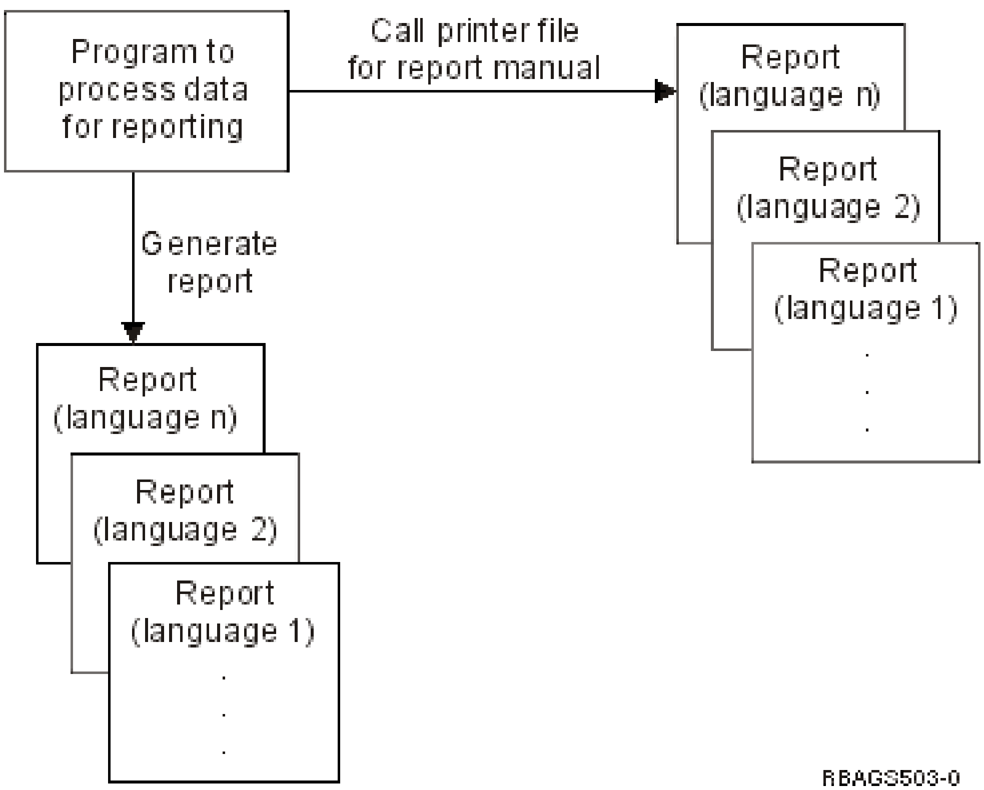
Program-described printer files and externally described printer files are two types of printer files. When you design printer files to be translated into a national language version, you should follow some specific guidelines.

- 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.

The following figure shows how externally described printer files are used in creating reports for a different national language version.



Printer file translation

When you design printer files to be translated into a national language version, consider these guidelines:

- 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 devices 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.
- When entering data, consider these parameters on the Create Printer File (CRTPRTF) command.
  - PAGESIZE (page size)

    Different countries have different page-size standards.
  - OVRFLW (overflow line number)

    The overflow line number must be less than or equal to the page length.
  - CHRID (character set and code page)

    If the CHRID parameter of the printer file is set to *DEVD, the printer uses the character identifier that was set on the control panel or specified in the device description.

    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: All code pages and character sets cannot be handled by all printers.

Related concepts

"Textual data code design" on page 68
You can use different techniques to specify, store, and use constant text. You can use each technique for specific types of textual data components. Each technique has its advantages and disadvantages.

Related reference

Create Printer File (CRTPRTF) command

Source file design:

Database source files are implicitly assigned the CCSID of the job when they are created, unless they have been explicitly assigned a CCSID value through the CCSID parameter on the Create Physical File (CRTPF) or Create Source Physical File (CRTSRCPF) command.

If the job CCSID is 65535, the job default CCSID (DFTCCSID) is used as the implicitly assigned CCSID. The job default CCSID is determined by the system language identifier value and the job DBCS-capable indicator.

Related reference

Create Physical File (CRTPF) command

Create Source Physical File (CRTSRCPF) command

Character data representation architecture design:

To enable your application for a multilingual environment, avoid coding CCSID values directly in your DDS for physical files. When database sharing takes place, you need to define your files with the CCSID of the primary language or use Unicode.

- Avoid coding CCSID values directly in your DDS for physical files. When creating different physical files for different languages, change the CCSID for your job (using the CHGJOB command). Only one set of DDS source code needs to be maintained.

  Conversions between all CCSIDs might not make sense in all cases. For example, if you access a Greek database with a CCSID of 00875 from a German display station with a job CCSID of 00273, you see garbled data on your display.

  Countries outside the Latin-1 character set use character sets that include non-Latin characters. No meaningful conversion is possible between the non-Latin code points and the Latin code points. Arabic, Greek, Hebrew, and Turkish are SBCS languages with non-Latin characters.
- When database sharing takes place, define your files with the CCSID of the primary language being used. Make sure that all users have the CCSID of the language that they use defined in their user profile.

Related concepts

"Working with CCSIDs" on page 126
Using the system implementation of Character Data Representation Architecture (CDRA), you can achieve consistent representation, processing, and interchange of coded characters (data) on the i5/OS operating system and across IBM Systems. The primary implementation of CDRA on the i5/OS operating system is through coded character set identifier (CCSID) support.

"Working with Unicode" on page 111
Unicode is a standard that precisely defines a character set as well as a small number of encodings for it. It enables you to handle text in any language efficiently. It allows a single application to work for a global audience.

Related reference

Change Job (CHGJOB) command

Use of the Send Network File command:

When you use the Send Network File (SNDNETF) command, the data (if the command is sending a member only) is assumed to be in the CCSID of the job that is running the command. Therefore, no conversion takes place.

When the data is received, you should store the member in a file with the same CCSID as the originating file.

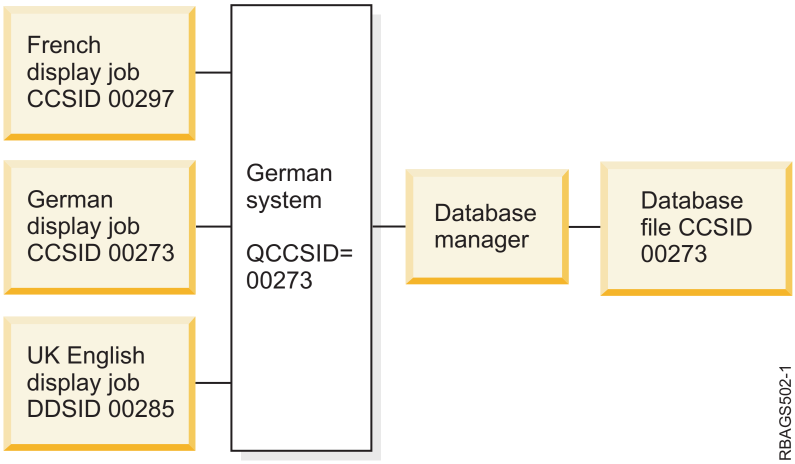
If the receiver does not know the CCSID of the incoming file member, it can be received into a file with a CCSID of 65535, which indicates that no conversion takes place.

Related reference

Send Network File (SNDNETF) command

Scenario: Multilingual single system:

This scenario shows a multilingual single system with German as the primary language and English and French as secondary languages. All users enter data into the same database file.



On this multilingual system, all users are entering character data into a single file with CCSID 00273 (German), and character data entered from the English and French display stations is being mapped into the German file.

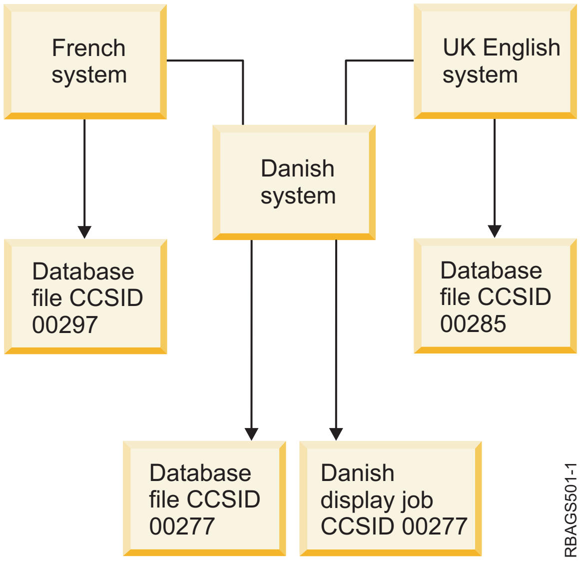
To preserve correct mapping, fields defined as character fields should be actual character fields. If the fields contain application development values (for example, control characters or fields that are not used as real character fields), the fields either should be specified as hexadecimal fields or assigned a CCSID value of 65535.

Using CCSIDs, characters that cannot be converted between different code pages are replaced with a substitution code. If you are using a user-defined data stream (UDDS) to format and lay out your display (instead of using DDS), you might get substitution codes returned after the system reads and inserts that data in your user-defined data stream. Substitution codes might cause unpredictable results on the display.

Scenario: Multilingual network:

This scenario shows a multilingual network with three systems located in three different countries, each with a different language.

In this example, the application on the Danish system is using distributed relational database. All national characters (regardless of the language that the data is stored in) are displayed correctly at the Danish display. When the CCSID of the language is used by the database, the integrity of the database is preserved. The conversion of data between the different code pages is completely automatic and part of the database management.



Handling languages with no NLV support:

If you need to support a language that does not have a supported national language version, follow these steps.

1. Study the available national language versions. Find out which national language version most closely resembles your language in character representation.
2. Install the most appropriate national language version as your primary language.
3. Modify the system values to meet your cultural needs. For example, set date and time formats to meet those of the culture that you are supporting.
4. Configure your workstations and printers to match your primary language. Then, handle discrepancies between support for the installed NLV and your own language.

   Note: The workstation customization functions can support only those capabilities built into your hardware. You cannot support functions through workstation customization that your hardware is unable to support.
5. Use the Create Table (CRTTBL) command to create a sort sequence table based on the existing table that most nearly matches the appropriate sorting sequence for your language.
6. If your language is a DBCS language, create your own characters (UDC) to represent missing characters in the code page associated with the NLV you installed. UDC is an acronym for a user-defined character that is created through the character generator utility (CGU). CGU is an extension of the code page with special user-defined ideographic characters, symbols, or logos.

Related concepts

"System values for other languages with no national language version" on page 270
Some of the system values are associated with languages and countries that do not have a national language version. You should set these values immediately after initially installing the i5/OS operating system.

Related reference

Create Table (CRTTBL) command

Programming considerations in globalized application design

As you develop your globalized applications, the national language version environment often requires that you pay additional attention to how you prepare and compile your code.

Coding globalized applications with high-level languages

Your major goal must be to have only one general set of running code that is common for all language versions and to make your programs table-driven as much as possible.

You should do as follows:

- Base validity checks on database accesses and message files rather than on hard-coded literals or tables.
- Base calculations on variable factors retrieved from a file rather than coding them inline.
- Place cultural-dependent functions into separate modules of the application and call them when you cannot code them flexibly.

Do not use hard-coded values unless they are fully language and cultural independent on comparison, scan, replace, or call operations. In addition, do not use uppercase or lowercase-sensitive values. For example, never hard code Yes and No (Y or N) responses in your program, because these values are different for every language, and should be part of the textual data.

For literals and constants in source code, use characters only from the invariant character set. If input data is checked for validity in the program, make sure that the characters checked belong to the invariant character set; otherwise you might get a situation where the user is requested to enter a character that is not even on his keyboard. For example, the left brace ({) and right brace (}) do not appear on Arabic keyboards.

Do not use compile-time arrays to hold messages or any other language or cultural-sensitive data.

For better performance, when you need to call external NLS-dependent modules, call them by a fixed name as a literal (but based on the library list) rather than by a variable field containing the program name. This allows your application to call the modules of different libraries based on the associated library list.

To allow users to work with an application in the language and habits of their culture, specify the editing values (for example, date, time, and date separators) as dependent on the language and country or region. You can then retrieve them according to the information in the user profile. The parameters are LANGID (language identifier) and CNTRYID (country or region identifier). You need to retrieve the cultural-sensitive information only once at program initiation. You can do this by an initial CL program or by the high-level language program and prepare them as:

- Parameters on the call operation
- Parameters on the local data area (LDA)
- Program load tables

Using an initial program allows you to set the user's job attributes to present a consistent application.

Related concepts

"Invariant character set" on page 324
An *invariant character set* is a character set, such as the syntactic character set, whose code point assignments do not change from code page to code page. The table illustrates the invariant character set (character set 00640) on the i5/OS operating system.

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

"Country or region identifier (QCNTRYID) system value" on page 17
The country or region identifier (QCNTRYID) system value indicates the default country or region identifier for the system.

Language compilers CCSID:

Some language compilers expect syntactical operators and the naming convention for the source code to be in CCSID 00037.

You can refer to the documentation for the language compiler you use.

For these compilers, incorrect mapping occurs if the source is compiled with a CCSID other than 00037 or 65535. You must ensure that these compilers receive any variant characters used in language syntax in CCSID 00037.

ILE language compilers

When an ILE C, ILE RPG, or ILE COBOL program is compiled, source from database source files is converted to the CCSID of the primary source file.

Compilers for these languages can handle syntactical operators in most CCSIDs. These compilers can also handle naming conventions for the source code in most CCSIDs.

Non-ILE language compilers

When a non-ILE CL, non-ILE RPG, or non-ILE COBOL program is compiled, source from database source files is converted to the CCSID of the job.

If you do not want your names, constants, or literals converted to the CCSID of the job, you can change your job CCSID to 65535. Your constants, literals and names then remain intact.

Note: REXX/400 procedures and the literal data coded within them are not converted to the job CCSID.

Example 1

The following example shows a sample non-ILE RPG program. This example shows English source on a system in the United States.

```
*  RPG Source   (Source file created using CCSID 00037 but tagged
*                with CCSID 65535)
FFILE1  IF  E                    DISK                         80
C                     READ FILE1
C* Test char
C*
C           FLD1      IFEQ '$'
C                     ...
C* Move char
C*
C                     MOVE FLD1        FLD$
C                      ...
C*
C                     SETON                                  LR
```

Example 2

In Finland, the program in the first example does not compile because the field name FLD$ contains a variant character (the dollar sign). The variant character represents a different code point in a code page other than 00037. This figure shows the same sample non-ILE RPG program as English (U.S.) source on a system in Finland (CCSID 278).

```
*  RPG Source   (Source file created with CCSID 00037, but tagged
*                with 65535)
FFILE1  IF  E                    DISK                         80
C                     READ FILE1
C* Test char
C*
```

```
C FLD1 IFEQ 'ä '
C                    ...
C* Move char
C*

C MOVE FLD1 FLDä
C                     ...
C*
C                    SETON                              LR
```

Example 3

You can correct this error by changing the file CCSID to 00037 and setting the job CCSID to 00278 (for Finland). The following example shows the changed file as seen English source in Finland.

```
*  RPG Source   (Source file created using CCSID 00037 and tagged
*                with CCSID 00037)
FFILE1  IF  E                   DISK                         80
C                    READ FILE1
C* Test char
C*
C          FLD1      IFEQ '$'
C                    ...
C* Move char
C*
C                    MOVE FLD1        FLD$
C                     ...
C*
C                    SETON                              LR
```

Session manager:

For all applications that use a session manager, you must ensure that the output data stream has no X'3F' values in it. The i5/OS operating system uses X'3F' values to make a screen blank.

General sort sequence

The sort sequence used by a program might influence the program logic. The following figure shows an example of this.

Using the Latin 1 shared-weight sort sequence, character test 3 is equivalent to character test 4 (not all characters are shown). When using hexadecimal or unique sorting, they are completely different. The following example shows an RPG program using different sort sequences.

```
*  RPG Source   (Program created with Latin 1 sort sequence)
*
C* Test char 3
C*
C          FLD1      IFEQ 'a'
C                    ...
C* Test char 4
C*
C          FLD1      IFEQ 'a'
C          FLD1      OREQ 'A'

C FLD1 OREQ 'á '

C FLD1 OREQ 'Á '
```

```
C                     ...
C*
C                    SETON                                LR
```

If you compile the program with *JOBRUN specified for the SRTSEQ parameter and *JOBRUN specified for the LANGID parameter, the sort sequence table used at run time is not known at compile time.

The IBM DB2 Query Manager and SQL Development Kit for i5/OS licensed program and ILE C have additional special considerations.

Related concepts

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

ILE C considerations:

When you compile programs with ILE C, consider this information.

- You can compile a source file in any EBCDIC code page except code page 00290.
- If the CCSID of the primary source file is 65535, code page 00037 is assumed.
- All secondary source files are converted to the CCSID of the primary source file.

  Note: While most secondary source files are converted to the CCSID of the primary source file, some conversions are not supported. Contact your IBM service representative if you require support for an unsupported CCSID conversion.
- If the CCSID of the secondary source files is 65535, no conversion takes place.
- Any modules are created in the code page of the primary source file. A module is an operating system object that can be a collection of one or more procedures and one or more definitions for external or internal variables. A module is compiled from source code.
- When binding modules of different CCSIDs, no conversion takes place and unpredictable results might occur.
- You can use the trigraph support for the C characters that are not available on all keyboards. Trigraph support generally uses invariant characters to represent variant characters. For example, the left bracket ([) is represented by ??(.

The ILE C runtime library functions that parse strings containing variant characters use the variant character code point value associated with the CCSID of the job.

ILE RPG sort sequence:

The ILE RPG feature, an option of the IBM WebSphere Development Studio for System i licensed program, provides the possibility for a user to specify a sort sequence table and to use the table in comparison operations that are performed with nonnumeric data.

For each of the supported languages, two tables (a shared-weight table and a unique-weight table) are included with the system. With sort sequence support, you can create sort sequence tables based on the existing ones.

The control specifications provide the ILE RPG compiler with information about your program and your system. The sort sequence used in ILE RPG programs is controlled by all of the following items:

- The control specifications.
- The SRTSEQ (sort sequence table) parameter on the Create RPG Module and the Create Bound RPG Program commands.
- The LANGID (language identifier) parameter on the Create RPG Module and the Create Bound RPG Program commands.

The alternative collating sequence field (ALTSEQ) in the control specifications allows the following values:

blank No alternative collating sequence is used in the RPG program. The normal collating sequence is used in the RPG program. The compile options SRTSEQ and LANGID are ignored.

***NONE**
No alternative collating sequence is used in the RPG program. The normal collating sequence is used in the RPG program. The compile options SRTSEQ and LANGID are ignored.

***SRC** The alternative collating sequence is used in the RPG program, according to the tables entered at the end of the RPG program. The alternative collating sequence table is loaded at compile time, and ordering, sorting, comparing, and match field processing is done according to that table.

The SORTA and LOOKUP operation codes do not use specified alternative collating sequence tables.

The SRTSEQ and LANGID parameters on the Create RPG Module and Create Bound RPG Program commands are ignored.

***EXT** The alternative collating sequence is specified outside of the RPG program. RPG compiler imports an external sort sequence table, based on the SRTSEQ and LANGID parameters on the Create RPG Module and the Create Bound RPG Program commands.

The SORTA and LOOKUP function with the arrays and tables at compile time and processing time take effect only when you specify D in the control specifications.

The sort sequence table to be used by the program can be determined at compile time or when the job is run. If the SRTSEQ parameter of the Create RPG Module and Create Bound RPG Program commands:

- Is set to *HEX, no sort sequence table is used.
- Specifies a table name, then that table is stored with the program object to be used when the job is run. For system-supplied default sort sequence tables for the supported languages, refer to Sort sequence tables.
- Is set to *LANGIDSHR or *LANGIDUNQ, the shared-weight or unique-weight table for the language determined by the LANGID parameter is stored with the program object. For a list of valid language identifiers, refer to Language and country and region identifiers.
- Is set to *JOB, the SRTSEQ parameter of the compile time job is used to determine the sort sequence. The table is stored with the program object.
- Is set to *JOBRUN, the attributes of the job running the compiled program determine the sort sequence to be used. If the SRTSEQ attribute of the job refers to the LANGID, the LANGID stored with the program object is used. If the LANGID stored with the program is also *JOBRUN, the LANGID of the runtime job is used.

Notes:

1. If the table to be stored with the program object at compile time does not exist, a table defining hexadecimal sort sequence and tagged with a CCSID value of 65535 is used.
2. If the sort sequence table and the CCSID of the job running the program differ, the table is converted to the CCSID of the job.

SORTA and LOOKUP operation codes

The implementation of compare operation codes, match field and control field processing with the sort sequence tables is the same for the alternative collating sequence and for the sort sequence support. Compare operation codes are ANDxx, COMP, CABxx, CASxx, DOUxx, DOWxx, IFxx, ORxx, and WHxx. Additional functions provided with the SORTA and LOOKUP operation codes follow:

SORTA
The data in the array is sorted according to the sort sequence table.

LOOKUP

To provide proper table searching, the sort sequence table is used with the search arguments in the arrays and tables.

The search argument and either the table or array element are compared using the sort sequence table.

The array and table data are checked using the sort sequence table, whenever ascending or descending sequence is specified. If the SRTSEQ and LANGID parameter values resolve to retrieve the sort sequence table again at run time, then the array and table elements are loaded without a sequence check at the compile time. The sequence checks are performed at run time, according to the sort sequence table.

Related concepts

"Sort sequence (QSRTSEQ) system value" on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

"Sort sequence tables" on page 156
A sort sequence table is an object that contains the weight of each single-byte graphic character within a specified coded character set identifier (CCSID). The system-recognized identifier for the sort sequence table object type is *TBL.

"Country and region identifiers" on page 249
This table lists the country and region identifiers.

ILE COBOL sort sequence:

The ILE COBOL feature, an option of the IBM WebSphere Development Studio for System i licensed program, uses the sort sequence support in several ways.

- Create COBOL Module command
- Create Bound COBOL Program command
- PROCESS clause
- ALPHABET clause

The ILE COBOL licensed program uses sort sequence tables that are system supplied or user supplied.

Creating COBOL module and creating bound COBOL program commands

These CL commands have two compiler options relating to sort sequence support: the SRTSEQ parameter and LANGID parameter. The SRTSEQ parameter allows the user to specify any of the system-supplied or user-supplied sort sequence tables residing in a specified library. You can specify whether the sort sequence table should be taken at compile time or run time. Also, you can choose between the shared-weight and unique-weight tables.

With the LANGID parameter, you can specify one of the system-defined language identifiers, or leave that parameter to be defined at the run time.

The meanings of the SRTSEQ and LANGID parameters on the Create COBOL Module and Create Bound COBOL Program commands are the same as on the Create RPG Module and Create Bound RPG Program commands as described in "ILE RPG sort sequence" on page 100.

PROCESS statement

Sort sequence support options can be supplied in the PROCESS statement. The syntax for that command is like that for the Create COBOL Module and Create Bound COBOL program commands. The only exception to this is that the values for the parameters in the PROCESS statement are entered without an asterisk (*) for the predefined values. Any options specified in the PROCESS statement override the corresponding options on the Create COBOL Module and Create COBOL program commands.

ALPHABET clause

The alphabet-name in the ALPHABET clause of the SPECIAL-NAMES paragraph can use the NLSSORT option. Use the SRTSEQ and LANGID parameters of the compiler for alternative collating sequence options. Otherwise, it means the same as the NATIVE option.

The following COBOL lines are affected by the NLSSORT option:

- PROGRAM COLLATING SEQUENCE phrase of OBJECT-COMPUTER paragraph

  When evaluating the result of nonnumeric comparisons, the alphabet name has to be referenced in this phrase to enable the program to use the specified sort sequence options. This option also applies to the nonnumeric sort or merge operation. Otherwise, the hexadecimal collating sequence is used.
- ALPHABET CLAUSE in the SPECIAL-NAMES paragraph

  This clause should specify the NLSSORT option.
- COLLATING SEQUENCE in the MERGE (or SORT) statement

  This phrase is used to specify the collating sequence to be used for nonnumeric comparisons for the KEY data name in the MERGE or SORT operation. If omitted, the PROGRAM COLLATING SEQUENCE clause in the OBJECT-COMPUTER paragraph defines the collating sequence to be used. If neither is specified, hexadecimal collating sequence is used.
- Nonnumeric relation names and condition names

  The selected sort sequence table affects the result of certain statements, using nonnumeric relation names and condition names: EVALUATE, IF, PERFORM...UNTIL, SEARCH and START. The truth values of the nonnumeric comparisons depend on the corresponding weights of the characters in the selected sort sequence table. For example, if you specify unique-weight table (LANGIDUNQ) for French (Latin 1), the following statement is true for the single value of the variable ITEM-1,e.

  ```
  IF ITEM-1 = "e"
  ```

  If you specify a shared-weight table (LANGIDSHR) for French (Latin 1), the same statement is true for several values of the variable ITEM-1. All have the same shared weight of 77:

  ```
  lowercase e (e), uppercase e (E),

  lowercase e acute ( é ), uppercase e acute ( É ),

  lowercase e grave ( è ), uppercase e grave ( È ),

  lowercase e caret ( ê ), uppercase e caret ( Ê ),

  lowercase e umlaut ( ë ), uppercase e umlaut ( Ë )
  ```

DB2 and SQL sort sequence:

For Interactive SQL, the SRTSEQ and LANGID parameters can be specified on the STRSQL command. You can change these parameters by using the session services for interactive displays.

Sort sequence tables are used for all string comparisons. String comparisons are performed in the following SQL statements:

- ORDER BY clause
- WHERE clause

- GROUP clause
- HAVING clause
- UNION and UNION ALL clauses
- DISTINCT clause
- BETWEEN predicate
- IN predicate
- LIKE predicate
- MIN and MAX scalar functions
- MIN and MAX column functions

In addition, any indexes or views that are created using the CREATE INDEX or the CREATE VIEW statements are created with the specified sort sequence table.

IBM DB2 Query Manager and SQL Development Kit for i5/OS

The IBM DB2 Query Manager and SQL Development Kit for i5/OS licensed program does not assume a particular CCSID when precompiling source. Any variant characters in the language syntax, such as the not (¬) symbol, are assumed to be encoded in the CCSID of the source file.

For example, if the source file has a CCSID of 00037, the ¬ not symbol is correctly interpreted to be at code point X'5F'. If the source file has a CCSID of 00500, however, the ¬ not symbol is correctly interpreted to be at code point X'BA'.

A literal is stored in the CCSID of the source file.

The IBM DB2 Query Manager and SQL Development Kit for i5/OS licensed program calls the appropriate language compiler to create an SQL program; therefore, you must consider the general guidelines for high-level languages.

Related concepts

"Sort sequence (QSRTSEQ) system value" on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

System i Access sort sequence:

You can specify the sort sequence in System i Access functions. When performing queries on the system databases and SQL tables, you can specify the system-supplied or user-supplied sort sequence tables.

Remote SQL support

You can specify the way the selected data has to be sorted when performing the query. For that purpose, sort fields must be specified in the ORDER BY clause. The following clauses also use the specified sort sequence:

- WHERE clause
- GROUP BY clause
- HAVING clause
- JOIN BY clause
- UNION clause

- DISTINCT clause
- IN predicate
- LIKE predicate
- BETWEEN predicate
- RANGE predicate
- MAX function
- MIN function

The actual sort sequence table is retrieved from the job attributes of the user. The SRTSEQ and LANGID parameters can be affected through changing the user profile or changing the job attributes.

Data transfer support

When transferring data from the system to the workstation, you can specify the sort sequence to be applied on selected data. The sort sequence table is also used in the following string comparison operations:
- WHERE clause
- GROUP BY clause
- HAVING clause
- JOIN BY clause
- IN predicate
- LIKE predicate
- BETWEEN predicate
- MAX function
- MIN function

You can specify in the OPTION statement the following parameters related to sort sequence:
- SRTSEQ (sort sequence table)
  - *JOB
  - *HEX
  - *LANGIDSHR
  - *LANGIDUNQ
  - *LIBL/sort-seq-table-name
  - *CURLIB/sort-seq-table-name
  - library-name/sort-seq-table-name
- LANGID (language identifier)
  - *JOB
  - language-identifier

You can choose the appropriate sort sequence through options on System i Access displays.

Related concepts

"Sort sequence (QSRTSEQ) system value" on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

"Language identifier (QLANGID) system value" on page 21
The language identifier (QLANGID) system value specifies the default language identifier for the system.

Coding globalized applications that use bidirectional data

When you are developing NLV-enabled applications, you should consider some specific restrictions on bidirectional languages.

- Bidirectional language display layout

  The presentation of data should have a right-to-left orientation. Literals should appear on the right side of the fields that they describe. The following examples illustrate a U.S. English display with a left-to-right orientation and the same display in a right-to-left orientation.

  Left-to-right layout of a U.S. English display

```
Display employee record (DSPEMPRCD)
Type choices, press enter.

Employee code . . . . . . . . ______________ Code, *ALL
Field name  . . . . . . . . . ______________ Name, *ALL
File name  . . . . . . . . . . ______________ Name
Library name . . . . . . . . . ______________ Name, *LIBL
Output to . . . . . . . . . . . ______________ *CONS, *PRINT
```

  Right-to-left layout of a U.S. English display

```
                       (DSPEMPRCD)  drocer eeyolpme yalpsiD
                                  .retne sserp ,seciohc epyT

   *ALL ,edoC ______________ . . . . . . . . . edoc eeyolpmE
   *ALL ,emaN ______________ . . . . . . . . . . . . .eman dleiF
         emaN ______________ . . . . . . . . . . . . . .eman eliF
  *LIBL ,emaN ______________ . . . . . . . . . . . eman yrarbiL
*CONS ,*PRINT ______________ . . . . . . . . . . . . . . ot tuptuO
```

- Long fields in bidirectional languages

  Avoid defining input fields that span more than one line. When the field is displayed or printed as one entity, the result for bidirectional languages is not what the user intended.

- Variable positioning in bidirectional languages

  Your application must allow for variables to be in any order. For example, consider the following message in English:

  ```
  File &1 in library &2 not found
  ```

  When translated to another language, the message might look like this:

  ```
  dnuof ton &2 yrarbil ni &1 eliF
  ```

  In this case, variable 2 is positioned before variable 1.

- CHECK(RL) and CHECK(RB) keywords with bidirectional languages

  These options are valid only for display stations capable of right-to-left movement, and have the following restrictions:

  – Option indicators are not valid with cursor control codes.
  – CHECK(RZ) and CHECK(RB) are not valid with these keywords.
  – A field that spans more than one line gives a warning message.
  – The check digit for modulus checking is the farthest-right byte in the field.

– CHECK(RL) applies to character fields only.

- Online information for bidirectional languages

  The special bidirectional tags have a restriction. When combining online help information from several panel groups that do not have the same value for the BIDI tag, the user must use the hot key sequence to read the opposite orientation of the online help information.

- CCSIDs for bidirectional languages

  As bidirectional languages have special character sets that are unique, no exchange of data into other languages is feasible. You might need to use data mapping between EBCDIC and ASCII data streams, however. For example, you need data mapping between EBCDIC and ASCII data streams if you are using Distributed Relational Database Architecture™ (DRDA®).

  When exchanging data in a language that uses Latin characters and when special characters that are not part of the invariant character set are needed, use CCSID 00424 for Hebrew and CCSID 00420 for Arabic for data mapping to take place.

Related concepts

"CCSID reference information" on page 333
Coded character set identifier (CCSID) is a 16-bit number that includes a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and other information that uniquely identifies the coded graphic-character representation.

"Working with bidirectional data" on page 174
Arabic and Hebrew languages use an alphabet written and read from right to left. Numerics and Latin text imbedded in the right-to-left text are written and read from left to right. Therefore, these languages are called bidirectional languages.

Related reference

"Checklist: Bidirectional support guidelines" on page 175
When you create an application with bidirectional support, follow the guidelines in this table.

Using message catalogs

The i5/OS operating system can use message catalogs to store messages. Messages in a message catalog are grouped as sets. Each message has a unique number within a set.

You can create a message catalog as a stream file, a source file member, or a user space object type from one or more source files.

Because you can store message catalogs as stream files, you can use directories to isolate messages for specific products or national language versions.

Creating or updating a message catalog with the GENCAT and MRGMSGCLG commands

You can use both the Generate Message Catalog (GENCAT) command or the Merge Message Catalog (MRGMSGCLG) command to create or update a message catalog. When a message catalog exists, continued use of these commands updates a catalog by comparing the original messages to the messages in the source. New message text replaces specific messages without changing the other messages within the set. With these commands, you can add or delete messages from an existing set of messages. You can also delete sets of messages from an existing message catalog.

Related concepts

"CCSID support for messages" on page 136
You can use CCSID support for handling messages and message catalogs on the i5/OS operating system by using commands and application programming interfaces. You can send messages tagged with one CCSID to users with a different CCSID.

Related reference

Generate Message Catalog (GENCAT) command

Merge Message Catalog (MRGMSGCLG) command

Source for message catalogs:

The source for a message catalog is either a source physical file, a stream file, or multiple files. The source contains fields to define set numbers, message numbers, message text, or to specify sets to delete.

The following information provides additional information and examples relating to message catalogs.

Message catalog source format

A message catalog contains five fields of message text source lines. A single blank character separates each of the five fields. Any other blank characters are considered as part of the subsequent field data. See "Special characters and escape sequences" on page 110 for additional information.

Note: Enter the key fields exactly as in the following list, using the dollar sign ($) and lowercase characters. Definitions for maximum and minimum values are stored in QSYSINC/QSYS/LIMITS.

- **$ comment**

  A line that begins with $ that is followed by one or more blank characters is treated as a comment line. A omment line should be placed directly beneath the message to which it refers. Place comments for an entire set directly below the $set directive in the source file.
- **$quote** *C*

  This line specifies an optional quote character *C* that is used to surround message text. This character enables trailing spaces or null (empty) messages to be visible in a message source line. By default, or if an empgy $quote directive is supplied, no quoting of message text is recognized.
- **$set n comment**

  This line specifies the set identifier of the messages to follow until the next $set or end-of-file appears. The *N* denotes the set identifier that is defined by a number between 1 and NL_SETMAX. Place set identifiers in ascending order within a single source file. They do not need to be contiguous. A character string that follows a set identifier is treated as a comment and ignored.
- **$delsetncomment**

  This line deletes message set *n* from an existing message catalog. The *n* specifies the set number. Data that follows the set number is treated as a comment. The $set and $delset identifiers can both be in the message catalog source or the field tags.
- **m message text**

  The *m* specifies the message identifier that is defined by a number between 1 and NL_SETMAX. The message text is stored in the message catalog with message identifier *m* with the set identifier that is specified in the last $set directive. If the message text is empty and a blank character field separator is present, it stores an empty string in the message catalog. Existing messages get deleted from the catalog if the message line does not have a field separator or MESSAGE TEXT and a NEWLINE or carriage return follows the message line. Message identifiers must be in ascending order, noncontiguous, and within a single set. The length of the MESSAGE TEXT must be in the range of 0 to NL_TEXTMAX.

  Note: Empty lines in a message text source file will be ignored.

Messages programming format

MESSAGES should follow these recommendations:

- The last line of all messages should end with \n.
- The second and remaining lines of a message should begin with \t, indicating a tab.
- All lines of messages that continue to the next line should end with \n\, indicating that the message continues to the next line.

- The quotation mark at the end or beginning of a line should be omitted. The quotation mark delineates the beginning and end of a complete message.

Using multiple source files

You can specify multiple source files for the source file parameter. The messages that are contained in all of the files must follow the same rules for sets and messages as defined in a single source file. For example, the first source file contains messages in sets 1 through 3. The next source file must begin with set 3 and have a message number greater than the last message number in the first source file. If not, it must contain sets that begin with a number higher than the highest number (set 3) in the previous source file.

Replacing messages

Messages in an existing message catalog can be replaced by specifying a source file that contains the same set number and message number as the message text you want to change. All other messages in the source file remain the same. To update a value for the $QUOTE in a catalog, use the same $QUOTE character in subsequent source files.

Example source for a message catalog

Here is a sample format for the source that is used to create a message catalog. A quotation mark delineates each message. The message text that is stored in the message catalog has had the extraneous blank characters removed. This example describes three sets of messages. Set 2 is deleted while sets 1 and 3 remain stored in the message catalog.

```
$ Messages for my new product
$quote "

$set 1

1 "Error occurred.\n"
$ The next message is continued on the next line.
2 "This is a very long message \n\
\t that requires another line to display. \n"
3 "Specify a value greater than %d.\n"
4 "File %c cannot be used at this time.\n"

$set 2
1 "Error %d occurred. \n"
2 "Flag not set.\n"
3 "Number of arguments must be %d.\n"

$set 4
1 "Before using this command, you must \
set the correct values in the %c box.\n"
2 "You have not properly NLS enabled this function.\n"
10 "Messages should end with a %c.\n"

$delset 2
```

Note: Message 2 in set 1 will be displayed in two lines. Message 1 in set 4 will display as a one line message.

Here is an example for using the MRGMSGCLG command to create a message catalog.

```
MRGMSGCLG CLGFILE('/MYPRODUCT/MESSAGES?US')
     SRCFILE('QSYS.LIB/MYLIB.LIB/MYSOURCE.FILE/US.MBR')
     CLGCCSID(*SRCCCSID) SRCCCSID(*SRCFILE)
     TEXT('Message catalog for USA')
```

This example creates a message catalog into the stream file US in directory /MYPRODUCT/MESSAGES using the source from MYLIB library in file MYSOURCE and member US. The CCSID of the data in the message catalog is the same as the CCSID tag of the source file.

Special characters and escape sequences

Text strings can contain special characters and escape sequences as defined in the following table.

Description of special characters	Sequence
\	\\
backspace	\b
carriage return	\r
form feed	\f
horizontal tab	\t
NEWLINE	\n
octal bit pattern	\ddd **Note:** The escape sequence \ddd consists of a backslash followed by up to three octal digits that specify the value of the required character. If the character following the backslash is not an octal digit, the backslash and data following are included as part of the text.

Opening, extracting, and closing message catalogs:

After you have created a message catalog, you can use these functions: CATOPEN(), CATGETS(), and CATCLOSE().

CATOPEN()
Opens a message catalog

CATGETS()
Extracts a message from a message catalog, given a set identifier and a message identifier

CATCLOSE()
Closes the message catalog

The C function CATOPEN opens the message catalog. If no slash (/) characters are found in the name, the NLSPATH environment variable and the LC_MESSAGES category are used to find the specified message catalog. If the name contains one or more slash (/) characters, the name is interpreted as a path name of the catalog to open.

A default path is used if there is no NLSPATH environment variable or a message catalog cannot be found in the NLSPATH path specified. If the value of oflag is NO_CAT_LOCALE the environment variable setting of LC_MESSAGES may affect the default path. If the value of oflag is zero the LANG environment variable may affect it also.

Related information

WebSphere Development Studio: ILE C/C++ Language Reference PDF

Delivering globalized applications

As you prepare to deliver your globalized application, you should consider how globalization issues might affect the ways that your customers install and use your application.

Hardware support for multilingual systems

Hardware, in this context, means the physical keyboards, displays, printers, and controllers that make up a System i platform. The extent to which this hardware supports national languages might 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 some basic rules to ensure translation goes smoothly. A subset of these rules is provided in the User interfaces topic.

Delivering your globalized application to customers

Delivering your application to customers includes the processes of packaging, servicing, supporting, and educating users about your application. You must consider various tasks when following these processes in different countries and cultures throughout the world.

Related concepts

"User interfaces" on page 64
A user interface is the part of a software product that your customer actually sees.

"Packaging and installation process" on page 55
You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

Handling data in globalized applications

The i5/OS operating system enables you to handle data in a globalized environment. This topic collection describes Unicode and Unicode 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.

One of the most critical challenges you might encounter as you work with globalized systems and applications is the effective interaction with data. The operating system provides a wide range of options that you can use to ensure that data is viewed and processed seamlessly across national languages.

Related concepts

"Developing globalized applications" on page 50
Globalized applications are applications that have national language support. National language support allows users to enter, store, process, retrieve, print, and display data in their chosen language. It also allows users to see and enter commands, prompts, messages, and documentation in their chosen language, in formats matching their cultural expectations.

Working with Unicode

Unicode is a standard that precisely defines a character set as well as a small number of encodings for it. It enables you to handle text in any language efficiently. It allows a single application to work for a global audience.

Before Unicode, the encoding systems that existed did not cover all the necessary numbers, characters, and symbols in use. Different encoding systems might assign the same number to different characters. If you used the wrong encoding system, your output might not have been what you expected to see.

Unicode provides a unique number for every character, regardless of platform, language, or program. Using Unicode, you can develop a software product that works with various platforms, languages, and countries. Unicode also allows data to be transported through many different systems. Modern systems provide Internationalization solutions based on Unicode.

Unicode was developed as a single-coded character set that contains support for the common languages around the world. The first version of Unicode used 16-bit numbers, which allowed for encoding 65 536 characters without complicated multibyte schemes. With the inclusion of more characters, and following implementation needs of many different platforms, Unicode was extended to allow more than one million characters. In addition, other encoding schemes were added, such as UTF-8, UTF-16, and UTF-32. This introduced more complexity into the Unicode standard, but far less than managing a large number of different encodings.

The original Unicode repertoire covered all major languages commonly used in computing. Unicode continues to grow and to include more scripts.

The design of Unicode differs in several ways from traditional character sets and encoding schemes:

- Its repertoire enables users to include text efficiently in almost all languages within a single document.
- It can be encoded in a byte-based way with one or more bytes per character, but the default encoding scheme uses 16-bit units that allow much simpler processing for all common characters.
- Many characters, such as letters with accents and umlauts, can be combined from the base character and accent or umlaut modifiers. This combining reduces the number of different characters that need to be encoded separately. *Precomposed* variants for characters that existed in common character sets at the time were included for compatibility. For example, Latin small letter A used with a combining tilde results in ã.

Characters and their usage are well-defined and described. Traditional character sets typically provide only the name or a picture of a character and its number and byte encoding; Unicode has a comprehensive database of properties available. It also defines a number of processes and algorithms for dealing with many aspects of text processing to make it more interoperable.

The early inclusion of all characters of commonly used character sets makes Unicode a useful mechanism for converting between traditional character sets, and makes it feasible to process non-Unicode text by first converting the text into Unicode, processing the text, and then converting it back to the original encoding without loss of data.

Related concepts

"Scenarios: Setting up i5/OS with a national language version" on page 45
These scenarios demonstrate how you can enable multilingual support on the i5/OS operating system.

"Character data representation architecture design" on page 93
To enable your application for a multilingual environment, avoid coding CCSID values directly in your DDS for physical files. When database sharing takes place, you need to define your files with the CCSID of the primary language or use Unicode.

Related information

Unicode Home Page

Why use Unicode

Unicode has many advantageous functions.

The operating system provides multilingual support. Unicode provides the means to store and retrieve data in the user's national language of choice in a single file, and therefore provides for one database file to support all text needs, regardless of the language of the input device. For example, the same parts file can have Greek, Russian, and English descriptions and names in it.

Related concepts

“GB18030: The Chinese standard” on page 125
GB 18030-2000 is a Chinese standard that specifies an extended code page for use in the Chinese market.

Different encodings of Unicode

The Unicode standard has several main ways in which a Unicode value can be encoded. They are UTF-8, UTF-16, and UTF-32. Unicode Transformation Format (UTF) is the algorithmic mapping from every Unicode value to a unique byte sequence.

UTF-8:

UTF-8 converts Unicode data through a mathematical algorithm so that UTF-8 uses 8 data bits to encode the data, keeps all ASCII codes from 00 to 7F encoded as themselves, and contains nulls only when they are the intended characters.

For example, the string ″ABC″ in Unicode is ″004100420043″x. However, in UTF-8 it is ″414243″.

Because UTF-8 allows Unicode data to flow over an 8-bit network without the network needing to know that it is Unicode, UTF-8 is used to store Unicode on several UNIX® platforms and is used as the default encoding for most new internet standards.

UTF-8 is used mainly as a direct replacement for older MBCS encodings, which all use 8-bit code units, but it takes some more code to process it. It is a good encoding if 90% of your data is English, because all English letters use only one byte.

The i5/OS operating system supports UTF-8 encoding with CCSID 1208. Beginning with i5/OS V5R3, CCSID 1208 is supported in database.

UTF-16:

UTF-16 is an encoding of Unicode in which each character is composed of either one or two 16-bit elements.

Unicode was originally designed as a pure 16-bit encoding, aimed at representing all modern scripts. Over time, and especially after the addition of over 14 500 composite characters for compatibility with established sets, it became clear that 16 bits were not sufficient for most users. Out of this arose UTF-16.

UTF-16 allows access to about 60 000 characters as single Unicode 16-bit units. It can access an additional 1 000 000 characters by a mechanism known as surrogate pairs.

Two ranges of Unicode code values are reserved for the high (first) and low (second) values of these pairs. Highs are from 0xD800 to 0xDBFF, and lows from 0xDC00 to 0xDFFF. Because the most common characters have already been encoded in the first 64 000 values, the characters requiring surrogate pairs are relatively rare.

UTF-16 is extremely well designed as the best compromise between handling and space, and all commonly used characters can be stored with one code unit per code point. This is the default encoding for Unicode.

The i5/OS operating system supports UTF-16 encoding with CCSID 1200 (and CCSID 13488). Beginning with i5/OS V5R3, CCSID 1200 is supported in database. CCSID 13488 has been supported in database for several releases.

Related concepts

"UCS-2 and its relationship to Unicode (UTF-16)"
The UCS-2 standard, an early version of Unicode, is limited to 65 535 characters. However, the data processing industry needs over 94 000 characters; the UCS-2 standard has been superseded by the Unicode UTF-16 standard.

Related information

Unicode Home Page

UTF-32:

UTF-32 is an encoding of Unicode in which each character is composed of 4 bytes.

The i5/OS operating system does not support UTF-32 encoding with a CCSID value.

Unicode was originally designed as a pure 16-bit encoding, aimed at representing all modern scripts. Over time, and especially after the addition of over 14 500 composite characters for compatibility with established sets, it became clear that 16 bits were not sufficient for many users. Out of this arose UTF-32.

UTF-32 allows characters to be encoded as 4 bytes at any code point from 00000000 to 0010FFFF. For example, the string *ABC* in UTF-32 is encoded as `x"000000410000004200000043"`.

Related information

Unicode Home Page

UCS-2 and its relationship to Unicode (UTF-16)

The UCS-2 standard, an early version of Unicode, is limited to 65 535 characters. However, the data processing industry needs over 94 000 characters; the UCS-2 standard has been superseded by the Unicode UTF-16 standard.

The i5/OS operating system supports CCSID 13488, defined as UCS-2, and CCSID 1200, defined as UTF-16. The system treats both CCSID 13488 and CCSID 1200 as UTF-16 encodings.

Using either scheme, you will have the same results for almost all system operations. However, certain SQL functions that operate on a character boundary defined by the SQL standard can produce different results. For instance, the SQL functions of CHARACTER, LENGTH, POSITION, and SUBSTRING distinguish UTF-16 and UCS-2, and therefore you get different results. See the SQL reference for more information about these functions.

UCS, UCS-2 (Universal Multiple-Octet Coded Character Set)

The ISO 10646 standard is a character code designed to encode text for storage in computer files. The design of the ISO 10646 standard is based on today's prevalent character code, ASCII (and ISO 8859-1, an extended version of the ASCII code). But ISO 10646 goes beyond ASCII's ability to encode only the Latin alphabet. The ISO 10646 encoding provides the capability to encode all of the characters used for written languages throughout the world.

Two UCS encoding schemes

To accommodate the many thousands of characters used in international text, ISO/IEC 10646 specifies the Universal Multiple-Octet Coded Character Set (UCS). UCS can be implemented through the following encoding schemes:

- UCS-2: Each character is represented by 16 bits or 2 bytes. (The number 2 in UCS-2 indicates 2 bytes.) For example, uppercase A is represented by 0041. This encoding is no longer sufficient and has been superseded by the UTF-16 encoding.

- UCS-4: Each character is represented by 32 bits or 4 bytes. (The number 4 in UCS-4 indicates 4 bytes.) For example, uppercase A is represented by 0000 0041.

The i5/OS operating system does not support UCS-4 encoding with a CCSID value.

Related concepts

"UTF-16" on page 113
UTF-16 is an encoding of Unicode in which each character is composed of either one or two 16-bit elements.

Related information

SQL reference

How Unicode relates to prior standards such as ASCII and EBCDIC

The Unicode standard is advantageous to other standards. It can reduce the complexity of handling character data in globalized applications.

Evolving standards based on limited platforms

The representation of character data in modern computer systems can be fairly complicated, depending on the needs of your globalized application. One of the reasons for this complexity is that the methods for handling this data have evolved from early methods that served less complicated environments and hardware platforms.

In fact, many early decisions about how to encode characters on a system were guided by the functional requirements of specific devices, such as the early Telex (TTY) terminals and punch card technologies. For example, the Delete character (with an ASCII value of x'7F') was required in order to punch out all of the holes in a column of a punch card to signify that the column should be ignored. The storage capacities of these early computing systems placed additional limitations on system and application designers.

The character encoding schemes that have grown out of these early systems were built on this historical foundation:

- The ASCII (American Standard Code for Information Interchange) character set uses 7-bit units, with a trivial encoding designed for 7-bit bytes. It is the most important character set in use today, despite its limitation to very few characters, because its design is the foundation for most modern character sets. ASCII provides only 128 numeric values, and 33 of those are reserved for special functions.
- The EBCDIC (Extended Binary-Coded Decimal Interchange Code) character set and a number of associated character sets, designed by IBM for its mainframes, uses 8-bit bytes. It was developed at a similar time as ASCII, and shares the same set of base characters and has other similar properties. Unlike ASCII, the Latin letters are not combined in two blocks for upper- and lower-case. Instead, the letters are arranged so that their hexadecimal values have second digits of 1 through 9 (another punch card-friendly design).

Historical simplicity creates modern complexity

The physical and functional limitations of the early character sets gave way to rapidly expanding hardware and functional capabilities. Character representation on computing systems became less dependent on hardware; instead, software designers used the existing encoding schemes to accommodate the needs of an increasingly global community of computer users.

Character sets for many characters

The most common encodings (character encoding schemes) use a single byte per character, and they are often called single-byte character sets (SBCS). They are all limited to 256 characters. Because of this, none of them can even cover all of the accented letters for the Western European languages. Consequently, many different such encodings were created over time to fulfill the needs of different user communities.

The most widely used SBCS encoding today, after ASCII, is ISO-8859-1. It is an 8-bit superset of ASCII and provides most of the characters necessary for Western Europe.

However, East Asian writing systems needed a way to store over 10 000 characters, and so double-byte character sets (DBCS) were developed to provide enough space for the thousands of ideographic characters in East Asian writing systems. Here, the encoding is still byte-based, but each two bytes together represent a single character.

Even in East Asia, text contains letters from small alphabets like Latin or Katakana. These are represented more efficiently with single bytes. Multi-byte character sets (MBCS) provide for this by using a variable number of bytes per character, which distinguishes them from the DBCS encodings. MBCSs are often compatible with ASCII; that is, the Latin letters are represented in such encodings with the same bytes that ASCII uses. Some less often used characters may be encoded using three or even four bytes.

An important feature of MBCSs is that they have byte value ranges that are dedicated for lead bytes and trail bytes. Special ranges for lead bytes, the first bytes in multibyte sequences, make it possible to decide how many bytes belong together to encode a single character. Traditional MBCS encodings are designed so that it is easy to go forwards through a stream of bytes and read characters. However, it is often complicated and very dependent on the properties of the encoding to go backwards in text: going backwards, it is often hard to find out which variable number of bytes represents a single character, and sometimes it is necessary to go forward from the beginning of the text to do this.

Examples of commonly used MBCS encodings are Shift-JIS and EUC-JP (for Japanese), with up to 2 or 3 bytes per character.

Stateful encodings

Some encodings are stateful; they have bytes or byte sequences that switch the meanings of the following bytes. Simple encodings, like mixed-byte EBCDIC, use Shift-In and Shift-Out control characters (bytes) to switch between two states. Sometimes, the bytes after a Shift-In are interpreted as a certain SBCS encoding, and the bytes after a Shift-Out as a certain DBCS encoding. This is very different from an MBCS encoding where the bytes for each character indicate the length of the byte sequence.

The most common stateful encoding is ISO 2022 and its language-specific variations. It uses Escape sequences (byte sequences starting with an ASCII Escape character, byte value 27) to switch between many different embedded encodings. It can also *announce* encodings that are to be used with special shifting characters in the embedded byte stream. Language-specific variants like ISO-2022-JP limit the set of embeddable encodings and specify only a small set of acceptable Escape sequences for them.

Such encodings are very powerful for data exchange but hard to use in an application. Their flexibility allows you to embed many other encodings, but direct use in programs and conversions to and from other encodings are complicated. For direct use, a program has to keep track not only of the current position in the text, but also of the state--which embeddable encoding is currently active--or must be able to determine the state for a position from considerable context. For conversions to other encodings, converting software might need to have mappings for many embeddable encodings, and for conversions from other encodings, special code must figure out which embeddable encoding to choose for each character.

Why Unicode?

Hundreds of encodings have been developed, each for small groups of languages and special purposes. As a result, the interpretation of text, input, sorting, display, and storage depends on the knowledge of all the different types of character sets and their encodings. Programs are written to either handle one single encoding at a time and switch between them, or to convert between external and internal encodings.

Part of the problem is that there is no single, authoritative source of precise definitions of many of the encodings and their names. Transferring of text from one machine to another one often causes some loss of information. Also, if a program has the code and the data to perform conversion between a significant subset of traditional encodings, then it carries several megabytes of data around.

Unicode provides a single character set that covers the languages of the world, and a small number of machine-friendly encoding forms and schemes to fit the needs of existing applications and protocols. It is designed for best interoperability with both ASCII and ISO-8859-1, the most widely used character sets, to make it easier for Unicode to be used in applications and protocols.

Unicode is in use today, and it is the preferred character set for the Internet, especially for HTML and XML. It is slowly being adopted for use in e-mail, too. Its most attractive property is that it covers all the characters of the world (with exceptions, which will be added in the future). Unicode makes it possible to access and manipulate characters by unique numbers (that is, their Unicode code points) and use older encodings only for input and output, if at all.

International Components for Unicode

The International Components for Unicode (ICU) is a C library that provides a full-featured, industrial strength, Unicode support.

The library provides:

- Calendar support
- Character set conversions
- Collation (language-sensitive)
- Date and time formatting
- Locales (many)
- Message catalogs (resources)
- Message formatting
- Normalization
- Number and currency formatting
- Time zones
- Transliteration
- Word, line, and sentence breaks

ICU is a collaborative, open-source development project jointly managed by a group of companies and individual volunteers throughout the world, using the Internet and the Web to communicate, plan, and develop the software and documentation.

ICU is open source. For more information about ICU license, see the International Components for Unicode Web site.

Related concepts

International Components for Unicode

Mapping of data

The i5/OS operating system uses the EBCDIC encoding scheme. However, not all clients attached to the system use an EBCDIC encoding scheme to store, retrieve, and process data. Therefore, some clients use Unicode as an exchange mechanism that is safe across all platforms.

Some clients might use ASCII, PC DATA, or other encoding schemes. They can using Unicode to prevent the loss of data due to incomplete conversion between encoding schemes and code pages.

Example: Displaying data without Unicode:

This example highlights two users on the same system, one English and the other Greek. The English user has the display device CCSID set to 37, and the Greek user has the display device CCSID set to 875. Both users query, update, and replace data in the DATABASE1 database.

DATABASE1 is tagged with CCSID 37.

Problems with data integrity develop because users are operating with CCSIDs that have varied character support. That is, not all characters in CCSID 37 are available in CCSID 875 and vice versa.

Assume that the following names are to be entered by the English-speaking user (display device supports a CCSID of 37):

- Å alson
- Gifford

When these entries are stored, the data integrity remains intact. That is, an Å is stored as an Å. This is because the display device CCSID and the database CCSID are both 37.

Assume the following names are also input into DATABASE1 by the Greek-speaking user (display device CCSID of 875):

- Å π έ ν
- Ω ρ ι μ α

DATABASE1 now consists of the following logical entries:

- Å alson
- Gifford
- Μ π έ ν
- Ω ρ ι μ α

The Greek characters that make up the name are stored as those characters only if the same character exists within CCSID 37. If the character does not exist, the system converts the characters using a predetermined algorithm to a code point from code page 37. The algorithm converts Ω to Å.

The following list shows the code point used to store the first character of each name in DATABASE1. (Using only the first character makes the example easier by eliminating long strings of code points which are shown if the code point is presented for each character in the name.)

Name CCSID 37 Stored Code Point (Hexadecimal)

Å alson
    67 . . .

Gifford
    C7 . . .

Μ π έ ν
    53 . . .

Ω ρ ι μ α
    67 . . .

The next step in this example is to show how data can be incorrectly selected due to the character conversion when it was stored in the database.

Assume that the Greek user wants to find all names beginning with Ω. The following SQL statement can provide two names: Ωριμα and Åalson

```
Select from DATABASE1 where name LIKE 'Ω %'
```

The search yielded an unexpected name (Åalson). This is because the first character in Åalson is stored with the same code point as the first character in Ωριμα.

Related concepts

"Example: Displaying data with Unicode"
This example highlights two users on the same system, one English and the other Greek. The English user has the display device CCSID set to 37, and the Greek user has the display device CCSID set to 875. Both users query, update, and replace data in the DATABASE1 database.

Example: Displaying data with Unicode:

This example highlights two users on the same system, one English and the other Greek. The English user has the display device CCSID set to 37, and the Greek user has the display device CCSID set to 875. Both users query, update, and replace data in the DATABASE1 database.

DATABASE1 is tagged with CCSID 37.

This example, using UCS-2 as the CCSID of DATABASE1, shows how data integrity is maintained both in storing and retrieving data.

As in the example of Displaying data without Unicode information, one user is English using CCSID 37 and the other user is Greek using CCSID 875.

DATABASE1 is used as in the previous example. However DATABASE1 is now defined with CCSID 13488. (13488 is a UCS-2 CCSID.)

- Åalson
- Gifford
- Μπέν
- Ωριιμα

The key difference in using UCS-2 as the CCSID of DATABASE1 is that data integrity is maintained for each user who inputs data to the database. That is each character, regardless of the CCSID of the inputting device, is stored with a unique code point. (Remember that in this example the CCSID of DATABASE1 is 13488.)

Name CCSID 13488 Stored Code Point (Hexadecimal)

Åalson
00C5 . . .

Gifford
0047 . . .

Μ π έ
03A9 . . .

Ω ρ ι μ α
039C . . .

Assume that the Greek user wants to find all names beginning with Ω. The following SQL statement can provide one name, Ω Ω ρ ι μ α, as compared to two in the previous example:

```
Select from DATABASE1 where Substr(name,1,1) = ' Ω '
```

The reason for this is that each character stored in a UCS-2 tagged database has a unique code point. This contrasts to the example of Display data without Unicode information that had the first character in Å alson stored with the same code point as the first character in Ω ρ ι μ α

Related concepts

"Example: Displaying data without Unicode" on page 117
This example highlights two users on the same system, one English and the other Greek. The English user has the display device CCSID set to 37, and the Greek user has the display device CCSID set to 875. Both users query, update, and replace data in the DATABASE1 database.

Unicode on i5/OS

The i5/OS operating system provides support for Unicode.

Unicode cannot be specified as a value for:
- The system CCSID
- A user profile CCSID
- A job CCSID

The i5/OS operating system provides external support for Unicode in the following parts of the system:
- Database files and functions
- DB2 for i5/OS
- DDS
- Display file and panel groups
- ILE high-level languages such as RPG
- Message handling and message catalogs
- Query files and tools
- SQL tables
- Unicode variables in UIM

Several other i5/OS functions use Unicode internally so that character data integrity is maintained for users across multilingual platforms.

Note: These topics do not give detailed information about application development as it relates to the implementation of Unicode. Rather, they provide highlights of i5/OS support for Unicode. Where possible, these topics provide reference to a book that provides detailed information for UCS-2 implementation. You need to have the information about the Unicode standard available and understand the information.

Related information

Unicode Home Page

Database files and functions:

When you create Unicode database applications, you need to consider the implications for creating physical files, creating logical files, and for database input/output.

Creating physical files

Unicode graphic fields can be created in physical files. This is done by specifying a G data type and a Unicode CCSID for the CCSID keyword.

The following example shows the DDS for a physical file containing four fields, and the command for creating the file:

```
A          R FMT1
A            EMPNO          6A
A            NAME          30G         CCSID(1200)
A            DESCR1       500G         CCSID(1200) VARLEN
A            DESCR2       500A
```

```
CRTPF FILE(UNICODEPF) SRCFILE(CLR/QDDSSRC)
```

In the example:

- The first field, `EMPNO`, is a character field of length 6. The CCSID of the EMPNO field is the SBCS CCSID of the job. The decision was made to use a character field because the EMPNO field contains only numerics and Unicode support is not needed.
- The `NAME` and `DESCR1` fields are both Unicode fields. Both of these fields may need to contain data from more than one EBCDIC code page so the decision was made to make these fields Unicode graphic.
- The `DESCR2` field is the SBCS CCSID of the job. This field is used as illustration of mapping to a logical field in Creating logical files.

You can specify the default (DFT) keyword for Unicode graphic fields. The default value can be specified as SBCS, bracketed-DBCS, or bracketed-DBCS-graphic character strings. If you do not specify the DFT keyword, the default value for fixed-length Unicode fields is the Unicode blank (hexadecimal 0020). For varying-length Unicode fields, the default is the empty string.

Creating logical files

You can use logical files to map Unicode data to and from character, DBCS-open, or DBCS-graphic. This allows Unicode graphic data to be manipulated in a character based form.

The following example shows the DDS for a logical file containing 4 character fields. The Unicode graphic data is converted to character data when reading from the logical file, and character data is converted to Unicode graphic data when writing to the file.

```
R FMT1                      PFILE(UNICODEPF1)
A            EMPNO
A            NAME            A         CCSID(37)
A            DESCR1          A         CCSID(37)
A            DESCR2          G         CCSID(1200)
```

Database input/output

Whenever reading or writing data from or to a field tagged with a Unicode CCSID to the job physical files, the data is passed as Unicode data without any conversions occurring. Regardless of the job CCSID, data is passed as Unicode data. When writing data to a logical file, the *from* CCSID is the job CCSID; however, if the job CCSID is 65535, the *from* CCSID is the CCSID of the field in the logical file.

Here are some scenarios from the previous physical and logical files. For the scenarios, the job CCSID is 297.

Scenario 1. When reading the data from the physical file:
- EMPNO is converted from its CCSID to 297.
- NAME is not converted but is left as Unicode data.
- DESCR1 is not converted but is left as Unicode data.
- DESCR2 is converted from its CCSID to 297.

Scenario 2. When writing the data to the physical file:
- EMPNO is converted from 297 to its CCSID.
- NAME is not converted but is left as Unicode data.
- DESCR1 is not converted but is left as Unicode data.
- DESCR2 is converted from 297 to its CCSID.

Scenario 3. When reading the data from the logical file:
- EMPNO is converted from its CCSID to 297.
- NAME is converted from Unicode data to character data with a CCSID of 297.
- DESCR1 is converted from Unicode data to character data with a CCSID of 297.
- DESCR2 is converted from character data to Unicode data and not converted to the job CCSID.

Scenario 4. When writing the data to the logical file:
- EMPNO is converted from 297 to its CCSID.
- NAME is converted from 297 to Unicode data.
- DESCR1 is converted from 297 to Unicode data.
- DESCR2 is converted from Unicodeto its CCSID in the physical file.

Scenario 5. If the job was 65535, the conversions for the previous fields are:
- EMPNO is not converted.
- NAME is converted from 37 to Unicode data.
- DESCR1 is converted from 37 to Unicode data.
- DESCR2 is converted from Unicode to its CCSID in the physical file.

Related concepts

"Object-level coded character set identifier 65535" on page 137
CCSID 65535 is the default object-level CCSID for message files and message queues.

DB2 for i5/OS:

When using DB2 for i5/OS applications, you need to be aware of some restrictions of Unicode and some commands.

- Implicit conversion when comparing Unicode fields with character/IGC/graphic fields as well as with literals and host variables can occur.
- Physical and logical files with Unicode fields cannot have their CCSIDs changed with the Change Physical File (CHGPF) command.
- A Unicode CCSID is not allowed on the CHGPF command.
- The Copy File (CPYF) and Copy From Query File (CPYFRMQRYF) commands with FMTOPT(*MAP) specified is not allowed when copying from or to a Unicode graphic field unless:
  - The corresponding field is a Unicode or DBCS-graphic field.

- The corresponding field is a character, DBCS-open, DBCS-either, or DBCS-only field with a CCSID other than 65535.

- The Copy File (CPYF) command supports copying of SBCS character, DBCS-open, DBCS-only, DBCS-either, and DBCS-graphic fields to and from Unicode graphic fields. There is limited support for Unicode on the FROMKEY, TOKEY, INCCHAR, and INCREL parameters.

  Related concepts

  "Object-level coded character set identifier 65535" on page 137
  CCSID 65535 is the default object-level CCSID for message files and message queues.

  Related reference

  Change Physical File (CHGPF) command

  Copy File (CPYF) command

  Copy From Query File (CPYFRMQRYF) command

SQL tables:

SQL supports tables that contain Unicode graphic columns by specifying a Unicode CCSID for the GRAPHIC and VARGRAPHIC data types.

The following SQL example creates the table U_TABLE. U_TABLE contains one character column called EMPNO, and two Unicode graphic columns. NAME is a fixed-length Unicode graphic column and DESCRIPTION is a variable-length Unicode graphic column. The decision was made to use a character field because the EMPNO field only contains numerics and Unicode support is not needed. The NAME and DESCRIPTION fields are both Unicode fields. Both of these fields may contain data from more than one EBCDIC code page.

```
CREATE TABLE U_TABLE (EMPNO CHAR(6) NOT NULL,
NAME GRAPHIC(30) CCSID 1200,
DESCRIPTION VARGRAPHIC(500) CCSID 1200)
```

Inserting data

SBCS character, mixed character, and DBCS graphic data can be inserted into Unicode graphic columns using the SQL INSERT statement. DB2 for i5/OS SQL converts the data to Unicode graphic data. In SQL programs, the DECLARE VARIABLE statement can be used to attach a Unicode CCSID to graphic host variables.

The following SQL example converts character data to Unicode graphic data for the NAME and DESCRIPTION columns and inserts the row into the U_TABLE.

```
INSERT INTO U_TABLE VALUES('000001','John Doe','Engineer')
```

Selecting Unicode data

Implicit conversion of Unicode graphic data is supported on a FETCH or select INTO and CALL.

In the following example, the EMPNO column is returned in empno_hv as character data. The NAME column is returned in name_hv as Unicode graphic data because name_hv is a Unicode variable. It is not converted to character, mixed character, or DBCS graphic.

```
...
char empno_hv[7];
wchar_t name_hv[31];
EXEC SQL DECLARE :name_hv VARIABLE CCSID 13488;
...
EXEC SQL SELECT EMPNO, NAME
INTO :empno_hv, :name_hv
.FROM U_TABLE;
...
```

To return Unicode graphic data as EBCDIC data, the prior example can be changed to return the Unicode data as character data, EMPNO and NAME are returned in the job CCSID.

```
...
char empno_hv[7];
char name_hv[31];
...
EXEC SQL SELECT EMPNO, NAME
INTO :empno_hv, :name_hv
FROM U_TABLE;
...
```

When doing selection, implicit conversions are done when comparing Unicode graphic data and character or DBCS graphic data.

The following example converts the character string 'John Doe' to Unicode graphic and then selects the rows where the NAME column is 'John Doe'.

```
EXEC SQL DECLARE C1 CURSOR FOR
SELECT *
FROM U_TABLE
WHERE NAME = 'John Doe';
```

The SQL Reference information includes additional information about using SQL with Unicode graphic data.

Related concepts

SQL reference

Query files and tools:

The Open Query File (OPNQRYF) command can retrieve or perform the selection of Unicode data. IBM Query for i5/OS, DB2 Query Manager, and i5/OS DB2 Query Management all have Unicode support.

Open query file (OPNQRYF) command considerations

The Open Query File (OPNQRYF) command, shown as follows, can retrieve or perform selection of Unicode data. Using the MAPFLD parameter, data can be mapped to or from Unicode.

```
OPNQRYF FILE(U_TABLE)
QRYSLT('NAME=MAPNAME')
MAPFLD((MAPNAME 'John Doe' *GRAPHIC *N *N 1200))
```

Interactive query tools considerations

IBM Query for i5/OS, DB2 Query Manager, and i5/OS DB2 Query Management all have Unicode support. Unicode data can be displayed or printed on a report by implicitly converting to either character or mixed art.

Related reference

Open Query File (OPNQRYF) command

Related information

Query Manager Use PDF

Query Management Programming PDF

Data description specifications:

In data description specifications (DDS), you use the CCSID file-, record-, or field-level keyword to specify that a G-type field supports Unicode data instead of DBCS-graphical data.

Related concepts

CCSID (Coded Character Set Identifier) keyword for physical and logical files

Display files and panel groups:

Unicode data is not supported on display devices that currently support the 5250 data stream. Therefore, conversions between the Unicode data and EBCDIC are necessary during input and output operations.

On output, the Unicode data is converted to the CCSID of the device. On input, the data is converted from the device CCSID to the Unicode CCSID.

Because the device CCSID, which is determined from the device configuration, determines what the Unicode data is converted to, the converted data appears differently on different devices. For example, a Unicode character that maps to a SBCS character is displayed as a DBCS replacement character on a graphic-DBCS capable device. On a DBCS or SBCS capable device, the character appears as a SBCS character. A Unicode character that maps to a DBCS character is displayed as a graphic-DBCS character on a graphic-DBCS capable device. On a DBCS device, a DBCS character is bracketed (enclosed in a shift-out and shift-in). A SBCS replacement character is displayed on a SBCS device.

It is also suggested that all Unicode capable fields be initialized in the output buffer before writing the fields to the screen. Unpredictable results can occur if default initialization is allowed to take place.

Related concepts

Unicode considerations for database files

Unicode variables in user interface manager:

This example shows how to define a Unicode variable in user interface manager (UIM).

```
1     :class name=example basetype='graphic 6 13488' width=10,
2
3     :class name=example2 basetype='graphic 10 13488' width=20.
4
```

Line 1 defines a class for variables that will contain 6 Unicode characters and is to be displayed in a field that is 10 bytes long.

Line 3 defines a class for variables that will contain 10 Unicode characters and is to be displayed in a field that is 20 bytes long.

Related reference

DDS concepts

Related information

Application Display Programming PDF

GB18030: The Chinese standard

GB 18030-2000 is a Chinese standard that specifies an extended code page for use in the Chinese market.

The i5/OS operating system supports this encoding with CCSID 1392. Generally, you should use Unicode instead of CCSID 1392 for complete national language support. CCSID 1392 is provided if you need to handle or interchange GB18030 encoded data.

A brief history of major GB code pages

A common base code page standard for Chinese is GB 2312-1980. It encodes more than 6000 frequently used Chinese ideographs. With the growing importance of Unicode and the parallel standard ISO 10646 (which was adopted by China as GB 13000), an extension of GB 2312-1980 was created. This extension was called GBK; it encoded all 20 902 unified ideographs that are assigned in Unicode 2.1. GBK is not a formal standard, but is a widely implemented specification.

Unicode 3.0 added more than 6000 ideographs, and version 3.1 added about 42 000 additional ideographs.

GB 18030 was created as an update of GBK for Unicode 3.0 with an extension that covers all of Unicode. It has the following general features:

- GB 18030 character assignments are compatible with the GB 2312-1980 standard and the GBK specification.
- The mapping table between GB 18030 and Unicode is compatible with the one between GB 2312-1980 and Unicode. With some exceptions (with the one between GBK and Unicode), most of the changes compared to the GBK mapping table are due to updates for Unicode 3.0.
- GB 18030 specifies a mapping table that covers all Unicode code points. It is functionally similar to a UTF (Unicode Transformation Format) while maintaining compatibility of GB-encoded text with GBK and GB 2312-1980.

Related concepts

"Why use Unicode" on page 112
Unicode has many advantageous functions.

Related information

Unicode Home Page

Working with CCSIDs

Using the system implementation of Character Data Representation Architecture (CDRA), you can achieve consistent representation, processing, and interchange of coded characters (data) on the i5/OS operating system and across IBM Systems. The primary implementation of CDRA on the i5/OS operating system is through coded character set identifier (CCSID) support.

Related concepts

"Automatic character set and code page conversion" on page 37
The i5/OS operating system provides automatic conversion between character set and code pages for all applications that are enabled for national language support.

"Character data representation architecture design" on page 93
To enable your application for a multilingual environment, avoid coding CCSID values directly in your DDS for physical files. When database sharing takes place, you need to define your files with the CCSID of the primary language or use Unicode.

"Sort sequence tables" on page 156
A sort sequence table is an object that contains the weight of each single-byte graphic character within a specified coded character set identifier (CCSID). The system-recognized identifier for the sort sequence table object type is *TBL.

Related tasks

"Enabling the secondary language" on page 41
You must ensure that secondary languages can be used after they have been installed on the system.

Recommendations and guidelines for using CCSIDs

These recommendations are useful when you write globalized applications.

- Because the system is included with a default CCSID of 65535, character data conversions do not normally occur in applications. You should look over the CCSID information in this topic, however, because the system might need to participate in a multilingual environment, a network, or exchanging data at a later time.
- Applications implementing their own mapping scheme should use CCSID 65535, where a CCSID assignment is necessary. For example, depending on what an application does, it might need to use

CCSID 65535 for the files, or it might need to use CCSID 65535 for the jobs. Because other applications may require CCSIDs other than 65535, consider changing such applications by replacing the mapping scheme with CCSID support.

- Correctly define fields based on their usage. If fields contain application-dependent values (for example, control characters or fields that are not used as real character fields), define the fields as hexadecimal data or character fields with CCSID 65535.
- Avoid using characters that are not in the invariant character set for names and literals in programs.

Follow these guidelines when using CCSIDs:

- Use CCSIDs in multilingual applications to maintain character integrity in database files, displays, and printed data.
- You can find a suggested CCSID for a language in Language identifiers and associated default CCSIDs.
- If the QIGC system value is set on, set QCCSID as a mixed CCSID or 65535.
- If you use DBCS support, set the job CCSID to a mixed CCSID. If you do not, set the job CCSID to a single-byte CCSID.
- Ensure that the QCHRID code page is compatible with the character set and code page of the QCCSID value, unless the QCCSID value is 65535. If the QCCSID value is changed to a value that is incompatible with the current QCHRID value, the QCHRID value is changed to a compatible value by the system.
- If you use a user-defined data stream (UDDS), remove any X’3F’ values inserted by CCSID conversions. Otherwise, your data can cause the system to blank out a screen. Some CCSID conversions use a X’3F’ value for a substitution character.
- If you are using any interactive jobs, such as Application Development ToolSet/400, ensure that the code page of the job CCSID matches the code page of the keyboard type. If these CCSID values do not match, or the job CCSID is 65535, unpredictable results might occur.
- Be aware that the *JOBCCSID support is not used by any system-supplied displays or panel groups, although CHRIDCTL support is used.
- Be aware of character data that has been defined or specified as control information. For new database files, fields that contain control information should be defined as hexadecimal data type or use CCSID 65535 instead of another CCSID.
- Because of workstation hardware restrictions, you might not see all of the characters on displays other than 3486, 3487, 3488, or Personal System/2 (PS/2®) displays when CCSID conversion occurs. However, the character data is retained in the system.
- Be aware that when a CCSID conversion is performed, substitution characters might cause a loss of data. The situation occurs if enforced subset match conversion is performed.

Related concepts

“Object-level coded character set identifier 65535” on page 137
CCSID 65535 is the default object-level CCSID for message files and message queues.

“Language identifiers and associated default CCSIDs” on page 358
This table shows the language identifiers and the job default CCSID (DFTCCSID) values associated with those identifiers.

“DBCS system indicator (QIGC) system value” on page 20
The DBCS system indicator (QIGC) system value specifies whether a double-byte character set (DBCS) national language version (NLV) is installed. This value is set when the primary national language version is installed.

“National language keyboard types and SBCS code pages” on page 303
This table lists the keyboard types and code pages for each national language supported by the i5/OS operating system. The Create Device Display (CRTDEVDSP) command uses the KBDTYPE parameter.

“Conversion of character data” on page 5
The Character Data Representation Architecture (CDRA) system of tags ensures that you can convert character data in a predictable, repeatable way.

i5/OS function support for CCSIDs

The system provides CCSID support in the functions as shown in the table.

Function	Description of support
CL commands	Some control language (CL) commands have internal functions that support CCSID conversions. For more information about CL commands that support CCSID conversions, see the CL Reference topic.
Copy	Coded character set identifier (CCSID) support is built into the copy function. The Copy File (CPYF) and Copy from Query File (CPYFRMQRYF) commands support CCSIDs. To use the CPYF command to change a physical file, see Changing the CCSID of a physical file. The Copy Source File (CPYSRCF) command supports CCSID conversion.
Database management	Database management support provides default coded character set identifier (CCSID) values for database files on the system.
DDM	Coded character set identifier (CCSID) support is built into distributed data management (DDM). DDM provides support to pass CCSID tags in homogeneous environments. DDM passes a CCSID parameter when sending files. With DDM, you can also specify a CCSID when creating files on a remote system. DDM only converts data to the job CCSID of the source system when: • The i5/OS operating system is running on both the source and target systems. • The source and target systems are at an operating system level of Version 2 Release 1.1 or later. Program-described files are always created with a CCSID of 65535 if they are created: • On a target system running a release level from OS/400® Version 2 Release 1.1 through OS/400 Version 2 Release 3 • From a source system that is not running the i5/OS operating system • From a source system that is a system at a release level before OS/400 Version 2 Release 1.1 You can use the Submit Remote Command (SBMRMTCMD) command on a source system to change the file CCSID (externally described files only) by specifying the Change Physical File (CHGPF) command and the CCSID parameter.
DDS	Coded character set identifier (CCSID) support is built into data description specifications (DDS). DDS supports file-level and field-level CCSID keywords for all character fields in physical files. DDS also supports file-level and field-level keywords for all DBCS fields in physical files.

Function	Description of support
Distributed relational database	Coded character set identifier (CCSID) support is built into distributed relational database. Distributed relational database passes the CCSID of an application requester (AR) job to an application server (AS) job and vice versa during connect processing. Distributed relational database also performs a conversion of error information and text-describing fields according to the job CCSID. Distributed relational database uses CCSID information to determine how to build data exchanged between application requester jobs and application server jobs. It also uses CCSID information to describe data exchanged between application requester jobs and application server jobs (for example, a format description).
IDDU	Coded character set identifier (CCSID) support is built into interactive data definition utility (IDDU). Interactive data definition utility provides support to specify a CCSID for a character field or a DBCS field.
Open Query File (OPNQRYF)	Coded character set identifier (CCSID) support is built into i5/OS query. You can use the Open Query File (OPNQRYF) command to specify a CCSID on the MAPFLD parameter. The MAPFLD parameter specifies the definition of query fields that are either mapped to, or derived from, other fields. i5/OS query supports CCSID conversion on CHAR, OPEN, EITHER, and UCS-2 graphic field operators for join, record selection, group-by, and minimum or maximum values functions. CCSID conversion is performed whenever fields do not have the same CCSID value. After the query is opened, database management support converts data read or written to the database files as described in the Database Management topic. i5/OS query does not support CCSID conversion if at least one of the fields is assigned a CCSID of 65535.
Query management	Coded character set identifier (CCSID) support is built into query management. Query management assigns a CCSID to queries and forms. Query management: • Converts queries to the job CCSID. • Presents data to the display device using the job CCSID. • Assigns a CCSID to the files it creates.
SNA	Coded character set identifier (CCSID) support is built into SNA Distributed Services (SNADS). SNADS supports CCSIDs by any user ID, system name, or destination queue name. However, other SNADS services such as SNDNETF do not provide CCSID conversion.
Work management	Work management support provides the function to assign or change coded character set identifier (CCSID) values at three different levels. See the Work management topic for details.
Workstations	The workstation management function provides support for display files, printer files, and panel groups. See Workstation function management for details.

Function	Description of support
Message management	Coded character set identifier (CCSID) support is built into the system's message support. Use CCSID support for handling messages and message catalogs on i5/OS: • You can send messages tagged with one CCSID to users with a different CCSID. • You can use CCSID support to handle messages by using commands and application programming interfaces.

Related concepts

Changing the CCSID of a physical file

Related reference

Control language (CL)

Copy File (CPYF) command

Copy From Query File (CPYFRMQRYF) command

Copy Source File (CPYSRCF) command

Submit Remote Command (SBMRMTCMD) command

Change Physical File (CHGPF) command

Open Query File (OPNQRYF) command

Database management:

Database management support provides default coded character set identifier (CCSID) values for database files on the system. All database files are assigned a CCSID. At file creation time, the CCSID is either explicitly assigned through DDS, SQL, or IDDU, or implicitly assigned the job default CCSID (DFTCCSID).

Database files support for CCSIDs

IBM system files and licensed program database files are created with the CCSID of choice for each of the national language versions. Only the customer files are automatically assigned the CCSID of the job creating the file. You can use the Display File Description (DSPFD) command to view the CCSID of a file.

Program-described files are assigned CCSID 65535. If a CCSID is not explicitly specified on the CRTPF or CRTSRCPF command, database source files default to the job default CCSID at file creation.

If a database logical file is defined over several physical files, it is assigned a CCSID at the field level and assumes the CCSID value of the physical file. Logical files cannot be explicitly assigned a CCSID value.

Database fields and support for CCSIDs

Except for numeric database fields, database fields are supported by CCSIDs. You can use the Display File Field Description (DSPFFD) command to view the CCSID of the fields in a file.

Hexadecimal fields are assigned CCSID 65535.

An implicit CCSID value is assigned to the following fields if a CCSID was not explicitly assigned through DDS, SQL, or IDDU at file creation:
- Physical-file character
- DBCS-open
- DBCS-only

- DBCS-either
- Graphic

The implicitly assigned CCSID is the job default CCSID, or a CCSID associated with the job default CCSID.

- A character field is assigned the single-byte character set (SBCS) CCSID that is associated with the job default CCSID.
- A DBCS-open, DBCS-only, and DBCS-either field is assigned the mixed byte CCSID.
- A Graphic field is assigned the double-byte character set (DBCS) CCSID that is associated with the job default CCSID.

For example, if the job default CCSID is 5026 (which is a CCSID that identifies mixed data), an SBCS character field is assigned the SBCS CCSID associated with 5026. Thus, the CCSID for that field is 290. If there is no CCSID of the required character set type then a CCSID of 65535 is used.

Database logical-file fields are assigned a CCSID value based on their data type and the data type of the underlying physical file field.

Database management and conversion support for CCSIDs

Database management support converts non-graphic character data read from, or written to, database files using the file CCSID and the job CCSID.

- If data is being read from a database file and the CCSID of the file is the same as the job CCSID, no conversion is done.
- If data is being read from a database file and the CCSID of the file and the job CCSID are different, the data is converted to the CCSID of the job.
- If data is being written to a database file and the CCSID of the file is the same as the job CCSID, no conversion is done.
- If data is being written to a database file and the CCSID of the file and the job CCSID are different, the data is converted to match the CCSID of the file.

No conversion is performed if either the CCSID of the job or the CCSID of the database file is equal to 65535.

Related concepts

"Job attributes" on page 12
Job attributes are set at the time a job starts.

"Language identifiers and associated default CCSIDs" on page 358
This table shows the language identifiers and the job default CCSID (DFTCCSID) values associated with those identifiers.

Related tasks

"Job default coded character set identifier" on page 13
A job attribute, job default CCSID (DFTCCSID), is created for jobs with a CCSID of 65535. The DFTCCSID value is used by a system code when a CCSID other than 65535 is needed.

Related reference

Display File Description (DSPFD) command

Display File Field Description (DSPFFD) command

Work management:

The job's coded character set identifier (CCSID) value is assigned or changed at any one of these levels: job level, user profile level, or system level.

All jobs run with a CCSID value established at one of these levels:

- *Job level.* A CCSID is assigned to a job.
- *User profile level.* A CCSID is specified in a user profile and the value is assigned to all jobs run under that user profile. The CCSID can be set or changed with the Create User Profile (CRTUSRPRF) and Change User Profile (CHGUSRPRF) commands.
- *System level.* The system value QCCSID is the default CCSID for all jobs running on the system. QCCSID can be set or changed with the CHGSYSVAL and WRKSYSVAL commands.

Work management support initializes the job CCSID for an interactive job to the CCSID on the user profile when the job starts. If *SYSVAL is specified for the CCSID on the user profile, work management support gets the CCSID from the system value (QCCSID). For batch jobs, the CCSID of the current job is used as the default CCSID for the submitted job.

You can change the CCSID of a job by using the Change Job (CHGJOB) command. Make a note of the current job CCSID. You can use it later to reset the job CCSID to its original value, if necessary. The new CCSID value is reflected in the job immediately. The job DFTCCSID cannot be changed. To retrieve the CCSID or DFTCCSID for a job, use the Retrieve Job Attributes (RTVJOBA) command or the Retrieve Job Information QUSRJOBI application programming interface (API). Interactively, use the Work with Job (WRKJOB) command and select the Display Job Definition Attributes option on the Work with Job display.

Related reference

Create User Profile (CRTUSRPRF) command

Change User Profile (CHGUSRPRF) command

Change System Value (CHGSYSVAL) command

Work with System Value (WRKSYSVAL) command

Retrieve Job Attributes (RTVJOBA) command

Retrieve Job Information (QUSRJOBI) API

Work with Job (WRKJOB) command

Workstation function management:

Workstation function management involves working with display files, printer files, as well as panel group objects and user interface manager (UIM) menus.

Note: All source files on the system are tagged with a coded character set identifier (CCSID).

Display files:

When a display 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.
- User message text (identified by the MSGCON keyword in DDS) remains the same because it is assumed to be in the same CCSID as the primary source file.

At run time, the constant data is converted based on the CHRID parameter value used to create the display file object. This conversion is optional and can occur only when the CHRID is set to *JOBCCSID or indirectly with CHRIDCTL. This conversion is from the display file CCSID to the character identifier (CHRID) of the device. The field-level keyword NOCCSID (no coded character set identifier) allows the user to specify fields within the DDS that are never to be converted.

Note: To use data management support of CCSIDs, you must change source physical files tagged with CCSID 65535 to a CCSID value that is associated with the data.

CHRID parameter on the Create Display File command

The CHRID parameter on the Create Display File (CRTDSPF) command affects the conversion that occurs for the display file.

If the *JOBCCSID value is specified on the CHRID parameter of the CRTDSPF command:

- Input characters are converted from the device character identifier (CHRID) to the job CCSID.
- Character data is sent to output-capable fields and converted from the job CCSID to the device CHRID.
- Constant text from the display file is converted from the CCSID of the display file to the CHRID of the device.
- All message files are tagged with a CCSID. Message text is converted from the CCSID of the message file to the CHRID of the device. When message files are tagged with a CCSID of 65535 (the system default), it is assumed that the contents of the message files are already in the CHRID of the device. To ensure that appropriate conversions occur, you can enable CCSID support for messages.
- Message replacement data is converted from the CCSID of the job, or from the CCSID of the display file, to the CHRID of the device.
- All status messages that are tagged with a CCSID other than 65535 are converted to the CHRID of the device.
- Message text for messages on a message line or in a message subfile (identified by the ERRMSG, ERRMSGID, SFLMSG, and SFLMSGID keywords in DDS) is converted from the message file CCSID to the device CHRID.

If a specific value is specified for the CHRID parameter on the CRTDSPF command, conversion is done between the CHRID specified on the CRTDSPF command and the CHRID of the device. This conversion affects only fields defined with the CHRID DDS keyword.

If the *DEVD value is specified on the CHRID parameter of the CRTDSPF command, no conversion is performed. This is the default setting.

Migration of display files with CCSID 65535

All source files in Version 3 of the i5/OS licensed program have an implicit CCSID value of 65535. To have appropriate CCSID support, display files must be recompiled with a source file that has a CCSID value other than 65535 if either of the following conditions are true:

- The display file was originally compiled from a source file with a CCSID value of 65535.
- The display file was originally compiled before Version 2 Release 3 Modification 0 of the OS/400 licensed program.

By recompiling, the display file object is tagged and all necessary conversions take place when needed.

No conversions take place if the source files are explicitly tagged CCSID 65535.

Related concepts

"Character identifier control (QCHRIDCTL) system value" on page 15
The character identifier control (QCHRIDCTL) system value controls the type of CCSID conversion that occurs for display files, printer files, and panel groups.

"Changing the CCSID of a physical file" on page 155
You can use the Change Physical File (CHGPF) command to change the coded character set identifier (CCSID) of a physical file. However, the physical file cannot be changed under certain conditions.

"CCSID support for messages" on page 136
You can use CCSID support for handling messages and message catalogs on the i5/OS operating system by using commands and application programming interfaces. You can send messages tagged with one CCSID to users with a different CCSID.

Related reference

Create Display File (CRTDSPF) command

Related information

MSGCON (Message Constant) keyword for display files

ERRMSG (Error Message) and ERRMSGID (Error Message Identifier) keywords for display files

SFLMSG (Subfile Message) and SFLMSGID (Subfile Message Identifier) keywords for display files

Printer files:

When a printer file object is created, it is tagged with the coded character set identifier (CCSID) of the source file. Processing of the source files for printer files is the same as for display files. 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 by the device description. If the CHRID information is obtained from the device description, it is then sent to the printer.

Related reference

Create Printer File (CRTPRTF) command

User interface manager menus and panel groups:

Like display files and printer files, panel group objects and user interface manager (UIM) menus are tagged with the CCSID of the primary source file. The contents of embedded source members are converted to this CCSID.

When the panel group or UIM menu is created with *JOBCCSID specified for the CHRID parameter, conversion is performed at run time. Conversion is performed between the CCSIDs of the panel group or menu, the job, and the CHRID of the display or printer.

CCSID conversions of user interface manager menu and panel groups

The following CCSID conversions occur for displays of panel groups and UIM menus:

- Text in the panel group is converted from the panel group CCSID to the device CHRID.
- Text in the UIM menu is converted from the UIM menu CCSID to the CHRID of the device.
- Variables from the user job are converted from the job CCSID to the device CHRID.
- Variables from the job are converted from the CHRID of the device to the job CCSID.
- Online help information imported from a different panel group is converted from the imported panel group CCSID to the device CHRID.

CCSID conversions when printing UIM menus and panel groups

CCSID conversions for printed UIM menus and panel groups are shown in the following table. In this table, *xxx* and *yyy* are explicitly assigned CCSID values. For example, a printer file CHRID is explicitly assigned a value of 00697 00037. The panel group is set to *JOBCCSID. The panel group constant text is converted from the panel group primary source file tagged with CCSID 00500 to the printer file CHRID 00697 00037.

Printer file CHRID is	And the panel group or menu CCSID is xxx	or *JOBCCSID	or *DEVD
yyy	No conversion occurs for panel group constant text.	Panel group constant text is converted from panel group primary source file CCSID to yyy.	No conversion occurs for panel group constant text.
	Variables with CHRID=PNLGRP on class tag are converted from xxx to yyy.	Variables with CHRID=PNLGRP on class tag are converted from job CCSID to yyy.	No conversion occurs for variables with CHRID=PNLGRP on class tag.
	No conversion occurs for variables without CHRID=PNLGRP on class tag.	Variables without CHRID=PNLGRP on class tag are converted from job CCSID to YYY.	No conversion occurs for variables without CHRID=PNLGRP on class tag.
*JOBCCSID	No conversion occurs for panel group constant text.	Panel group constant text is converted from panel group primary source file CCSID to job CCSID.	Panel group constant text is converted from panel group primary source file CCSID to job CCSID.
	Variables with CHRID=PNLGRP on class tag are converted from XXX to job CCSID.	No conversion occurs for variables with CHRID=PNLGRP on class tag.	No conversion occurs for variables with CHRID=PNLGRP on class tag.
	No conversion occurs for variables without CHRID=PNLGRP on class tag.	No conversion occurs for variables without CHRID=PNLGRP on class tag.	No conversion occurs for variables without CHRID=PNLGRP on class tag.
*DEVD	No conversion occurs for panel group constant text.	Panel group constant text is converted from panel group primary source file CCSID to job CCSID. This conversion occurs because variables are in the job CCSID and the device CHRID is unknown.	No conversion occurs for panel group constant text.

Printer file CHRID is	And the panel group or menu CCSID is xxx	or *JOBCCSID	or *DEVD
	No conversion occurs for variables with CHRID=PNLGRP on class tag.	No conversion occurs for variables with CHRID=PNLGRP on class tag.	No conversion occurs for variables with CHRID=PNLGRP on class tag.
	No conversion occurs for variables without CHRID=PNLGRP on class tag.	No conversion occurs for variables without CHRID=PNLGRP on class tag.	No conversion occurs for variables without CHRID=PNLGRP on class tag.

CCSID support for messages:

You can use CCSID support for handling messages and message catalogs on the i5/OS operating system by using commands and application programming interfaces. You can send messages tagged with one CCSID to users with a different CCSID.

Note: You do not need a multinational character set (MNCS) when using CCSIDs for handling messages.

For example, if you do not set CCSID support on, the following message, encoded in CCSID 00037:

```
Joe, I need to see you right away!
```

appears to a user with CCSID 00500 as

```
Joe, I need to see you right away]
```

Instead of seeing an exclamation mark (!), Joe sees a right square bracket (]). If you set CCSID support on, the text in a message encoded in CCSID 00037 is converted to CCSID 00500. Both the person sending the message and the person receiving the message see identical text.

CCSID support helps preserve data integrity in messages. As you read through this information, you will see other advantages to using CCSID support for messages.

Message-handling commands for CCSIDs

The following message handling commands support CCSIDs:

- CRTMSGF (Create Message File)
- CRTMSGQ (Create Message Queue)
- CHGMSGQ (Change Message Queue)
- ADDRPYLE (Add Reply List Entry)
- CHGRPYLE (Change Reply List Entry)
- CHGMSGD (Change Message Description)
- RTVMSG (Retrieve Message)
- RCVMSG (Receive Message)
- SNDBRKMSG (Send Break Message)
- SNDMSG (Send Message)
- SNDPGMMSG (Send Program Message)
- SNDRPY (Send Reply)
- SNDUSRMSG (Send user Message)

Related concepts

"Program message design" on page 75
A message can be predefined or immediate.

"Display files" on page 132
When a display file object is created, it is tagged with the coded character set identifier (CCSID) of the source file.

"Using message catalogs" on page 107
The i5/OS operating system can use message catalogs to store messages. Messages in a message catalog are grouped as sets. Each message has a unique number within a set.

Related reference

Create Message File (CRTMSGF) command

Create Message Queue (CRTMSGQ) command

Change Message Queue (CHGMSGQ) command

Add Reply List Entry (ADDRPYLE) command

Change Reply List Entry (CHGRPYLE) command

Change Message Description (CHGMSGD) command

Retrieve Message (RTVMSG) command

Receive Message (RCVMSG) command

Send Break Message (SNDBRKMSG) command

Send Message (SNDMSG) command

Send Program Message (SNDPGMMSG) command

Send Reply (SNDRPY) command

Send User Message (SNDUSRMSG) command

Handling messages with a specific object-level CCSID:

These listed objects support CCSIDs. Each of them has an object-level CCSID.
- Message files
- Message queues
- Job message queues
- System reply lists
- History log

The object-level CCSID is the CCSID in which all the messages in that object are encoded.

Object-level coded character set identifier 65535:

CCSID 65535 is the default object-level CCSID for message files and message queues.

If an object has a CCSID of 65535, no conversions occur when adding messages to that object or when receiving messages from that object. Use CCSID 65535 if you do not want CCSID processing to occur.

CCSID 65535 is also known as *HEX.

Object-level coded character set identifier 65534

CCSID 65534 is the default object-level CCSID for job message queues, system reply lists, and the history log. If the CCSID of an object is 65534, each message in the object has its own CCSID. No conversion occurs when a message is added to the object. When a message is received, it is converted based on the CCSID stored with the message.

CCSID 65534 is also known as *MSG or *MSGD.

CCSID 65534 is the preferred setting for object-level CCSIDs. An object-level CCSID of 65534 requires fewer CCSID conversions. Fewer CCSID conversions of text result in better performance and improved data integrity.

Related concepts

"Character identifier (QCHRID) system value" on page 15
The character identifier (QCHRID) system value specifies 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.

"Database files and functions" on page 121
When you create Unicode database applications, you need to consider the implications for creating physical files, creating logical files, and for database input/output.

"DB2 for i5/OS" on page 122
When using DB2 for i5/OS applications, you need to be aware of some restrictions of Unicode and some commands.

"Recommendations and guidelines for using CCSIDs" on page 126
These recommendations are useful when you write globalized applications.

Using a specific object-level CCSID for handling messages:

If the CCSID of an object is any value other than 65535 or 65534, all messages in that object are considered encoded in that CCSID. The object-level CCSID overrides the CCSID stored with the messages.

Use this type of object-level CCSID if both of the following conditions are true:

- You expect the object to be sent messages or have message descriptions added in a CCSID different from the CCSID in which you will receive the messages or retrieve the message descriptions.
- You intend to receive the same message or retrieve the same message description many times.

If these conditions are true, set the object-level CCSID to the CCSID in which you will receive or retrieve the messages. When the system uses this type of object-level CCSID, the message text or data is converted at the time the message is sent or is added to the object. No conversion occurs when the message is received or retrieved because the text and data are already in the CCSID requested on the receive operation or retrieve operation.

Do not change system-supplied message files to use this type of object-level CCSID. Each system-supplied message description is tagged separately. No one object-level CCSID value can represent all of the message descriptions in the message file. Changing the object-level CCSID of a system-supplied message file to anything other than CCSID 65535 or CCSID 65534 might cause unpredictable results.

Message-level support:

When a message is sent to a message queue, you must communicate the CCSID of the replacement data or the immediate message text to the i5/OS operating system. Use the CCSID parameter on any of the send message commands or APIs to communicate this CCSID to the system.

The default CCSID setting in the send message commands and APIs indicate that the replacement data or immediate message text is in the CCSID of the job that is running the command or API. You can override the job default CCSID value by specifying a different CCSID value.

If the replacement data or immediate message text supplied is not in the CCSID specified, incorrect conversion results may occur. See Can I correct the CCSID of a message? if this occurs.

Determining the CCSID of a message file

To determine the CCSID of a message file, follow these steps:

1. Type

   ```
   WRKMSGD MSGF(MYLIB/MYMSGF)
   ```

   where MYLIB is the library in which the message file is stored and MYMSGF is the name of the message file.
2. Press F22 (Display list details).

Note: You can also use the Retrieve Message File Attributes (QMHRMFAT) application programming interface (API) to determine the CCSID of a message file.

For job message queues, system reply lists, and the history log, the object-level CCSID is always 65534. You cannot change nor display object-level CCSIDs for job message queues, system reply lists, and the history log.

Related concepts

"Can I correct the CCSID of a message?" on page 154
You cannot correct the message-level CCSID of a message. You can change the message queue CCSID to match the message-level CCSID. You can also delete the message and send it again with the correct message-level CCSID.

Message-level CCSID with a message queue CCSID of 65535 or 65534:

When a message is sent to the message queue and the CCSID of the message queue is 65535 or 65534, no conversion occurs on the message. The message-level CCSID is set to the CCSID specified.

For example, message queue MYMSGQ has a CCSID of 65534. You enter the following Send Message command:

```
SNDMSG MSG('MSG #1') CCSID(37) TOMSGQ(MYLIB/MYMSGQ)
```

The immediate message text, MSG #1, is not converted when added to the message queue. The message is tagged with CCSID 00037.

Message-level CCSID with a specific message queue CCSID:

When a message is sent to the message queue and the CCSID of the message queue is something other than 65535 or 65534, the replacement data or immediate message text is converted to the CCSID of the message queue. The message is then tagged with the CCSID of the message queue.

For example, message queue MYMSGQ has a CCSID of 00277. The replacement data for TST0002 is defined as *CCHAR data. You enter the following Send Program Message command:

```
SNDPGMMSG MSGDTA(X'0006D4E2C7407BF2') MSGID(TST0002) MSGF(MYMSGF)
  CCSID(37) TOMSGQ(MYLIB/MYMSGQ)
```

The replacement data is converted from CCSID 00037 to CCSID 00277 before it is sent to the message queue. X'0006' is the length required for variable-length fields. X'D4E2C7407BF2' is *MSG #2* on code page 00037. The number sign (#), X'7B' on code page 00037, is converted to a number sign, X'4A' on code page 00277. All other code points do not change during the conversion because they are the same on both code page 00037 and code page 00277.

When the replacement data or immediate message text of a message is 65535 and it is sent to a message queue with a CCSID other than 65535 or 65534, no conversion occurs. However, the message is tagged with the CCSID of the message queue. Therefore, messages can be tagged with an incorrect CCSID when you send them to a message queue with a CCSID that overrides the message-level CCSID.

For example, message queue MYMSGQ has a CCSID of 00277. You enter the following Send Message command:

```
SNDMSG MSG('MSG #2') TOMSGQ(MYLIB/MYMSGQ) CCSID(*HEX)
```

The immediate message text *MSG #2* is not converted before it is sent to the message queue. Although the immediate message text is not converted to CCSID 00277, it is displayed using CCSID 00277. Unless you entered the Send Message command from a device configured to support code page 00277, you lost the integrity of the immediate message text.

Message-level CCSID when a message queue CCSID conversion error occurs:

If a conversion error occurs while sending a message to a message queue, the message is still sent to the message queue. However, the immediate text or data of the message is not converted.

A diagnostic message is sent and the message is tagged with the message-level CCSID specified on the send command or API, not with the CCSID of the message queue.

You can recover the replacement data or immediate message text with the proper CCSID setting. First, set the message queue CCSID to 65534. Then use the Receive Message command or API to return the correct message-level CCSID.

Message-level CCSID when a message is a stored message:

If a message is a stored message, the message-level CCSID applies only to *CCHAR replacement data. The CCSID of the first- and second-level text of the message is retrieved from the message file.

Replies to stored messages are never converted from one CCSID to another. Only replies to immediate messages are affected by CCSID processing.

Message description-level support:

When a message description is added to a message file, the CCSID of the message text must be communicated to the i5/OS operating system.

You can use the CCSID parameter on the Add Message Description (ADDMSGD) or the Change Message Description (CHGMSGD) command to communicate this CCSID to the operating system.

The default settings of these commands indicate that the message text is in the CCSID of the job that is running the command. You can change this value by specifying a different CCSID value. You can also change this value by indicating that no CCSID processing should occur. You indicate that no CCSID processing should occur on the message text by specifying a CCSID value of 65535 (*HEX).

If you set CCSID processing on, system-supplied display files and printer files that display or print message descriptions convert the CCSID of the message file to the CCSID of the job before displaying them or printing them. To print and display the messages correctly, your job CCSID setting must be the same as the code page portion of your device CHRID setting.

All message descriptions that existed in a message file that was created before V3R1 are tagged with CCSID 65535 on the first use or handling of that message description.

If the text of a message is not in the CCSID specified, incorrect conversion results might occur. See Can I correct the CCSID of a message description? if this occurs.

Related concepts

“Can I correct the CCSID of a message?” on page 154
You cannot correct the message-level CCSID of a message. You can change the message queue CCSID to match the message-level CCSID. You can also delete the message and send it again with the correct message-level CCSID.

Related reference

Add Message Description (ADDMSGD) command

Change Message Description (CHGMSGD) command

Message file with a CCSID of 65535 or 65534:

If the CCSID of the message file is 65535 or 65534, no conversion occurs on the message description when it is added to the file. The message description CCSID is set to the CCSID specified on the ADDMSGD or CHGMSGD command.

For example, a message file MYMSGF has a CCSID of 65534. The job that is running is in CCSID 00037. You enter an ADDMSGD command, as follows:

```
ADDMSGD MSG('MSG #1') MSGID(TST0001) MSGF(MYMSGF)
```

The message text, MSG #1, is not converted when added to the message file. The message text is tagged 00037 because the CCSID parameter was not coded on the ADDMSGD command and the default CCSID parameter is *JOB.

Related reference

Add Message Description (ADDMSGD) command

Change Message Description (CHGMSGD) command

Message file with a specific CCSID:

If the CCSID of the message file is something other than 65535 or 65534, the first- and second-level text of the message description is converted from the CCSID specified to the CCSID of the message file. It is then tagged with the CCSID of the message file.

For example, message file MYMSGF has a CCSID of 00277. The job that is running is in CCSID 00037. You enter the following command:

```
ADDMSGD MSG('MSG #2') MSGID(TST0002) MSGF(MYMSGF) CCSID(37)
```

Message ’MSG #2’ is converted from CCSID 00037 to CCSID 00277 before it is added to the message file. The number sign (#), X’7B’ on code page 00037, is converted to the number sign (#), X’4A’, on code page 00277. No other code points change during the conversion because they are the same on both code page 00037 and code page 00277.

When the text of a message description is specified as 65535 and it is added to a message file, no conversion occurs. If the CCSID of the message file is not 65535 or 65534, the message text is tagged with the CCSID of the message file.

When the message file CCSID is not 65535 or 65534, the message file CCSID overrides message description CCSIDs. Keep this rule in mind when adding and changing message descriptions to a message file with a CCSID other than 65535 or 65534. Otherwise, a message description can be marked incorrectly.

For example, message file MYMSGF has a CCSID of 00277. You enter the following command:

```
ADDMSGD MSG('MSG #2') MSGID(TST0002) MSGF(MYMSGF) CCSID(*HEX)
```

Message text ’MSG #2’ is not converted before it is added to the message file. Because the CCSID of the message file is 00277, the message text is tagged with CCSID 00277.

If the command was run in a job CCSID where the number sign (#) occupies a code point different than the code point for the number signon code page 00277, the message is displayed incorrectly.

A conversion error may occur while adding or changing a message description in a message file. If a conversion error occurs, the message description is still either added to or changed in the message file. The text of the message description, however, is not converted. A diagnostic message is sent and the message description is tagged with the CCSID specified, not with the CCSID of the message file.

When a conversion error occurs, you can recover the correct CCSID tagging for the message description by setting the message file CCSID to 65534. Then you can retrieve the correct CCSID for the message description using the Retrieve Message (RTVMSG) command or the Retrieve Message (QMHRTVM) API.

The CCSID of a message description applies only to first- and second-level message text.

Related reference

Retrieve Message (RTVMSG) command

Retrieve Message (QMHRTVM) API

Changing the CCSID of a message description:

When you take the option to change a message description from the Work with Message Descriptions display, all current values for the selected message description are retrieved and placed on the prompt display.

The first- and second-level text are converted from the CCSID of the message file to the CCSID of the job before they are put on the prompt display.

*JOB is displayed for the CCSID keyword and has two different meanings depending on what you do on the prompt display. If you change any part of the first- or second-level text, *JOB means that the text is converted from the CCSID of the job to the CCSID of the message file when you press the Enter key. If the text is unchanged, *JOB works like *SAME, and none of the following texts are changed:
- The first-level message text
- The second-level message text
- The CCSID of the message description

Both the first- and second-level text of a message description must be in the same CCSID. If you change the CCSID of one level, the system automatically converts the other level to match.

Example: Changing a message description

The CCSID of message file MYMSGF is 65534. The CCSID of the job that is running WRKMSGD is 00277. The CCSID of the message description is 00037.

Select option 2 to change a message description. The text of the message description is converted from CCSID 00037 to 00277 before being placed on the prompt display.

If only the first-level text is changed, the 00277-tagged text is stored in the message file. The CCSID of the message description is changed to 00277. The 00277-tagged second-level text is also stored in the message file to keep both the first- and second-level text in the same CCSID.

Message queues:

If you set CCSID processing on, system-supplied display files and printer files that display or print messages convert the CCSID of the message queue to the CCSID of the job before displaying or printing the messages.

To print and display the messages correctly, your job CCSID setting must be the same as the code page portion of your device CHRID setting.

All messages that existed on a message queue that was created in a release before V3R1 of the operating system are assigned CCSID 65535 on the first use of that message.

Determining the CCSID of a message queue

To determine the CCSID of a message queue, follow these steps:

1. Type
   ```
   DSPMSG MSGQ(MYLIB/MYMSGQ) ASTLVL(*BASIC)
   ```
   where MYLIB is the library in which the message queue is stored and MYMSGQ is the name of the message queue.
2. Press F22 (Display list details).

Note: You can also use the Retrieve Message Queue Attributes (QMHRMQAT) application programming interface (API) to determine the CCSID of a message queue.

For job message queues, system reply lists, and the history log, the object-level CCSID is always 65534. You cannot change nor display object-level CCSIDs for job message queues, system reply lists, and the history log.

Related concepts

"Job message queues"
The CCSID for all job message queues is 65534. You cannot change or display this value. A job message queue CCSID of 65534 requires fewer CCSID conversions. Fewer CCSID conversions of text result in better performance and improved data integrity.

"System reply list" on page 144
The system reply list has a CCSID of 65534. You cannot change or display this value.

Related tasks

"History log" on page 145
The history log is a database file that is tagged with CCSID 65535. You cannot change the CCSID of the history log. No conversions occur when you do database retrievals from the history file.

Related reference

Retrieve Nonprogram Message Queue Attributes (QMHRMQAT) API

Job message queues:

The CCSID for all job message queues is 65534. You cannot change or display this value. A job message queue CCSID of 65534 requires fewer CCSID conversions. Fewer CCSID conversions of text result in better performance and improved data integrity.

The CCSID of each message in the job log is used for CCSID processing. No conversion occurs when a message is sent to the job log.

Note: Request messages are always tagged with a CCSID of 65535 and are never converted.

If you set CCSID processing on, system-supplied display files and printer files that display or print job logs convert the CCSID of the messages to the CCSID of the job before displaying or printing the messages. To print and display the messages correctly, your job CCSID setting must be the same as the code page portion of your device CHRID setting. Status messages that appear on line 24 of a display are converted to the CCSID of the device before they are shown.

Related concepts

"Message queues" on page 142
If you set CCSID processing on, system-supplied display files and printer files that display or print messages convert the CCSID of the message queue to the CCSID of the job before displaying or printing the messages.

Related tasks

"History log" on page 145
The history log is a database file that is tagged with CCSID 65535. You cannot change the CCSID of the history log. No conversions occur when you do database retrievals from the history file.

System reply list:

The system reply list has a CCSID of 65534. You cannot change or display this value.

The only part of the system reply list that is affected by CCSID processing is the Compare data field. If the Compare data field references replacement data that is defined as *CCHAR, the data being compared must be in a common CCSID before the comparison is done.

Any reply list entry that has compare data is tagged with the CCSID supplied on the ADDRPYLE or CHGRPYLE commands. When the system reply list is used, the replacement data is converted to the CCSID of the compare data before the comparison is made and before the message is sent to the message queue. This ensures that the data is in a common CCSID before the comparison is done.

Example: System reply list and converted-character compare data

Enter the following Add Reply List Entry command:

```
ADDRPYLE SEQNBR(101) MSGID(TST0010) CMPDTA(X'00017B') RPY(*DFT) +
CCSID(37)
```

X'7B' is the number sign (#) on code page 00037. X'0001' is the length required for variable-length fields. The compare data is not converted when added to the system reply list. It is tagged with CCSID 00037. Message TST0010 has one replacement data field that is defined as *CCHAR with (*VARY 2) for its length. Message queue MYMSGQ has a CCSID of 00278.

Send message TST0010 in a job that has the system reply list turned on using the following Send Program Message command:

```
SNDPGMMSG MSGID(TST0010) MSGF(MYLIB/MYMSGF) MSGTYPE(*INQ) +
TOMSGQ(MYLIB/MYMSGQ) MSGDTA(X'00014A') CCSID(277)
```

The replacement data is converted from CCSID 00277 to CCSID 00037 and then compared with the compare data. The conversion results in replacement data X'00017B'. A match is found and the default reply is sent when this message is added to the message queue.

When the message is added to the message queue, the replacement data is converted from CCSID 00277 to CCSID 00278. The message queue CCSID does not matter when trying to match the compare data. The replacement data is converted to X'000163' when it is sent to the message queue and tagged 00278. X'63' is the code point for the number sign (#) in code page 00278.

Related concepts

"Message queues" on page 142
If you set CCSID processing on, system-supplied display files and printer files that display or print messages convert the CCSID of the message queue to the CCSID of the job before displaying or printing the messages.

Related reference

Add Reply List Entry (ADDRPYLE) command

Change Reply List Entry (CHGRPYLE) command

History log:

The history log is a database file that is tagged with CCSID 65535. You cannot change the CCSID of the history log. No conversions occur when you do database retrievals from the history file.

You can use CCSID processing when working with the history log. The CCSID of the replacement data or immediate message text is added to the history log record. If the history log record is for a stored message, CCSID processing occurs only for the *CCHAR replacement data in that record.

You can retrieve a message from the history log and convert it into a specific CCSID by following these steps:

1. Obtain the input variables &MSGFL, &MSGF, &MSGID, &MSGDTA, and &MDTACCSID, from the history log record. (See the CL Programming PDF for the layout of the history log record.)
2. Enter the following Retrieve Message command:
   ```
   RTVMSG MSGF(&MSGFL/&MSGF); MSGID(&MSGID); MSGDTA(&MSGDTA);  +
   MDTACCSID(&MDTACCSID); MSG(&MSG);
   ```

If you set CCSID processing on, system-supplied display files and printer files that display or print history log records convert the CCSID of the messages to the CCSID of the job before displaying or printing the messages. To print and display the messages correctly, your job CCSID setting must be the same as the code page portion of your device CHRID setting.

Related concepts

“Message queues” on page 142

If you set CCSID processing on, system-supplied display files and printer files that display or print messages convert the CCSID of the message queue to the CCSID of the job before displaying or printing the messages.

“Job message queues” on page 143

The CCSID for all job message queues is 65534. You cannot change or display this value. A job message queue CCSID of 65534 requires fewer CCSID conversions. Fewer CCSID conversions of text result in better performance and improved data integrity.

Related information

CL Programming

Setting up CCSID support for message handling:

The default setting of the CCSID for creating message queues and message files is 65535. Most message files delivered with the operating system have a CCSID of 65535.

Most message descriptions in system-supplied message files are tagged with a CCSID that corresponds to the national language version with which they are included.

Some message descriptions are not assigned a CCSID that corresponds to the national language version. These message descriptions are tagged 65535 and are not converted when used.

Messages sent to a message queue that has a CCSID of 65535 are not converted when placed on the queue. Message descriptions added to a message file that has a CCSID of 65535 are not converted when placed in the file. These messages and message descriptions are tagged with a CCSID associated with their text or data. By tagging them with a CCSID associated with their text or data, they are given the correct CCSID if the object-level CCSID is changed to 65534.

You can set CCSID support on for handling a specific message queue. For example, to set CCSID handling on for message queue MYMSGQ in library MYLIB, type:

```
CHGMSGQ MSGQ(MYLIB/MYMSGQ) CCSID(65534)
```

The Change Message Queue (CHGMSGQ) command also allows you to turn on CCSID support for more than one message queue at a time.

You can set CCSID support on for handling a specific message file. For example, to set CCSID handling on for message file MYMSGF in library MYLIB, type:

```
CHGMSGF MSGF(MYLIB/MYMSGF) CCSID(65534)
```

The Change Message File (CHGMSGF) command also allows you to turn on CCSID support for more than one message file at a time.

Related reference

Change Message Queue (CHGMSGQ) command

Change Message File (CHGMSGF) command

CCSID support for message catalogs:

The Message catalog CCSID (CLGCCSID) parameter allows you to specify the CCSID for storing data in a message catalog.

The Source file CCSID (SRCCCSID) parameter allows you to specify the CCSID of a source file. Data from the source is converted to the CCSID of the message catalog if the CCSIDs for both are not the same. This is also the default action. The source can be in any CCSID that supports conversion to any other CCSID.

The CCSID of the original message catalog is used to update the message catalog. It can be single or mixed and in extended binary-coded decimal interchange code (EBCDIC), American National Standard Code for Information Interchange (ASCII), or UCS-2. If the catalog is a QSYS source file member that does not exist, the CCSID of the existing file is used. The value that is specified on the CLGCCSID parameter is used if the CCSID of the file is 65535.

Converted character replacement data type field:

A replacement data type field supports CCSID processing. This replacement data type field is called a convertible character field (*CCHAR). A *CCHAR replacement data type field is a variable-length field, and therefore the field might increase or decrease in length when it is converted.

Example: Adding a message description with CCSID support

The following example shows how to add the message description TST0006 to message file MYMSGF. The message description has 2 replacement data type fields. One field is a character field length 10. The other field is a convertible character field with varying length. Use the ADDMSGD command as follows:

```
ADDMSGD MSG('This is *CHAR &1;  This is *CCHAR &2;') MSGID(TST0006) +
MSGF(MYLIB/MYMSGF) FMT((*CHAR 10) (*CCHAR *VARY 2))
```

Related reference

Add Message Description (ADDMSGD) command

Retrieving messages:

The Retrieve Message (RTVMSG) command and retrieve message (QMHRTVM) application programming interface (API) have a CCSID-to-convert-to parameter. This parameter determines which CCSID the first- and second-level text is converted to before the text is returned to the user.

The Retrieve Message command and the Retrieve Message API also have a replacement data CCSID parameter. This parameter communicates the CCSID of the replacement data to the system. The replacement data CCSID applies only to the parts of the replacement data that correspond to *CCHAR type data. No other replacement data is converted.

The Retrieve Message command and Retrieve Message API convert the first- and second-level text from the CCSID of the message file to the CCSID on the CCSID-to convert-to parameter. Any replacement data that is *CCHAR data is converted from the replacement data CCSID to the CCSID-to-convert-to CCSID before being substituted into the correct replacement variables. The default for both parameters is *JOB, which means that the CCSID of the job is used.

Retrieve Message command CCSID return fields

Three CCSID return fields are supported by the Retrieve Message (RTVMSG) command:

- TXTCCSID
- TXTCCSTA
- MDTACCSTA

Example 1: Retrieving a message with CCSID support

Message file MYMSGF has a CCSID of 65534. The CCSID of the message description is used to determine the CCSID from which to convert the message text. The CCSID of the message description (TST0003) is 00037. The first-level text is:

```
'MSG #3 is &1;'
```

&1 is defined as a *CCHAR variable field with a length of (*VARY 2). Enter the following RTVMSG (Retrieve Message) command:

```
RTVMSG MSGF(MYMSGF) MSGID(TST0003) MSG(&MSG); CCSID(277)  +
MDTACCSID(277) MSGDTA(X'0002D6D2')
```

In the message data, the first 2 bytes are a length field with a value of 2. All *VARY fields begin with a length. The next 2 characters are the actual *CCHAR data with a value of X'D6D2'. X'D6D2' represents the characters *O* and *K* on code page 00277.

The first-level text is converted from CCSID 00037 to CCSID 00277. The replacement data is not converted before it is substituted for &1; because the replacement data CCSID matches the CCSID-to-convert-to parameter. As a result, the text returned in the variable &MSG is:

```
'MSG #3 is OK.'
```

The code point for the number sign (#) is the only change that occurred in the conversion. The number sign was converted from code point X'7B' in code page 00037 to code point X'4A' in code page 00277. All other code points in the text of the message matched in code page 00037 and code page 00277.

Note: If the CCSID of a message file is 65535, no conversion occurs, even though the message description CCSID is 00037. The CCSID of the message file always takes precedence over the message description CCSID.

Example 2: Using return fields and converted character data

Message description TST0005 has the following first-level text:

```
This is *CHAR &1;  This is *CCHAR &2;
```

The message description is defined in message file MYMSGF, which has a CCSID of 65535. &1; is defined as a *CHAR field of length 1. &2; is defined as a *CCHAR field (*VARY 2) in length. The CCSID of the message description does not matter because the CCSID of the message file is not 65534. You enter the following RTVMSG command:

```
RTVMSG MSGF(MYMSGF) MSGID(TST0005) MSG(&MSG); CCSID(260)  +
MDTACCSID(37) MSGDTA(X'5A00015A') TXTCCSID(&TXTCCSID);
```

Note: X'5A' is the exclamation point (!) on code page 00037.

These are the returned values from the RTVMSG command:

- &MSG = 'This is *CHAR. This is *CCHAR !.'

  The EBCDIC value of the *CHAR character is X'5A'. X'5A' appears as an acute accent (´) on code page 00260. The *CHAR data did not convert because only *CCHAR data supports CCSID processing. The '&1' stayed at X'5A', while '&2' converted to X'4F'. X'4F' is the exclamation point on code page 00260.
- &TXTCCSID = 65535

  The TXTCCSID variable is set to 65535 because no conversion occurred. When no conversion occurs, the CCSID (if it is not 65534) of the message file is returned.

  Related reference

  Retrieve Message (RTVMSG) command

  Retrieve Message (QMHRTVM) API

CCSID of the text returned (TXTCCSID) return field:

TXTCCSID is the CCSID of the text returned.

If a conversion occurs and is successful, this value is always equal to the CCSID-to-convert-to value. If a conversion occurs and is not successful, this is the CCSID of the message file unless the CCSID of the message file is 65534. If the CCSID of the message file is 65534, the CCSID of the message description is returned.

For example, message file MYMSGF has a CCSID of 65534. Your program needs to know the CCSID of message description TST0004. Specify the RTVMSG command as follows:

```
RTVMSG MSGF(MYMSGF) MSGID(TST0004) CCSID(*HEX)
TXTCCSID(&TXTCCSID);
```

The CCSID of the message description is returned in the variable &TXTCCSID because you specified *HEX for the CCSID-to-convert-to parameter. *HEX means no conversion is to occur. If no conversion occurs and the message file CCSID is 65534, the message description CCSID is returned.

You can also obtain the message description CCSID from the Work with Message Descriptions (WRKMSGD) display.

1. On the WRKMSGD display, select option 5 to display details.
2. From the Select Message Details to Display menu, select option 5 to display message attributes.
3. Page forward to the CCSID value.

   The message description CCSID is shown if the CCSID of the message file is 65534. If the CCSID of the message file is not 65534, the CCSID of the message file is shown.

   Related reference

   Work with Message Descriptions (WRKMSGD) command

CCSID conversion status indicator (TXTCCSTA) return field:

TXTCCSTA is the text CCSID conversion status indicator. Return codes help you determine what happened when the system converted your message text to the CCSID-to-convert-to parameter.

Positive return code numbers indicate that your conversion was successful. A successful return code does not always indicate that a conversion occurred. Negative return code numbers indicate that a conversion error occurred.

The following list shows the available return codes:

0 No conversion was necessary. The CCSID of the text matched the CCSID that you wanted the text converted to.

1 No conversion occurred. Either the text was 65535 or the CCSID that you wanted the text converted to was 65535.

2 No conversion occurred. You did not ask for any text to be returned.

3 The text was converted to the CCSID specified. The conversion operation used the linguistic conversion tables.

4 A conversion error occurred when the conversion operation used the linguistic conversion tables. The conversion operation then used a default conversion table. The default conversion completed without error.

-1 An error occurred on both the linguistic and default conversions. The text was not converted.

Related concepts

"Replacement data CCSID conversion status indicator (MDTACCSTA) return field"
MDTACCSTA is the replacement data CCSID conversion status indicator. Return codes help you determine what happened when the system converted your replacement data to the CCSID-to-convert-to parameter.

Replacement data CCSID conversion status indicator (MDTACCSTA) return field:

MDTACCSTA is the replacement data CCSID conversion status indicator. Return codes help you determine what happened when the system converted your replacement data to the CCSID-to-convert-to parameter.

Positive return code numbers indicate that your conversion was successful. A successful return code does not always indicate that a conversion occurred. Negative return code numbers indicate that a conversion error occurred. These return codes are similar to the TXTCCSTA return codes. The return codes apply to the conversion that takes place on any *CCHAR replacement data being converted from the replacement data CCSID to the CCSID-to-convert-to value.

The following list shows the available return codes:

0 No conversion was necessary. The CCSID of the replacement data matched the CCSID that you wanted the text converted to.

1 No conversion occurred. Either the replacement data was 65535 or the CCSID that you wanted the replacement data converted to was 65535.

2 No conversion occurred. Either you did not ask for any replacement data to be returned or no *CCHAR replacement data fields were defined for the message description being retrieved.

3 The replacement data was converted to the CCSID specified. The conversion operation used the linguistic conversion tables.

4 A conversion error occurred when the conversion operation used the linguistic conversion tables. The conversion operation then used a default conversion table. The default conversion completed without error.

-1 An error occurred on both the linguistic and default conversions. The replacement data was not converted.

Related concepts

"CCSID conversion status indicator (TXTCCSTA) return field" on page 148
TXTCCSTA is the text CCSID conversion status indicator. Return codes help you determine what happened when the system converted your message text to the CCSID-to-convert-to parameter.

Receiving messages:

The Receive Message (RCVMSG) command, the Receive Nonprogram Message (QMHRCVM) API, and the Receive Program Message (QMHRCVPM) API have a CCSID-to-convert-to parameter. This parameter determines which CCSID the text or data is converted to before it is returned to the user.

The Receive Message command and APIs convert the text or data from the CCSID of the message queue or message file to the CCSID supplied on the CCSID-to-convert-to parameter. When replacement data is returned, only the *CCHAR data is converted from the CCSID of the message queue to the CCSID-to-convert-to value.

If the CCSID of the message file or message queue is 65534, the text or data is converted from the CCSID of the message description or message to the CCSID supplied on the CCSID-to-convert-to parameter.

The default for the CCSID-to-convert-to parameter is *JOB, which means that the CCSID of the job performing the receive operation is used.

Receive Message command CCSID return fields

Two CCSID return fields are supported by the Receive Message (RCVMSG) command:
- TXTCCSID
- DTACCSID

Receive Message API CCSID return fields

The Receive Message (QMHRCVM) API and the Receive Program Message (QMHRCVPM) API support the return fields defined in TXTCCSID return field for receive message command and DTACCSID return field for receive message command. The Receive Message API and the Receive Program Message API also support two additional return fields.

Example 1: Using the CCSID return fields

Message description TST0005 has the following first-level text:

```
This is &CHAR &1;  This is *CCHAR &2;
```

'&1' is defined as a *CHAR field of length 1. '&2' is defined as a *CCHAR field (*VARY 2) in length.

Message file MYMSGF has a CCSID of 65534. TST0005 is defined in message file MYMSGF. The message description CCSID is 65535. The CCSID of message queue MYMSGQ is 65534.

You enter the following Send Program Message command:

```
SNDPGMMSG MSGF(MYMSGF) MSGID(TST0005) CCSID(37) TOMSGQ(MYLIB/MYMSGQ) +
MSGDTA(X'7B00017B')
```

The message is not converted when it is sent to message queue MYMSGQ because the message queue CCSID is 65534. The message is tagged with CCSID 00037.

You enter the following Receive Message command to receive the message just sent:

```
RCVMSG MSGQ(MYLIB/MYMSGQ) MSG(&MSG); DTACCSID(&DTACCSID); +
CCSID(277) MSGDTA(&MSGDTA); TXTCCSID(&TXTCCSID);
```

Note: X'7B' is the number sign (#) on code page 00037.

Because the message description is tagged 65535, no conversion is performed when retrieving the message text of TST0005. The replacement data is tagged 00037. The *CCHAR part of the message data is converted from CCSID 00037 to CCSID 00277 before being inserted for &2; *CHAR data is never converted.

The following table shows the returned values after the Receive Message command runs:

Value	Description
&MSG =	This is &CHAR Æ. This is *CCHAR #. The *CHAR data was not converted when substituted for &1; The *CHAR data remains X'7B'. X'7B' is the code point on code page 00277 for A ligature (Æ). The *CCHAR data was converted to X'4A' before it was substituted for &2; X'4A' is the code point on code page 00277 for the number sign (#).
&TXTCCSID = 65535	The &TXTCCSID variable was set to 65535 because no conversion occurred. When no conversion occurs, the CCSID of the message description is returned if the CCSID of the message file is 65534.
&DTACCSID = 00277	The &DTACCSID variable was set to 00277 because a conversion occurred.

Example 2: Receiving a message with CCSID support

Message file MYMSGF has a CCSID of 00037. Message queue MYMSGQ has a CCSID of 65534. The message being received has a message-level CCSID of 00277. CCSID 65534 uses the message-level CCSID when determining the CCSID the replacement data is to be converted from.

The message being received is a stored message. The stored message has *CCHAR replacement data. The CCSID of the job is 00278. You enter the following Receive Message command:

```
RCVMSG MSGQ(MYMSGQ) MSG(&MSG); MSGDTA(&MSGDTA);
```

The first-level text of the stored message that you receive is converted from CCSID 00037 to CCSID 00278. The replacement data of the message that you receive is converted from CCSID 00277 to CCSID 00278. Then the replacement data is substituted into the first-level text and returned in &MSG.

Both the first-level text and the replacement data of the message that you received are converted to the CCSID of the job because the CCSID of the job is the default for the CCSID-to-convert-to parameter.

Two different conversions must occur because only the replacement data is stored in the message queue for stored messages. The text of a stored message must be retrieved from the message file. If the message contained other replacement data type fields that were not defined as *CCHAR, the non-*CCHAR data is not converted before being returned.

Note: If the CCSID of the message queue is 00278, no conversion occurs on the replacement data before the message is returned, even though the message CCSID is 00277. Remember that the message queue CCSID takes precedence over the message-level CCSID.

Related reference

Receive Message (RCVMSG) command

Receive Nonprogram Message (QMHRCVM) API

Receive Program Message (QMHRCVPM) API

CCSID of the message text returned (TXTCCSID) return field:

TXTCCSID is the CCSID of the message text returned. If a conversion occurs and the conversion is successful, this value is always the same as the CCSID-to-convert-to value.

For immediate text, if the conversion is not successful, TXTCCSID is the CCSID of the message queue, unless the message queue is 65534. If the message queue is 65534, TXTCCSID is the message-level CCSID of the immediate text.

For a stored message, if the conversion is not successful, TXTCCSID is the CCSID of the message file that contains the stored message, unless the message file is 65534. If the CCSID of the message file is 65534, TXTCCSID is the CCSID of the message description for the stored message.

CCSID of the replacement data returned (DTACCSID) return field:

DTACCSID is the CCSID of the replacement data returned. DTACCSID applies only to those parts of the replacement data defined as *CCHAR.

If the message being received is an immediate message, a value of 0 is returned. If a conversion occurs and the conversion is successful, this value is the same as the CCSID-to-convert-to value.

If the conversion is not successful, the DTACCSID returned is the CCSID of the message queue, unless the CCSID of the message queue is 65534. If the CCSID of the message queue is 65534, the DTACCSID returned is the CCSID of the message.

For example, a stored message TST0004 from message file MYMSGF is sent to message queue YOURMSGQ with replacement data. TST0004 is defined with *CCHAR replacement data. Message file MYMSGF is 65534. Message queue YOURMSGQ has a CCSID of 00037.

Your program needs to know the CCSID of the message description and the replacement data sent to message queue YOURMSGQ. You enter the following Receive Message command:

```
RCVMSG MSGQ(YOURMSGQ) CCSID(*HEX) TXTCCSID(&TXTCCSID); DTACCSID(&DTACCSID);
```

The message description CCSID is returned in the variable &TXTCCSID. The message description CCSID is returned because you specified *HEX for the CCSID-to-convert-to parameter. *HEX means that no conversion is to occur. If no conversion occurs and the message file CCSID tag is 65534, the CCSID of the message description is returned.

The CCSID of message queue YOURMSGQ (00037) is returned in the variable &DTACCSID. The message queue CCSID is returned because it is not 65534.

You can also obtain the message-level CCSID using the Display Messages (DSPMSG) display.

1. From the Display Messages display, press Help to display the Additional Message Information display.
2. Press F9 (Display Message Details).

   This displays the message-level CCSID when the CCSID of the message queue that this message is on is 65534. Otherwise, the CCSID of the message queue is displayed.

Common questions about CCSID support for handling messages:

Here are some common questions asked about CCSID support for handling messages.

When is the job default CCSID used for handling messages?:

A job default CCSID is always a CCSID with an encoding scheme of 1100 (single-byte EBCDIC) or 1301 (mixed-byte EBCDIC).

The job default CCSID is used whenever both of the following conditions are true:

- A conversion occurs from a CCSID with an encoding scheme other than 1100 or 1301 to a job CCSID.
- The job CCSID is 65535.

For example, ASCII data must be converted to a CCSID associated with the data when asked to convert to the CCSID of a job. The job default CCSID is used because it is never CCSID 65535.

*How can I determine if a message description is defined with *CCHAR?:*

You can use the Work with Message Description (WRKMSGD) command to determine if a message description is defined with *CCHAR data. You can also use the Retrieve Message (QMHRTVM) API to return the replacement data format fields.

Related reference

Work With Message Description (WRKMSGD) command

Application programming interfaces (APIs)

*Can the length of *CCHAR replacement data change?:*

The length of *CCHAR replacement data can change. This is why *CCHAR replacement data is required to be a variable-length field.

The length of the field grows when converting from an SBCS CCSID to the UCS-2 Level-1 CCSID. The length of the field shrinks when converting from the UCS-2 Level-1 CCSID to an SBCS CCSID.

For example, you define message description TST0011 as 'Printer &1; has error &2;' in message file MYMSGF that has a CCSID of 65535. '&1' is defined as *CCHAR data (*VARY 2) in length. This is the name of the printer. &2; is defined as a *CHAR data with a length of 1. This is an error code. Enter the following Send Program Message command to send this message to message queue MYMSGQ:

```
SNDPGMMSG MSGID(TST0011) MSGF(MYLIB/MYMSGF) TOMSGQ(MYLIB/MYMSGQ) +
MSGDTA(X'000400500030F1') CCSID(61952)
```

X'0004' is the length of the variable *CCHAR data. X'00500030' represents the characters P0 in CCSID 61952. If message queue MYMSGQ has a CCSID of 00037, the replacement data is converted to X'0002D7F0F1' before it is sent to the message queue. If message queue MYMSGQ has a CCSID of 65535, the data is not converted when it is sent to the message queue.

Your application programs cannot hard-code the position of the return code in this example. When message queue MYMSGQ has a CCSID of 00037, the return code is 5 bytes into the message text. When message queue MYMSGQ has a CCSID of 65535, the return code is 7 bytes into the message text.

Can I correct the CCSID of a message queue?:

You might have a message queue that has a CCSID that does not match the CCSID of the messages on it. This typically results from sending messages with a message-level CCSID of 65535 to a message queue with a CCSID that is not 65534 or 65535.

If all of the messages on a message queue have the same message-level CCSID and you know the message-level CCSID, you can enter the following command:

```
CHGMSGQ MSGQ(MYMSGQ) CCSID(nnnnn)
```

In this example, MYMSGQ is the name of the message queue and nnnnn is the message-level CCSID.

If you do not know the CCSID of all the messages on the queue or if the messages on the queue have different CCSIDs, the message queue should have a CCSID of 65535 or 65534. You can change the message queue CCSID to 65535. Or, you can follow these steps:

1. Delete all of the messages.
2. Change the CCSID of the message queue to 65534.
3. Send all of the messages again.

Can I correct the CCSID of a message file?:

You might have a message file that has a CCSID that does not match the CCSID of the message descriptions in it. This typically results from adding message descriptions with a message-level CCSID of 65535 to a message file with a CCSID that is not 65534 or 65535.

If all of the message descriptions in a message file have the same message-level CCSID, and you know the message-level CCSID, you can enter the following command:

```
CHGMSGF MSGF(MYMSGF) CCSID(nnnnn)
```

In this example, MYMSGF is the name of the message file and nnnnn is the message-level CCSID.

If you do not know the CCSID of all the message descriptions in the file or if the message descriptions in the file have different CCSIDs, the message file should have a CCSID of 65535 or 65534. You can handle this situation in either of the following ways:

- Change the CCSID of the message file to 65535.
- Follow these steps:
  1. Change the CCSID of the message file to 65534.
  2. Change the message-level CCSID of each message description to the correct value.

Related concepts

"Can I correct the CCSID of a message description?"
You can use the Change Message Description (CHGMSGD) command to change the CCSID of a message description. If you do not change the first- or second-level text at the same time that you change the message description CCSID, the text remains unchanged. Only the CCSID changes.

Can I correct the CCSID of a message?:

You cannot correct the message-level CCSID of a message. You can change the message queue CCSID to match the message-level CCSID. You can also delete the message and send it again with the correct message-level CCSID.

Related concepts

"Message-level support" on page 138
When a message is sent to a message queue, you must communicate the CCSID of the replacement data or the immediate message text to the i5/OS operating system. Use the CCSID parameter on any of the send message commands or APIs to communicate this CCSID to the system.

"Message description-level support" on page 140
When a message description is added to a message file, the CCSID of the message text must be communicated to the i5/OS operating system.

Can I correct the CCSID of a message description?:

You can use the Change Message Description (CHGMSGD) command to change the CCSID of a message description. If you do not change the first- or second-level text at the same time that you change the message description CCSID, the text remains unchanged. Only the CCSID changes.

For example, you can enter the following Change Message Description command to correct the CCSID of a message description without changing any of the first- or second-level message text:

```
CHGMSGD MSGF(MYLIB/MYMSGQ) MSGID(TST0001) CCSID(37)
```

Related tasks

"Can I correct the CCSID of a message file?"
You might have a message file that has a CCSID that does not match the CCSID of the message descriptions in it. This typically results from adding message descriptions with a message-level CCSID of 65535 to a message file with a CCSID that is not 65534 or 65535.

Related reference

Change Message Description (CHGMSGD) command

Changing the CCSID of a physical file

You can use the Change Physical File (CHGPF) command to change the coded character set identifier (CCSID) of a physical file. However, the physical file cannot be changed under certain conditions.

A physical file cannot be changed if one or more of the following conditions exist when working with a logical file defined over a physical file:

- The logical file has a sort sequence table associated with the CCSID of the physical file and the CCSID you want to change to is incompatible. That is, conversion between the original CCSID and the CCSID you want to change to is not allowed because all the characters of the original CCSID are not in the new CCSID.
- The logical file has a sort sequence table associated with the CCSID of the physical file and the CCSID you want to change to is incompatible. Additionally, the logical file has fields defined with CCSIDs that are not compatible to the new CCSID you want to change the physical file to. Again, conversion between the original CCSID and the CCSID you want to change to is not allowed because all the characters of the original CCSID of the logical file or the fields with specific CCSIDs are not in the new CCSID.
- 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 such that the CCSID of the logical file's secondary access path is different than the CCSID to which the physical file is being changed.

Related concepts

"Display files" on page 132
When a display file object is created, it is tagged with the coded character set identifier (CCSID) of the source file.

Related reference

Change Physical File (CHGPF) command

Character sorting

Traditionally, information is displayed in sorted order to enable users to easily find the items they are looking for. However, users of different languages might have very different expectations of what a sorted list should look like.

Not only does the alphabetical order vary from one language to another, but it also can vary from document to document within the same language. For example, phonebook ordering might be different than dictionary ordering. String comparison is one of the basic functions most applications require, and yet implementations often do not match local conventions.

For example, here are some of the ways languages vary in the ordering of strings:

- The letters A through Z can be sorted in a different order than in English. For example, in Lithuanian, the letter y is sorted between the letters i and k.
- Accented letters can be treated as distinct letters. For example, Å in Danish is treated as a separate letter that sorts just after the letter Z.
- Unaccented letters that are considered distinct in one language can be indistinct in another. For example, the letters v and w are two different letters according to English. However, v and w are considered variant forms of the same letter in Swedish.

i5/OS lets you customize the sequence in which characters are sorted. You can use any of the following methods to do this:

- i5/OS sort sequence support

- ICU-based sort support
- Locale sort support. If your application uses locales, you can use the sorting support provided by the LC_COLLATE locale category.

  Related concepts

  "Sort sequences" on page 86
  The i5/OS operating system supports sort sequence. By using one of the listed options, you can order your data according to cultural-dependent requirements for specific applications.

  "LC_COLLATE category" on page 203
  The LC_COLLATE category defines character or string collation information. Within LC_COLLATE you can specify a sort sequence to use using the cpysyscol keyword. The *cpysyscol* keyword value is used in place of the LC_COLLATE category definitions.

i5/OS sort sequence support:

i5/OS sort sequence support is the sort support that the system has provided for multiple releases. Using this support, users can create their own defined sort tables. However, the sort is limited to a single weight for each individual character.

i5/OS sort sequence also provides full support only for single-byte character set (SBCS) and CCSID 13488. These sort sequences are supported over most of the system.

Sort sequence tables:

A sort sequence table is an object that contains the weight of each single-byte graphic character within a specified coded character set identifier (CCSID). The system-recognized identifier for the sort sequence table object type is *TBL.

Depending on your requirements, you can define a table to have either a unique weight for each graphic character or shared weights for some graphic characters. If you define a table that contains unique weights for each character within the character set, your table is known as a unique-weight table. If you define a table that contains some graphic characters that share the same weight, your table is known as a shared-weight table. For example, if you want to sort the graphic character capital letter A and the graphic character small letter a together, you can define a shared-weight table. If you want to sort these graphic characters separately, you can define a unique-weight table.

A set of sort sequence tables is included with the system. This set of tables defines both unique-weight and shared-weight sort sequences for all single-byte character set (SBCS) languages.

Sort sequence table implementation notes

Sort sequence support does not take into consideration the following information:

- Special cases of single characters that should be handled as multiple characters (such as the German characters sharp).
- Sequences of characters that should be treated as a single character (such as the Danish aa, Hungarian ly, Serbian lj, Spanish ll).
- Nonalphanumeric characters that should be ignored because they are embedded in alphanumeric strings (such as the hyphen in co-op).
- Prefixes that should be ignored (such as *Van der* in the name *Van der Pool*).
- Program-described files.
- DBCS code pages.

If a sort sequence table has a weight other than hexadecimal 40 assigned to the blank character, unpredictable results can occur when strings of unequal lengths are compared.

Sort sequence tables included with the system

You can use the Work with Tables (WRKTBL) command to view the contents of the sort sequence tables that are included with the i5/OS operating system. The tables are located in the QSYS library.

When looking at these tables, consider the following information:
- Several tables included with the system represent a single sort sequence, each encoded with a different coded character set identifier (CCSID) value. Not all of the characters in a given sort sequence exist in every CCSID in which the sort sequence is encoded.
- Use the language identifier (LANGID) parameter and the sort sequence (SRTSEQ) parameter to access the unique-weight tables (*LANGIDUNQ) or the shared-weight tables (*LANGIDSHR).
- When using the sort sequence, the relative weights shown in these tables differ from the actual weights in the sort sequence table on the system. The relative weights shown in these tables are examples only.
- The relative unique weight of a character is shown by the order of the characters in the sort sequence table. The relative unique weight is determined by assigning a weight of 1 to the first character in the sort sequence table and incrementing by 1 for each of the following characters until the end of the table is reached.
- GCGID is the graphic character global identifier.

For example, the Arabic sort sequence table shows the relative sort sequence weights for characters that are sorted using the Arabic sort sequence table.

How to build sort sequence tables

To create a user-defined sort sequence table, copy an existing sort sequence table using the Create Table (CRTTBL) command, and then modify the copy of the table. Table functions allow you to perform the following tasks:
- Use a definition stored in a source member.
- Create a table based on another sort sequence table using an interactive interface.

You can create a sort sequence table (MYTEST) from a copy of an existing table using the following CRTTBL command:

```
CRTTBL TBL(MYTEST) SRCFILE(*PROMPT) TBLTYPE(*SRTSEQ)
BASESRTSEQ(QSYS/QLA10025S) CCSID(037)
```

This command displays a sort sequence table that you can modify. Your table is created from a function key on this display. Your resulting table has a coded character set identifier (CCSID) value of 00037. The table is named MYTEST and is stored in the current library.

The following table shows one way in which the resulting characters may be shown on the first display of the MYTEST sort sequence table. The actual panel shows characters instead of text descriptions. For example, the character shown for sequence 0100 is a question mark (?), and the character shown for sequence 0070 is a colon (:).

Note: The characters that you actually see on the first display of the MYTEST sort sequence table might vary, depending on the device that you use.

Sequence	Character
0010	Equal sign
0020	Overline
0030	(SHY)
0040	Hyphen

Sequence	Character
0050	Comma
0060	Semi-colon
0070	Colon
0080	Exclamation mark
0090	Inverted exclamation mark
0100	Question mark
0110	Inverted question mark
0120	Slash
0130	Period
0140	Acute accent mark
0150	Grave accent mark
0160	Caret
0170	Right square bracket
0180	Tilde
0190	Small multiply dot
0200	Comma

You can make changes to the tables to move characters in each code page to the preferred position for the national language sort sequence table. The ordering is done by increments of 10. Therefore, the first value is 10, then 20, and so on. If some characters have a shared weight, these groups of characters have the same sequenced weight.

Related concepts

"Sort sequence (QSRTSEQ) system value" on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

"ILE RPG sort sequence" on page 100
The ILE RPG feature, an option of the IBM WebSphere Development Studio for System i licensed program, provides the possibility for a user to specify a sort sequence table and to use the table in comparison operations that are performed with nonnumeric data.

"Working with CCSIDs" on page 126
Using the system implementation of Character Data Representation Architecture (CDRA), you can achieve consistent representation, processing, and interchange of coded characters (data) on the i5/OS operating system and across IBM Systems. The primary implementation of CDRA on the i5/OS operating system is through coded character set identifier (CCSID) support.

"Code pages" on page 306
Several IBM code pages match the International Standard ISO/IEC 8859. The i5/OS operating system supports parts of the ISO Standard with equivalent IBM code pages.

Related reference

Create Table (CRTTBL) command

Sort sequence for Arabic:

The Arabic sort sequence table is used with the Arabic language.

GCGID	Character	Shared weight	Unique weight
SP010000	(SP)	1	1
SP090000	_	2	2
SP320000	(SHY)	3	3
SP100000	-	4	4
SP080000	,	5	5
SP080007	،	6	6
SP140000	;	7	7
SP140007	؛	8	8
SP130000	:	9	9
SP020000	!	10	10
SP150000	?	11	11
SP150007	؟	12	12
SP120000	/	13	13
SP110000	.	14	14
SD130000	`	15	15
SD190000	~	16	16
SP050000	'	17	17
SP040000	"	18	18
SP060000	(	19	19
SP070000	)	20	20
SM060000	[	21	21

GCGID	Character	Shared weight	Unique weight
SM080000	]	22	22
SM110000	{	23	23
SM140000	}	24	24
SM050000	@	25	25
SC010000	¤	26	26
SC040000	¢	27	27
SC030000	$	28	28
SM040007	*	29	29
SM070000	\	30	30
SM030000	&	31	31
SM010000	#	32	32
SM020007	%	33	33
SA010000	+	34	34
SA060000	÷	35	35
SA070000	×	36	36
SA030000	<	37	37
SA040000	=	38	38
SA050000	>	39	39
SM660000	¬	40	40
SM130000	|	41	41
SM650000	¦	42	42
SP300000	(RSP)	43	43
SP310000	(NSP)	43	44
SM870000	[illegible]	43	45

GCGID	Character	Shared weight	Unique weight
ND100000	0	44	46
ND100001	٠	44	47
ND010000	1	45	48
ND010001	١	45	49
ND020000	2	46	50
ND020001	٢	46	51
ND030000	3	47	52
ND030001	٣	47	53
ND040000	4	48	54
ND040001	٤	48	55
ND050000	5	49	56
ND050001	٥	49	57
ND060000	6	50	58
ND060001	٦	50	59
ND070000	7	51	60
ND070001	٧	51	61
ND080000	8	52	62
ND080001	٨	52	63
ND090000	9	53	64
ND090001	٩	53	65
LA010000	a	54	66
LA020000	A	54	67
LB010000	b	55	68
LB020000	B	55	69

GCGID	Character	Shared weight	Unique weight
LC010000	c	56	70
LC020000	C	56	71
LD010000	d	57	72
LD020000	D	57	73
LE010000	e	58	74
LE020000	E	58	75
LF010000	f	59	76
LF020000	F	59	77
LG010000	g	60	78
LG020000	G	60	79
LH010000	h	61	80
LH020000	H	61	81
LI010000	i	62	82
LI020000	I	62	83
LJ010000	j	63	84
LJ020000	J	63	85
LK010000	k	64	86
LK020000	K	64	87
LL010000	l	65	88
LL020000	L	65	89
LM010000	m	66	90
LM020000	M	66	91
LN010000	n	67	92
LN020000	N	67	93

GCGID	Character	Shared weight	Unique weight
LO010000	o	68	94
LO020000	O	68	95
LP010000	p	69	96
LP020000	P	69	97
LQ010000	q	70	98
LQ020000	Q	70	99
LR010000	r	71	100
LR020000	R	71	101
LS010000	s	72	102
LS020000	S	72	103
LT010000	t	73	104
LT020000	T	73	105
LU010000	u	74	106
LU020000	U	74	107
LV010000	v	75	108
LV020000	V	75	109
LW010000	w	76	110
LW020000	W	76	111
LX010000	x	77	112
LX020000	X	77	113
LY010000	y	78	114
LY020000	Y	78	115
LZ010000	z	79	116
LZ020000	Z	79	117

GCGID	Character	Shared weight	Unique weight
AX300000		80	118
AA210000		80	119
AA210001		80	120
AA210002		80	121
AA210006		80	122
AA310000		80	123
AA310001		80	124
AA310002		80	125
AA310006		80	126
AW310000		80	127
AA310401		80	128
AA310406		80	129
AY310001		80	130
AY310000		80	131
AA010000		81	132
AA010001		81	133
AA010002		81	134
AA010006		81	135
AB010000		82	136
AB010003		82	137
AT020000		83	138
AT010000		83	139
AT010003		83	140
AT470000		84	141

GCGID	Character	Shared weight	Unique weight
AT470003	ثـ	84	142
AG230000	ج	85	143
AG230003	جـ	85	144
AH450000	ح	86	145
AH450003	حـ	86	146
AH470000	خ	87	147
AH470003	خـ	87	148
AD010000	د	88	149
AD470000	ذ	89	150
AR010000	ر	90	151
AZ010000	ز	91	152
AS010006	س	92	153
AS010000	س	92	154
AS010003	سـ	92	155
AS230006	ش	93	156
AS230000	ش	93	157
AS230003	شـ	93	158
AS450006	ص	94	159
AS450000	ص	94	160
AS450003	صـ	94	161
AD450006	ض	95	162
AD450000	ض	95	163
AD450003	ضـ	95	164
AT450000	ط	96	165

GCGID	Character	Shared weight	Unique weight
AZ450000	ظ	97	166
AC470000	ع	98	167
AC470002	ـع	98	168
AC470003	عـ	98	169
AC470004	ـعـ	98	170
AG310000	غ	99	171
AG310002	ـغ	99	172
AG310003	غـ	99	173
AG310004	ـغـ	99	174
AF010000	ف	100	175
AF010003	فـ	100	176
AQ010000	ق	101	177
AQ010003	قـ	101	178
AK010000	ك	102	179
AK010003	كـ	102	180
AL010000	ل	103	181
AL010003	لـ	103	182
AL220000	لآ	104	183
AL220003	ـلآ	104	184
AL320000	لأ	104	185
AL320003	ـلأ	104	186
AL020000	لا	104	187
AL020003	ـلا	104	188
AM010000	م	105	189

GCGID	Character	Shared weight	Unique weight
AM010003	ﻤ	105	190
AN010000	ن	106	191
AN010003	ﻧ	106	192
AH010000	ه	107	193
AH010003	ﻫ	107	194
AH010007	ﻬ	107	195
AH010004	ﻪ	107	196
AW010000	و	108	197
AA020000	ى	109	198
AA020001	ﻰ	109	199
AA020002	ﻰ	109	200
AY010000	ي	109	201
AY010001	ﻲ	109	202
AY010002	ﻳ	109	203
AY010003	ﻴ	109	204
AA070000	ً	110	205
AU070000	ٌ	111	206
AI070000	ٍ	112	207
AA050000	َ	113	208
AA050004	ـَ	113	209
AU050000	ُ	114	210
AU050004	ـُ	114	211
AI050000	ِ	115	212
AI050004	ـِ	115	213

GCGID	Character	Shared weight	Unique weight
AE050000	[illegible]	116	214
AE050004	[illegible]	116	215
AX100000	[illegible]	117	216
AX100004	[illegible]	117	217
SM860000	[illegible]	118	218

Sort sequence support:

The sort sequence support is provided in these i5/OS functions.

- A user interface for creating new tables based on system-supplied sort sequence tables
- The Work with Tables (WRKTBL) command for creating and displaying tables
- The Create Table (CRTTBL) command for creating tables
- CL, ILE RPG IV, and ILE COBOL for compilers.
- Program support
- Work management support
- Database management support
- Other system components support

Related reference

Work with Tables (WRKTBL) command

Create Table (CRTTBL) command

Sort sequence support in programs:

You can assign a sort sequence to a program that is used for ordering and comparing data by specifying the sort sequence to be used at compilation time.

Specify the sort sequence to be used with the sort sequence (SRTSEQ) parameter and language identifier (LANGID) parameters of the create program commands. Valid SRTSEQ parameter values are:

- SRTSEQ(*HEX) means that no sort sequence should be used (hexadecimal sorting).
- SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) means that the unique- or shared-weight sort sequence, determined by the LANGID parameter, should be used.
- A name for the system-supplied or user-supplied sort sequence name can be specified explicitly on the SRTSEQ parameter. If you explicitly specify a sort sequence name, the LANGID parameter is ignored.
- SRTSEQ(*JOB) or LANGID(*JOB) means that the sort sequence to be used is determined by the value associated with the job when the program is created.
- SRTSEQ(*JOBRUN) or LANGID(*JOBRUN) means that the sort sequence to be used is determined by the values from the job when the program is run.

The first three options assign the sort sequence to the program object at creation time. This sequence is always used when the program is run. Using the *JOBRUN value on the SRTSEQ or LANGID parameters, however, provides the possibility for dynamically assigning sort sequence to the program.

Sort sequence support in work management:

Work management involves the assigning of the SRTSEQ value at the job level, the user profile level, and the system value level.

Sort sequence support at the job level: A sort sequence (SRTSEQ) value is assigned to a job. It is valid on the Submit Job (SBMJOB), Batch Job (BCHJOB), and the Change Job (CHGJOB) commands. If a program is created with SRTSEQ(*JOB), the sort sequence is set from the job sort sequence. If a program is created with SRTSEQ(*JOBRUN), the sort sequence is set from the job sort sequence at run time.

Sort sequence support at the user profile level: The user profile assigns a SRTSEQ value to a user and, by default, to all jobs running under this user profile. The user profile SRTSEQ value defaults to the sort sequence system value (QSRTSEQ).

Sort sequence support at the system value level: The QSRTSEQ system value defines a sort sequence that can be referred to by other objects. The QSRTSEQ system value should be set according to the requirements of the primary language used on the system.

Related concepts

"Sort sequence (QSRTSEQ) system value" on page 24
The sort sequence (QSRTSEQ) system value, along with the QLANGID system value, determines the sort sequence table to be used for sorting character data.

Sort sequence support in database management:

Database management supports the SRTSEQ and LANGID parameters on the Create Physical File (CRTPF) and Create Logical File (CRTLF) commands.

The LANGID and SRTSEQ parameters determine a sort sequence table. The sort sequence table is captured at file creation time and is stored as an attribute of the file. The SRTSEQ job attribute has no effect on the processing of an existing database file. The sort sequence table associated with the file is used for key sequencing, select logic fields and omit logic fields, and for join field functions.

The ALTSEQ keyword in DDS can also be used to specify a sort sequence table. The ALTSEQ keyword applies only to the key fields, not to the select logic fields and the omit logic fields. If the SRTSEQ parameter is specified on the CRTPF command or the CRTLF commands and the ALTSEQ keyword in the DDS source file specify a sort sequence table, an error message is sent and the file is not created.

The default SRTSEQ parameter on CRTPF and CRTLF commands is *SRC, which indicates that the sort sequence table on the ALTSEQ keyword should be used. If ALTSEQ is not used in DDS, the SRTSEQ attribute of the job determines the file attributes when creating or changing the file.

How sort sequences are specified for database management

Sort sequence tables can be specified in the following areas:

- IBM Query for i5/OS support

  External sort sequence tables (including those included with the system) and user-defined tables can be specified.
- IBM DB2 Query Manager and SQL Development Kit for i5/OS

  The Create Structured Query Language xxx (CRTSQLxxx) commands and the Start Structured Query Language (STRSQL) command support the SRTSEQ and LANGID parameters.

  A sort sequence table can be specified when a query object is being defined with the Work with Queries display. The sort sequence (SRTSEQ) value and language identifier (LANGID) value are specified on the Specify Sort Sequence display.
- DB2 for i5/OS Query Management

  The Create Query Management Query (CRTQMQRY) command supports the SRTSEQ and LANGID parameters.

Related concepts

Database programming

Related reference

Create Physical File (CRTPF) command

Create Logical File (CRTLF) command

Start SQL Interactive Session (STRSQL) command

Create Query Management Query (CRTQMQRY) command

Related information

Arranging key fields using the SRTSEQ parameter

Sort sequence support in other system components:

Sort sequence support is found in these components of the system.

- CRTCLPGM (Create Control Language Program) command

  The LANGID and SRTSEQ parameters are supported.
- DSPPGM (Display Program) command

  The LANGID and SRTSEQ values that were specified when the program was created are displayed.
- CRTDSPF (Create Display File) command

  The LANGID and SRTSEQ parameters are supported. The values of the RANGE, VALUES, and COMP keywords are validated when the display file is compiled.
- High-level languages

  Using ILE COBOL and ILE RPG IV languages, you can specify SRTSEQ and LANGID values directly on the Create Bound Program (CRTBNDXXX) commands. Original Program Model RPG and COBOL compilers use the Create Program (CRTXXXPGM) commands. With ILE C, you can also specify SRTSEQ and LANGID values when you create a locale. You can then associate the locale with a program.
- System i Access

  The transfer function allows a sort sequence table to be specified when you perform queries on database files and SQL tables.

Related reference

Display Program (DSPPGM) command

Create Display File (CRTDSPF) command

Related information

Create CL Program (CRTCLPGM) command

System i Access

Sort sequence scenarios:

This table shows characters you can sort using a binary, a shared-weight, and a unique-weight sort sequence for the Danish code page 00277.

Character name	Character illustration	Code point in code page 277	Shared sort weight	Unique sort weight
AE ligature	Æ	X’7B’	96	183
O slash	Ø	X’7C’	97	187
A overcircle	Å	X’5B’	98	191

Character name	Character illustration	Code point in code page 277	Shared sort weight	Unique sort weight
Latin capital N	N	X’D5’	83	132
Latin capital Z	Z	X’E9’	95	181
O umlaut	Ö	X’EC’	97	189
Latin capital A	A	X’C1’	70	77

Using the information in the previous table, the characters are sorted in ascending order as shown in the following table.

Position in ascending order	Binary sort	Shared weight sort	Unique weight sort
First	A overcircle	Latin capital A	Latin capital A
Second	AE ligature	Latin capital N	Latin capital N
Third	O slash	Latin capital Z	Latin capital Z
Fourth	Latin capital A	AE ligature	AE ligature
Fifth	Latin capital N	O umlaut	O slash
Sixth	Latin capital Z	O slash	O umlaut
Seventh	O umlaut	A overcircle	A overcircle

The following table shows an example of a shared-weight sort sequence, a unique weight sort sequence, and the binary sort sequence for English code page 00037.

Binary sort sequence	Shared-weight sort sequence using LANGID(ENU) and SRTSEQ(*LANGIDSHR)	Unique-weight sort sequence using LANGID(ENU) and SRTSEQ(*LANGIDUNQ)
Jones, Mary	JOHNSON, JOHN	JOHNSON, JOHN
JOHNSON, JOHN	JONES, MARTIN	Jones, Mary
JONES, MARTIN	Jones, Mary	JONES, MARTIN
Smith, Ron	SMITH, ROBERT	Smith, Ron
SMITH, ROBERT	Smith, Ron	SMITH, ROBERT

Sort sequence types:

The i5/OS operating system provides a set of shared-weight and unique-weight sort sequence tables for SBCS languages.

A shared-weight sequence is a sort sequence in which some graphic characters may have the same weight as some other characters in the sequence. Those with the same weight sort together as though they were the same character. For example, the letters *a* and *A* might both have the same value 24. This ensures that words such as *able* and *Able* are kept together in a list. In a simple sort table, *a* and *A* might share the value 24, and *b* and *B* might share the value 25 and so on.

A unique-weight sequence is a sort sequence in which each graphic character has a weight different from the weight of every other graphic character in the sequence.

ICU-based sort support:

International Components for Unicode (ICU) based sort support is based on the ICU collation services, which provide a multiple-weight (tertiary level) sort support.

ICU-based sort supports the sorting of data in most CCSIDs supported by the operating system. It also provides a multiple-weight sort based on strings. ICU-based sort support does not allow users to generate additional sort tables.

ICU-based sort sequence types:

An ICU locale sequence is a sort sequence in which a tertiary level sort is used in determining a sort key based on the text string.

In tertiary level sorting, upper and lower case differences in characters are distinguished (for example, ″ao″ < ″Ao″ < ″aỏ″). In addition, a variant of a letter differs from the base form on the tertiary level (such as ″a″ and ″ả″).

ICU locales and sort tables provided by the system:

The ICU locale data covers 82 different languages, further divided into 197 regions and variants. For each language, data such as days of the week, months, and their abbreviations are defined. The sort sequence tables for ICU that are defined on the system are based on these locales and regions, as listed in the table in this topic.

The system provides a table (*TBL) object for each of these locales for you to use when you specify an ICU-based sort. For example, the table object QSYS/ES_MX directs the database to use the 2.6 version of the ICU sort for Spanish in Mexico. The table object QSYS/I34ES_MX directs the database to use the 3.4 version of the ICU sort for Spanish in Mexico. ICU provides over 100 locale choices that you can use. See the following example table for a list of the Spanish choices. For the full list of locales that you can use for sort choices, see the IBM Globalization - ICU Web page.

For systems running i5/OS V6R1, or later, the system ICU sort support has been updated to ICU 3.4. For systems running i5/OS V5R4, or earlier, this support is based on ICU 2.6. The ICU 3.4 tables follow the naming of I34xx_yy. You can see these choices by looking at the table objects on the operating system. Use the following command to see the ICU 3.4 tables: `WRKTBL TBL(QSYS/I34*)`

If your applications use ICU sort support, you need to update them to use the 3.4 version of the support for better performance.

Example locale name	Language or variant	Region
ES	Spanish	
ES_AR	Spanish	Argentina
ES_BO	Spanish	Bolivia
ES_CL	Spanish	Chile
ES_CO	Spanish	Colombia
ES_CR	Spanish	Costa Rica
ES_DO	Spanish	Dominican Republic
ES_EC	Spanish	Ecuador
ES_ES	Spanish	Spain
ES_ES_PREE	Spanish Pre-Euro	Spain

Example locale name	Language or variant	Region
ES_GT	Spanish	Guatemala
ES_HN	Spanish	Honduras
ES_MX	Spanish	Mexico
ES_NI	Spanish	Nicaragua
ES_PA	Spanish	Panama
ES_PE	Spanish	Peru
ES_PR	Spanish	Puerto Rico
ES_PY	Spanish	Paraguay
ES_SV	Spanish	El Salvador
ES_US	Spanish	United States
ES_UY	Spanish	Uruguay
ES_VE	Spanish	Venezuela
ES__TRADIT	Spanish Traditional	

Sort sequence scenarios:

This table shows an example of the results you might obtain, given the same input data but different ICU locales specified on the sort request. The two locales used are EN_US and FR_FR.

en_us ICU locale sort sequence using SRTSEQ(EN_US)	fr_fr ICU locale sort sequence using SRTSEQ(FR_FR)
cote	cote
cot é	c ô te
c ô te	cot é
c ô t é	c ô t é

Related concepts
ICU sort sequence

ICU-based sort sequence support:

i5/OS support for ICU sort sequence is provided in i5/OS functions in the Work management and Database management topics.

Related concepts
Work management

Related information
Database

Working with bidirectional data

Arabic and Hebrew languages use an alphabet written and read from right to left. Numerics and Latin text imbedded in the right-to-left text are written and read from left to right. Therefore, these languages are called bidirectional languages.

Because bidirectional languages are written and read from right to left, you should avoid using the terms left and right. For example, *right margin* in Hebrew or Arabic documents is the beginning of the line and not the end. Use the words *start* and *end* in place of the words *right* and *left*.

Hebrew and Arabic have no case-sensitive characters. To avoid the incorrect presentation of characters, no case-sensitive checking or substitution should be performed. In addition, the Arabic language does not use abbreviations, therefore, you should use only complete words.

Related concepts

"Coding globalized applications that use bidirectional data" on page 106
When you are developing NLV-enabled applications, you should consider some specific restrictions on bidirectional languages.

Bidirectional application support

Workstations, display files, and the user interface manager (UIM) provide support for bidirectional applications.

Workstation support

Workstations that have the ability to display Arabic and Hebrew character sets also have the ability of right-to-left cursor movement. Right-to-left cursor movement on input fields can be achieved in one of the following ways:

- Pressing a special function key available on Hebrew and Arabic keyboards called the reverse key. This is a toggle function that moves the cursor to the other side of the field, allows for cursor movement in the opposite direction, and also changes the language layer from Latin to Hebrew or Arabic and back again.
- Using the DDS cursor control codes for display files. When the CHECK keyword is used with a cursor-controlled code, it specifies that the cursor is to move from right to left. The following parameters are valid cursor control codes:
  - CHECK (RL): Moves the cursor from right to left in specified nonnumeric input fields or in all nonnumeric input fields on the display.
  - CHECK (RLTB): Moves the cursor from right to left between fields.

  When using these parameters, consider the following information:
  - Modulus check digit verification is supported, but the check digit is the byte to the extreme right of the field.
  - A field for which right-to-left cursor movement is specified can occupy more than one line on the display. However, the cursor still moves from the top of the display to the bottom.
  - You cannot use right-to-left cursor movement with user-defined data streams.

Note: If no cursor positioning is specified in the display file or by the program, the cursor is placed in the input-capable field to the extreme left of the top line.

Display file support

The system does not check to make sure that all display files that open to the display station are capable of right-to-left cursor movement. Therefore, it is the responsibility of application programmers to ensure that the appropriate display files are used.

User interface manager support

The user interface manager gives the following bidirectional support for creating online information and panels:

- BIDI= NONE | RTL | LTR

  This attribute controls the directional orientation of the panels in the panel group.

  RTL indicates that the panel in the panel group is bidirectional and should be displayed with a right-to-left orientation.

  LTR indicates that the panel in the panel group is bidirectional and should be displayed with a left-to-right orientation.
- :RT and :ERT

  Reverse-direction-text tags indicate that the enclosed text has an orientation that is opposite to the orientation of the panel group.

  For a list of UIM tags, see the Application Display Programming PDF.

  Related reference

  DDS concepts

  Related information

  Application Display Programming PDF

Checklist: Bidirectional support guidelines

When you create an application with bidirectional support, follow the guidelines in this table.

Complies	Not applicable	Rule
		Software design must allow for bidirectional data to be passed to applications in the same order that a speaker of the language can spell it out.
		The product design must allow for the implementation of the correct handling of bidirectional keyboard and presentation functions.
		Designing of a function that implies logical movement of cursor or characters must permit mirroring of that function.
		Keys or operations labeled with directional icons or symbols must perform according to the icon or symbol.
		Keyboard nomenclature for mirrored functions must be independent of the direction of data or text entry.
		Display functions must not assume a left-to-right orientation.
		Field attributes must contain room for directional information.
		Indicator location must be reserved for the current direction of the cursor (direction of input).
		The design must allow for independent handling of graphic and text orientation.
		Provision must be made to allow shape determination to be performed.
		The deshaping must be definable.
		Provision must be made to allow the selection of the appropriate presentation shape for the numerals.
		Characters must be allowed to touch each other on printers and displays.
		Indicator locations should be reserved for screen and field orientation, current level of nesting, status of push (nesting mechanism), and status of symmetric swapping.
		The design should provide for a method to indicate to the user the nesting structure of a string.

Complies	Not applicable	Rule
		A system-wide method of deshaping Arabic characters or character strings should be provided.
		An indicator location should be provided for the status of shape determination.
		A method should be provided so that proportional spacing can be provided.
		A method should be provided to allow alignment of the baseline of Arabic and Latin characters (including Hindi and Arabic shapes for numerals).

Related concepts

"Coding globalized applications that use bidirectional data" on page 106
When you are developing NLV-enabled applications, you should consider some specific restrictions on bidirectional languages.

Working with DBCS data

A DBCS file is a file that contains double-byte data or a file that is used to process double-byte data, for example, Japanese, Chinese, or Korean. Other files are called alphanumeric files. You can view DBCS files on display, printer, tape, diskette, and ICF devices.

A more modern method to support DBCS data is to use Unicode instead of DBCS fields. (IBM suggests that you use Unicode to develop new applications.)

You use data description specifications (DDS) to describe DBCS-capable device files.

You should indicate that a file is DBCS in one or more of the following situations:

- The file receives input, or displays or prints output, which has double-byte characters.
- The file contains double-byte literals.
- The file has double-byte literals in the DDS that are used in the file at processing time (such as constant fields and error messages).
- The DDS of the file includes DBCS keywords.
- The file stores double-byte data (database files).

DBCS strings in a mixed data stream

Typically, both single-byte characters and double-byte characters are used in a DBCS environment. For example, an accounting firm in Japan uses both English and Japanese for the spreadsheet. If both English and Japanese are being encoded as mixed SBCS and DBCS, the product must be able to understand a mixed character set that contains both single-byte coded characters and double-byte coded characters.

In IBM systems that use EBCDIC, a DBCS string is bracketed in a mixed data stream by a shift-out (SO) control character and a shift-in (SI) control character.

The following example shows the coding for a mixed string:

```
sss (SO) D1D2D (SI) ssss
```

The following example shows the coding for a mixed hexadecimal string:

```
818283 0E    41424143 0F 818283
```

Supported code ranges

The i5/OS operating system supports Japanese, Korean, Simplified Chinese, and Traditional Chinese character-set code ranges.

Using the System i Access Family of products, the systems also provide support for these non-IBM personal computer DBCS code pages:

- Republic of Korea National Standard graphic character set (KS)
- Taiwan Industry Standard graphic character set (Big5)
- The People's Republic of China National Standard graphic character set (GB)

Related concepts

DDS concepts

Related information

System i Access

Checklist: DBCS application design

When you create an application with double-byte coded character set (DBCS) support, follow these guidelines.

A complete list of these guidelines, as well as a full description of each guideline, is included in *Volume 1 Designing Enabled Products, Rules and Guidelines* (SE09-8001). For your convenience, a subset of these guidelines is provided in the following table.

Complies	Not applicable	Rule
		Double-byte coded character set code points in the graphic character range must be used only for graphic characters and must not be used for control purposes.
		Single-byte meaning must not be drawn from either byte of double-byte coded data.
		Double-byte coded character set character generators must be capable of producing user-accessible graphic characters.
		The ability to switch between single-byte coded character set and double-byte coded character set and the coexistence of single-byte coded character set and double-byte coded character set in the same session must be possible.
		User-interface text modules for double-byte coded character set systems must be loaded separately from the running code.

Developing applications that process DBCS data

You should design your application programs for processing double-byte data in the same way you design application programs for processing alphanumeric data.

Here are some additional considerations:

- Make sure that the double-byte data is always processed in a double-byte unit and does not split a double-byte character.
- Identify double-byte data used in the database files.
- Design display and printer formats that can be used with double-byte data.
- If needed, provide DBCS conversion as a means of entering double-byte data for interactive applications. Use the DDS keyword for DBCS conversion (IGCCNV) to specify DBCS conversion in display files. Because DBCS workstations provide a variety of double-byte data entry methods, you are not required to use the i5/OS DBCS conversion function to enter double-byte data.
- Create double-byte messages to be used by the program.
- Specify extended character processing so that the system prints and displays all double-byte data.
- Determine whether additional double-byte characters need to be defined. User-defined characters can be defined and maintained using the character generator utility (CGU). Information about CGU can be found in the *ADTS/400: Character Generator Utility*, book SC09-1769-00.

Related concepts

"Designing globalized applications" on page 56
Your goal in designing international application components is to create components that support national languages independently.

Use of double-byte data:

You can use double-byte data in several ways.

- As data in files:
  - Data in database files.
  - Data entered in input-capable and data displayed in output-capable fields of display files.
  - Data printed in output-capable fields in printer files.
  - Data used as literals in display files and printer files.
- As the text of messages.
- As the text of object descriptions.
- As literals and constants, and as data to be processed by high-level language programs.

Double-byte data can be displayed only at DBCS displays and printed only on DBCS printers. Double-byte data can be written onto diskette, tape, disk, and optical storage.

Where you cannot use double-byte data:

You cannot use double-byte data in the following ways:

- As i5/OS object names.
- As command names or variable names in control language (CL) and other high-level languages.
- As displayed or printed output on alphanumeric workstations.

Double-byte character size:

When displayed or printed, double-byte characters typically are twice as wide as single-byte characters.

Consider the width of double-byte characters when you calculate the length of a double-byte data field because field lengths are typically identified as the number of single-byte character positions used. The DDS concepts information includes more information about calculating the length of fields containing double-byte data.

Related reference

DDS concepts

Related information

Application Display Programming PDF

DBCS coding considerations:

If the application will be used in a DBCS environment, ensure that it is DBCS-enabled. Here are some suggestions to consider when you develop the general product design.

- Reserve more expansion space for DBCS textual data translation than you reserve for SBCS textual data translation. (It is possible, however, that the number of bytes used may be reduced when a SBCS sentence is being translated into DBCS.)
- Ensure programs can understand shift-out and shift-in delimiters. Otherwise, EBCDIC mixed-byte character strings cannot be handled.
- Do not enable short responses for DBCS. For short responses, it is difficult to shift in and out of DBCS. The `yes` and `no` are examples of short responses.
- Remember to use the graphic data type G where appropriate.

- Remember that the 5494 remote controller supports the graphic data type.
- Be careful when converting mixed data between DBCS-host code and DBCS-PC code, because the transition may change the data length. Losing and gaining SO and SI character pairs can upset field-length calculations.
- Make sure the double-byte data is always processed in a double-byte unit. Do not split a double-byte character.
- Design the display as well as the print format to avoid the problem of truncation of a double-byte character into two single-byte units.

Creating physical files:

When you create a physical file, display file, and printer file for a DBCS environment, consider the IGCDTA parameter in the these commands.

- Create Physical File (CRTPF) command

  If DBCS fields are described in DDS, the system treats the file as a DBCS file. Otherwise, specify *YES for the parameter of the CRTPF command so that the file can contain double-byte character set data. However, the system ignores the IGCDTA parameter value when a value for the RCDLEN parameter is not specified.
- Create Display File (CRTDSPF) and Create Printer File (CRTPRTF) commands

  Specify *YES for the parameter when using the CRTDSPF or CRTPRTF commands to create the externally described files. Then DBCS attributes, in addition to those defined in the DDS, can be specified.

Related reference

Create Physical File (CRTPF) command

Create Display File (CRTDSPF) command

Create Printer File (CRTPRTF) command

Target physical files:

When you specify OUTPUT(*OUTFILE) with the Copy Spooled File (CPYSPLF), Display Spooled File (DSPSPLF), or Work with Spooled Files (WRKSPLF) command under the DBCS version of the i5/OS operating system, the target physical file must be DBCS-enabled.

Note: The primary language of the system must support the double-byte character set to allow DBCS-enabled applications.

Use the QIGC system value to check if a DBCS version of the system is installed. Because it is set by the system, it cannot be changed. This system value can be referred to in an application program. QIGC can be:

- 0 (DBCS version is not installed)
- 1 (DBCS version is installed)

A DBCS system allows for concurrent use of SBCS and DBCS data. When the QIGC system value is 1, you should not assume all jobs are DBCS.

Related reference

Copy Spooled File (CPYSPLF) command

Display Spooled File (DSPSPLF) command

Work with Spooled Files (WRKSPLF) command

Using DBCS CCSIDs:

You should use DBCS CCSIDs for DBCS languages. When you design an application to be used in the DBCS environment, consider this information.

- If the QIGC system value is set on, system value QCCSID must have the value of a mixed CCSID.
- If the DBCS and SBCS language users are sharing the same system, they may want to store their data in different databases. It is possible to create DBCS-capable and SBCS-capable physical files in the same system. The CCSID parameter on the CRTPF command or the CCSID keyword on the physical file DDS definition can be used to specify the CCSID value that the data is stored in.
- If a CCSID was not explicitly assigned through DDS at file creation time, the database physical file character J (DBCS-only), E (DBCS-either), O (DBCS-open) or G (DBCS-graphic) fields are implicitly assigned a CCSID value.

Related reference

Create Physical File (CRTPF) command

Using DDS keywords:

With DDS keywords, you can specify alternative ways to enter data through display files, change input- and output-capable alphanumeric data fields to DBCS data fields, or specify the special features of the DBCS printer output.

- CHRSIZ (Character Size)

  This printer file keyword can expand the printer characters to twice the normal size (width and height). This keyword is valid only for IPDS printers and for printer files with a device type of *IPDS or *AFPDS specified.
- CONCAT (Concatenate)

  This keyword can be used only on logical files. This keyword does not support concatenation of a character field and a data type O field.
- DFLIN (Define Line)

  The printer file keyword draws horizontal and vertical lines.
- IGCALTTYP (DBCS Alternative Data Type)

  This display and printer keyword is used to change input- and output-capable character fields to DBCS fields with data type O.
- IGCANKCNV (Alphanumeric-to-DBCS Conversion)

  This printer file keyword converts alphanumeric SBCS characters to equivalent DBCS characters. Printed SBCS alphanumeric characters have the same appearance as printed DBCS characters. The printed DBCS characters, however, are twice as wide as the equivalent SBCS alphanumeric characters.
- IGCCDEFNT (DBCS Coded Font)

  This printer file keyword specifies the DBCS coded font for printing a named or constant field (or fields).
- IGCCNV (DBCS Conversion)

  This is a display file keyword that enables DBCS conversion.
- IGCCHRRTT (DBCS Character Rotation).

  This printer file keyword rotates each DBCS character 90 degrees counterclockwise before printing. By rotating characters, the system prints them in reading sequence. This keyword should be used only for printer files to be printed with 5553 printers or IPDS AFP(*YES) printers.

Related reference

DDS concepts

Related information

Application Display Programming PDF

DBCS file data types:

This table summarizes the changes of the data type of a field in a physical file when the data type is being referred to in a logical file.

Physical file data types	Logical file data types
J	J, O, E, H, G
O	O, H
E	E, O, H
A	A, O, E, H
H	J, O, E, A, H
G	G, O, J, E

The Katakana code page (00290):

The Katakana code page (code page 00290) for Japanese supports English and single-byte Katakana (phonetics) characters.

The lowercase English characters are located at code points different from other code pages, and some older hardware might not be able to display English uppercase, lowercase, and Katakana characters concurrently. Therefore, you need to take special considerations if the application is going to support this code page:

- Avoid using the lowercase alphabet for syntactic characters.
- Avoid using the SBCS lowercase alphabet with Japanese DBCS messages.

For more information about code page 290, see the IBM Coded Character Sets and Related Resources: Globalizing your e-business Web site .

Unicode support and IBM DBCS displays:

The i5/OS operating system supports Unicode. IBM DBCS-capable display stations, however, do not support Unicode data.

If you are designing an application to handle UCS-2 Level-1 data for display on an IBM DBCS-capable display, you must convert the data to a mixed-byte CCSID before sending the data to the display station.

Processing double-byte characters:

Because of the large number of double-byte characters, the i5/OS operating system needs more information to identify each double-byte character than is needed to identify each alphanumeric character.

There are two types of double-byte characters: basic and extended. These characters are typically processed by the device on which the characters are displayed or printed.

Basic double-byte characters

Basic characters are frequently used double-byte characters that reside in the hardware of a DBCS-capable device. The number of double-byte characters stored in the device varies with the language supported and the storage size of the device. A DBCS-capable device can display or print basic characters without using the extended character processing function of the operating system.

Double-byte extended characters

When processing extended characters, the device requires the assistance of the system. The system must tell the device what the character looks like before the device can display or print the character. Extended characters are stored in a DBCS font table, not in the DBCS-capable device. When displaying or printing extended characters, the device receives them from the DBCS font table under control of the operating system.

Extended character processing is a function of the operating system that is required to make characters stored in a DBCS font table available to a DBCS-capable device.

To request extended character processing, specify the double-byte extended character parameter, IGCEXNCHR(*YES), on the file creation command when you create a display (CRTDSPF) or create a printer file (CRTPRTF) command that processes double-byte data. Because IGCEXNCHR(*YES) is the default value, the system automatically processes extended characters unless you instruct it otherwise. You can change this file attribute by using the change file (CHGDSPF) or (CHGPRTF) command. You can override the file attribute with the override display file (OVRDSPF) or override printerfile (OVRPRTF) command. For example, to override the display file DBCSDSPF so that extended characters are processed, enter:

```
OVRDSPF DSPF(DBCSDSPF) IGCEXNCHR(*YES)
```

Notes:

1. The system ignores the IGCEXNCHR parameter when processing alphanumeric files.
2. When you use the Japanese 5583 Printer to print extended characters, you must use the Kanji print function of the Advanced DBCS Printer Support licensed program.

What happens when extended characters are not processed

When extended characters are not processed, the following events happen:

- Basic double-byte characters are displayed and printed.
- On displays, the system displays the undefined character where it would otherwise display extended characters.
- On printed output, the system prints the undefined character where it would otherwise print extended characters.
- The extended characters, though not displayed or printed, are stored correctly in the system.

Related reference

Create Display File (CRTDSPF) command

Create Printer File (CRTPRTF) command

Change Display File (CHGDSPF) command

Change Printer File (CHGPRTF) command

Override with Display File (OVRDSPF) command

Override with Printer File (OVRPRTF) command

Display support:

The i5/OS operating system inserts shift-control characters into DBCS-only fields automatically. The use of DBCS input fields affects the total number of input fields allowed on a display. Alphanumeric displays cannot display double-byte data correctly.

Inserting shift-control characters

The system inserts shift-control characters into DBCS-only fields automatically.

To insert shift-control characters into open fields or either fields, follow these steps:
1. Position the cursor in the field in which you want to insert double-byte data.
2. Press the Insert Shift Control Character key (according to your DBCS display user's guide).

The system inserts a pair of shift-control characters at the same time. The system leaves the cursor under the shift-in character and puts the keyboard in insert mode. Insert double-byte characters between the shift-control characters.

To find out if a field already has the shift-control characters, press the Display Shift Control Character key.

DBCS-graphic fields store double-byte characters without requiring the use of shift control characters. Shift control characters should not be inserted in graphic fields.

Number of displayed extended characters

The system can display up to 512 different extended characters on a Japanese display at one time. Additional extended characters are displayed as undefined characters. However, the additional extended characters are stored correctly in the system.

Number of input fields on a display

The use of DBCS input fields affects the total number of input fields allowed on a display. For a local 5250 display, you can specify as many as 256 input fields. However, each three instances of a DBCS field reduces the maximum number of fields by one. For example, if there are nine DBCS fields on a display, then the maximum is 256 - (9/3) = 253 input fields.

Effects of displaying double-byte data at alphanumeric workstations

Alphanumeric displays cannot display double-byte data correctly. If you try to display double-byte data at an alphanumeric display, the following happens:
- The system sends an inquiry message to that display, asking whether you want to continue using the program with double-byte data or to cancel it.
- If you continue using the program, the system ignores the shift-control characters and interprets the double-byte characters as though they were single-byte characters. The displayed double-byte data does not make sense.

Making printer files DBCS capable:

The system uses printer files to process data that will be printed or displayed. When the data involved contains double-byte characters, the printer file that is used to process the data must be capable of processing double-byte data.

In many cases, printer files are used by the system to produce data that will eventually be printed or displayed. In these cases, the data is first placed into a spooled file using one of the IBM-supplied printer files. The data is then taken from the spooled file and is displayed or printed based on the user's request.

When the data involved contains double-byte characters, the printer file that is used to place the data into the spooled file must be capable of processing double-byte data. A printer file is capable of processing double-byte data when *YES is specified on the IGCDTA parameter for the file. In most cases, the system recognizes the occurrence of double-byte data and takes appropriate measures to ensure the printer file that is used is capable of processing double-byte data.

In some cases, however, the system cannot recognize the occurrence of double-byte data and might attempt to use a printer file that is not capable of processing double-byte data. If this occurs, the output at the display or printer may not be readable. This can happen when object descriptions containing double-byte characters are to be displayed or printed on an alphanumeric device.

To ensure that you receive correct results when you display or print double-byte characters, some recommendations should be followed. Action is required on your part if you have a single-byte national language installed as a secondary language. Printer files that are received as part of the DBCS version of a product are always capable of processing DBCS data.

You should complete the following recommended actions after the product or feature has been installed:

1. If all printers and display devices attached to your system are DBCS-capable, you can enable all printer files for double-byte data. For IBM-supplied printer files that are received as part of a single-byte secondary language feature, you can enable all printer files by issuing the following command:

   ```
   CHGPRTF FILE(*ALL/*ALL) IGCDTA(*YES)
   ```

   After this command has been completed, all printer files in all libraries will be enabled for double-byte data. The change will be permanent.

2. If all printer and display devices that are attached to your system are not DBCS-capable, you should not enable all IBM-supplied printer files.

   Instead, use the library search capabilities of the system to control which printer files will be used for any particular job. When the potential exists that double-byte data will be encountered, the library list for the job should be such that the printer files that are DBCS-enabled will be found first in the library list. Conversely, if only single-byte data is expected to be encountered, the library list should be set up so the printer files that are not enabled for DBCS will be found first. In this way, the printer file capabilities will match the type of data that will be processed. The decision as to what type of printer file to use is made on the basis of what type of data will be processed. The device that will be used to actually display or print the data may also influence this decision.

In some cases it may be desirable to make the printer file only temporarily DBCS-capable instead of making a permanent change. For a specific job, you can make this temporary change by using the OVRPRTF command.

To temporarily enable a specific printer file, you can use the following command:

```
OVRPRTF FILE(filename) IGCDTA(*YES)
```

Where *filename* is the name of the printer file you want to enable.

Related reference

Override with Printer File (OVRPRTF) command

Copying spooled and nonspooled DBCS files:

When you copy spooled files to a database file, the database file must have been created with the IGCDTA(*YES) value specified. You can use the Copy File (CPYF) command to copy double-byte data from one file to another.

An extra column is reserved for the shift-out character, if the database file contains double-byte data.

Copying spooled files

Copy spooled files that have double-byte data by using the Copy Spooled File (CPYSPLF) command. However, the database file to which the file is being copied must have been created with the IGCDTA(*YES) value specified.

When copying spooled files to a database file that contains double-byte data, an extra column is reserved for the shift-out character. This shift-out character is placed between the control information for the record and the user data. The following table shows the shift-out character column number, based on the value specified for the Control character (CTLCHAR) keyword:

CTLCHAR value	Column for shift-out character
*NONE	1
*FCFC	2
*PRTCTL	5
*S36FMT	10

Copying nonspooled DBCS files

You can use the Copy File (CPYF) command to copy double-byte data from one file to another.

When you copy data from a double-byte database file to an alphanumeric database file, specify one of the following parameters on the CPYF command:

- If both files are source files or if both files are database files, you can specify either the FMTOPT(*MAP) parameter or the FMTOPT(*NOCHK) parameter.
- If one file is a source file and the other file is a database file, specify the FMT(*CVTSRC) parameter.

When you copy DBCS files to alphanumeric files, the system sends you an informational message describing the difference in file types.

Either the FMTOPT(*MAP) or FMTOPT(*NOCHK) option of the copy file function must be specified for copies from a physical or logical file to a physical file when there are fields with the same name in the from-file and to-file, but the data type for fields is as shown in the following table:

From-file field data type	To-file field data type
A (character)	J (DBCS-only)
O (DBCS-open)	J (DBCS-only)
O (DBCS-open)	E (DBCS-either)
E (DBCS-either)	J (DBCS-only)
J (DBCS-only)	G (DBCS-graphic)
O (DBCS-open)	G (DBCS-graphic)
E (DBCS-either)	G (DBCS-graphic)
G (DBCS-graphic)	J (DBCS-only)
G (DBCS-graphic)	O (DBCS-open)
G (DBCS-graphic)	E (DBCS-either)

When you use FMTOPT(*MAP) on the CPYF command to copy data to a DBCS-only field or DBCS-graphic field, the corresponding field in the from-file must not be:

- Less than a 2-byte character field

- An odd-byte-length character field
- An odd-byte-length DBCS-open field

If you attempt to copy with one of these specified in the from-field, an error message is sent.

When you copy double-byte data from one database file to another with the FMTOPT(*MAP) parameter specified, double-byte data is copied correctly. The system performs correct padding and truncation of double-byte data to ensure data integrity.

When using the CPYF command with FMTOPT(*MAP) to copy a DBCS-open field to a graphic field, a conversion error occurs if the DBCS-open field contains any SBCS data (including blanks).

Related reference

Copy Spooled File (CPYSPLF) command

Copy File (CPYF) command

Changing alphanumeric programs to DBCS programs:

If an alphanumeric application program uses externally described files, you can change that application program to a DBCS application program by changing the externally described files.

To convert an application program, follow these steps:

1. Create a duplicate copy of the source statements for the alphanumeric file that you want to change.
2. Change alphanumeric constants and literals to double-byte constants and literals.
3. Change fields in the file to the open (O) data type or specify the Alternative Data Type (IGCALTTYP) DDS keyword so that you can enter both double-byte and alphanumeric data in these fields. You might want to change the length of the fields as the double-byte data takes more space.
4. Store the converted file in a separate library. Give the file the same name as its alphanumeric version.
5. When you want to use the changed file in a job, change the library list, using the Change Library List (CHGLIBL) command, for the job in which the file is used. The library in which the DBCS display file is stored is then checked before the library in which the alphanumeric version of the file is stored.

Related concepts

IGCALTTYP (Alternative Data Type) keyword

Related reference

Change Library List (CHGLIBL) command

Entering DBCS text in CL commands:

You can use double-byte character data anywhere in a CL command that descriptive text can be used.

Enter double-byte character text as follows:

1. Begin the double-byte character text with a single quotation mark (′).
2. Enter a shift-out character.
3. Enter the double-byte character text.
4. Enter a shift-in character.
5. End the double-byte character text with a single quotation mark (′).

For example, to enter the double-byte character literal ABC, enter the following characters, where SO represents the shift-out character and SI represents the shift-in character:

```
'SOABCSI'
```

Limit the length of a double-byte character text description of an object to 14 double-byte characters, plus the shift control characters, to make sure that the description is properly displayed and printed.

DBCS conversion:

You can enter an alphanumeric entry or DBCS code and convert the entry or code to its related DBCS word. DBCS conversion is intended for Japanese character sets and its use is limited for application to other double-byte character sets.

When you use DBCS displays to enter double-byte data, you can use the various data entry methods supported on the display, or you can choose to use the system DBCS conversion support.

Specifically, you can convert the following characters:
- A string of alphanumeric characters to a DBCS word
- English alphanumeric characters to double-byte alphanumeric characters
- Alphanumeric Katakana to double-byte Hiragana and Katakana letters
- A DBCS code to its corresponding double-byte character
- A DBCS number to its corresponding double-byte character

Conversion dictionaries:

The DBCS conversion dictionary is a collection of alphanumeric entries and their related DBCS words. The system refers to the dictionary when performing DBCS conversion.

All DBCS conversion dictionaries have an object type of *IGCDCT. A system-supplied and a user-created dictionary are used with DBCS conversion.

User-created dictionary

A user-created dictionary contains any alphanumeric entries and related DBCS words that you choose to include. You might create a user dictionary to contain words unique to your business or words that you use regularly but that are not included in the system-supplied dictionary.

You can create one or more DBCS conversion dictionaries with any name and store them in any library. When performing DBCS conversion, however, the system only refers to the first user dictionary named QUSRIGCDCT in the user's library list, no matter how many dictionaries you have or what they are named. Make sure that the library list is properly specified so that the system checks the correct dictionary.

During DBCS conversion, the system checks QUSRIGCDCT before checking QSYSIGCDCT.

DBCS conversion dictionary commands

You can use the following commands to perform object management functions with the DBCS conversion dictionary. Specify the OBJTYPE(*IGCDCT) parameter when entering these commands:
- CHGOBJOWN: Change the owner of a DBCS conversion dictionary
- CHKOBJ: Check a DBCS conversion dictionary
- CRTDUPOBJ: Create a duplicate object of the dictionary
- DMPOBJ: Dump a DBCS conversion dictionary
- DMPSYSOBJ: Dump the system-supplied dictionary
- DSPOBJAUT: Display a user's authority to the dictionary
- GRTOBJAUT: Grant authority to use the dictionary
- MOVOBJ: Move the dictionary to another library

- RNMOBJ: Rename the dictionary
- RSTOBJ: Restore the dictionary
- RVKOBJAUT: Revoke authority to use the dictionary
- SAVOBJ: Save the dictionary
- SAVCHGOBJ: Save a changed dictionary

The system saves or restores DBCS conversion dictionaries when you use these commands:
- RSTLIB: Restore a library in which the dictionary is stored
- SAVLIB: Save a library in which the dictionary is stored
- SAVSYS: Save QSYSIGCDCT, the system DBCS conversion dictionary, when saving the system

You can use the following commands to create, edit, display, and delete a dictionary:
- CRTIGCDCT: Create DBCS Conversion Dictionary
- EDTIGCDCT: Edit DBCS Conversion Dictionary
- DSPIGCDCT: Display DBCS Conversion Dictionary
- DLTIGCDCT: Delete DBCS Conversion Dictionary

Related reference

Change Object Owner (CHGOBJOWN) command
Check Object (CHKOBJ) command
Create Duplicate Object (CRTDUPOBJ) command
Dump Object (DMPOBJ) command
Dump System Object (DMPSYSOBJ) command
Display Object Authority (DSPOBJAUT) command
Grant Object Authority (GRTOBJAUT) command
Move Object (MOVOBJ) command
Rename Object (RNMOBJ) command
Restore Object (RSTOBJ) command
Revoke Object Authority (RVKOBJAUT) command
Save Object (SAVOBJ) command
Save Changed Objects (SAVCHGOBJ) command
Restore Library (RSTLIB) command
Save Library (SAVLIB) command
Save System (SAVSYS) command
Create DBCS Conversion Dict (CRTIGCDCT) command
Edit DBCS Conversion Dict (EDTIGCDCT) command
Display DBCS Conversion Dict (DSPIGCDCT) command
Delete DBCS Conversion Dict (DLTIGCDCT) command

Working with DBCS conversion dictionaries:

You can create, edit, display, print, and delete conversion dictionaries.

Creating a DBCS conversion dictionary

To create a DBCS conversion dictionary, follow these steps:
1. Use the Create DBCS Conversion Dictionary (CRTIGCDCT) command.
2. Name the dictionary, QUSRIGCDCT, so it can be used during DBCS conversion. The system uses the dictionary if it is the first user-created dictionary found when searching a user's library list.

You might call the dictionary by another name while it is being created to prevent application programs from using it for conversion. Later, change the dictionary name using the Rename Object (RNMOBJ) command.

3. Use the EDTIGCDCT command to put entries and related words into the dictionary after creating it.

Editing a DBCS conversion dictionary

Use the Edit DBCS conversion dictionary (EDTIGCDCT) command to edit the DBCS conversion dictionary. Use editing to add user-defined characters to the dictionary, so that users can enter characters using DBCS conversion, and rearrange terms in a DBCS conversion dictionary to suit individual needs.

The display needed for use while editing the DBCS conversion dictionary depends on the value that you entered for the ENTRY parameter on the EDTIGCDCT command:

- If you specified a specific string with the ENTRY parameter or if you want to display double-byte characters, you must use a DBCS display.
- If you did not specify a specific string with the ENTRY parameter, or if you do not want to display double-byte characters, use either a DBCS display, or a 24-row by 80-column alphanumeric display.

You can perform the following editing operations on a user-created DBCS conversion dictionary:

- Add entries to the dictionary (including adding the first entries to the dictionary after it is created). The dictionary can contain as many as 99 999 entries.
- Delete entries from the dictionary.
- Change entries in the dictionary, such as replacing the DBCS words related to an alphanumeric entry.
- Move the DBCS words related to an alphanumeric entry to rearrange the order in which they appear during DBCS conversion.

The only editing function that you can perform with QSYSIGCDCT, the system-supplied dictionary, is to move DBCS words related to an alphanumeric entry. Move words in order to rearrange the order in which they appear during DBCS conversion.

Displaying and printing the DBCS conversion dictionary

Use the Display DBCS Conversion Dictionary (DSPIGCDCT) command to display and print the DBCS conversion dictionary. You can display or print the entire dictionary or just a certain part of it, depending on the value you specify for the ENTRY parameter. For example, to print the entry ABC from the dictionary QUSRIGCDCT and its related words, enter the following command:

```
DSPIGCDCT IGCDCT(DBCSLIB/QUSRIGCDCT)  +
ENTRY(ABC) OUTPUT(*PRINT)
```

To display all of the entries from the system-supplied dictionary QSYSIGCDCT and their related words, enter the following command:

```
DSPIGCDCT IGCDCT(QSYS/QSYSIGCDCT)
```

Deleting a DBCS conversion dictionary

Use the Delete DBCS Conversion Dictionary (DLTIGCDCT) command to delete a DBCS conversion dictionary from the system. In order to delete the dictionary, you must have object existence authority to the dictionary and object operational authorities to the library in which the dictionary is stored.

When you delete a dictionary, make sure that you specify the correct library name. It is possible that many users have their own dictionaries, each named QUSRIGCDCT, stored in their libraries. If you do not specify any library name, the system deletes the first DBCS conversion dictionary in your library list.

Related reference

Create DBCS Conversion Dictionary (CRTIGCDCT) command

Edit DBCS Conversion Dictionary (EDTIGCDCT) command

Display DBCS Conversion Dictionary (DSPIGCDCT) command

Delete DBCS Conversion Dictionary (DLTIGCDCT) command

Japanese DBCS conversion:

DBCS conversion is intended for Japanese character sets and its use is limited for application to other double-byte character sets.

When you use DBCS displays to enter double-byte data, you can use the various data entry methods supported on the display, or you may choose to use the i5/OS DBCS conversion support. DBCS conversion lets you enter an alphanumeric entry or DBCS code and convert the entry or code to its related DBCS word.

Specifically, DBCS conversion lets you convert the following characters:
- A string of alphanumeric characters to a DBCS word
- English alphanumeric characters to double-byte alphanumeric characters
- Alphanumeric Katakana to double-byte Hiragana and Katakana letters
- A DBCS code to its corresponding double-byte character
- A DBCS number to its corresponding double-byte character

Japanese system-supplied dictionary

The QSYSIGCDCT is the system-supplied dictionary that is stored in the library, QSYS. It is a collection of entries with a Japanese pronunciation, expressed in alphanumeric characters, and the DBCS words related to those entries. The system checks this dictionary second when performing DBCS conversion.

QSYSIGCDCT contains these entries:
- Personal names
  - Family names
  - First names
- Organization names
  - Private enterprises registered in the security market
  - Public corporations
  - Typical organizations in the central and local governments
  - Most universities and colleges
- Addresses
  - Public administration units within the prefectures
  - Towns and streets in 11 major cities
- Business terms, such as department names and position titles commonly used in enterprises
- Individual double-byte characters, including basic double-byte characters, as defined by IBM

You cannot add or delete entries from this dictionary. However, you may rearrange the related DBCS words so that the words used most frequently are displayed first during DBCS conversion.

SQL and DBCS:

The basic symbols of keywords and operators in the SQL language are single-byte characters that are part of all character sets supported by the IBM relational database products. Characters of the language are classified as letters, digits, or special characters.

SQL host identifiers and double-byte characters

A host-identifier is a name declared in the host program. The rules for forming a host-identifier are the rules of the host language, except that DBCS characters cannot be used.

SQL character subtypes and double-byte characters

Each character string is further defined as follows:

- **Bit data:**

  Data that is not associated with a coded character set and is never converted. The CCSID for bit data is 65535.
- **SBCS data:**

  Data in which every character is represented by a single byte. Each SBCS data character string has an associated CCSID. If necessary, an SBCS data character string is converted before it is used in an operation with a character string that has a different CCSID.
- **Mixed data:**

  Data that contains a mixture of characters from a single-byte character set (SBCS) and a double-byte character set (DBCS). Each mixed data character string has an associated CCSID. If necessary, a mixed data character string is converted before an operation with a character string that has a different CCSID. If mixed data contains a DBCS character, it cannot be converted to SBCS data.

The database manager does not recognize subclasses of double-byte characters, and it does not assign any specific meaning to particular double-byte codes. However, if you choose to use mixed data, then two single-byte EBCDIC codes are given special meanings:

- X'0E', the shift-out character, is used to mark the beginning of a sequence of double-byte codes.
- X'0F', the shift-in character, is used to mark the end of a sequence of double-byte codes.

In order for the database manager to recognize double-byte characters in a mixed data character string, the following condition must be met:

- Within the string, the double-byte characters must be enclosed between paired shift-out and shift-in characters.

  The pairing is detected as the string is read from left to right. The code X'0E' is recognized as a shift out character if X'0F' occurs later; otherwise, it is invalid. The first X'0F' following the X'0E' that is on a double-byte boundary is the paired shift-in character. Any X'0F' that is not on a double-byte boundary is not recognized.

  There must be an even number of bytes between the paired characters, and each pair of bytes is considered to be a double-byte character. There can be more than one set of paired shift-out and shift-in characters in the string.

The length of a mixed data character string is its total number of bytes, counting two bytes for each double-byte character and one byte for each shift-out or shift-in character.

When the job CCSID indicates that DBCS is allowed, CREATE TABLE will create character columns as DBCS-Open fields, unless FOR BIT DATA, FOR SBCS DATA, or an SBCS CCSID is specified. The SQL user will see these as character fields, but the system database support will see them as DBCS-Open fields.

SQL graphic strings:

A graphic string is a sequence of double-byte characters that do not include shift-out or shift-in characters.

The length of the string is the number of its characters. Like character strings, graphic strings can be empty.

Every graphic string has a CCSID that identifies a double-byte coded character set. If necessary, a graphic string is converted before it is used in an operation with a graphic string that has a different CCSID.

SQL fixed-length and double-byte characters

All values of a fixed-length graphic-string column have the same length, which is determined by the length attribute of the column. The length attribute must be between 1 through 16383 inclusive.

SQL graphic-string constants

A graphic-string constant is a varying-length graphic string. The length of the specified string cannot be greater than 16370.

In the normal form, the SQL delimiters and the G or the N are SBCS characters. The SBCS apostrophe (′) is the EBCDIC apostrophe, X′7D′.

In the PL/I form, the apostrophes and the G are DBCS characters. Two consecutive DBCS string delimiters are used to represent one string delimiter within the string. Notice that this PL/I form is only valid for static statements embedded in PL/I programs.

A hexadecimal graphic constant is also supported. The form of the hexadecimal graphic constant is:

```
GX'ssss'
```

In the constant, **ssss** represents a string from 0 to 32766 hexadecimal digits. The number of characters between the string delimiters must be an even multiple of 4. Each group of 4 digits represents a single graphic character. The hexadecimal for shift-in and shift-out (X′0E′ and X′0F′) are not included in the string.

The CCSID assigned to constants is the DBCS CCSID associated with the CCSID of the source unless the source is encoded in a foreign encoding scheme (such as ASCII). In this case, the CCSID assigned to the constant is the DBCS CCSID associated with the default CCSID of the application server when the SQL statement containing the constant is prepared. If there is no DBCS CCSID associated with the CCSID of the source, the CCSID is 65535.

SQL assignments and comparisons:

The basic operations of SQL are assignment and comparison. Assignment operations are performed during the running of CALL, INSERT, UPDATE, FETCH, and SELECT INTO statements. Comparison operations are performed during the running of statements that include predicates and other language elements such as MAX, MIN, DISTINCT, GROUP BY, and ORDER BY.

The basic rule for both operations is that the data type of the operands involved must be compatible. The compatibility rule also applies to UNION, concatenation, and the VALUE, COALESCE, MIN, and MAX scalar functions.

SQL string assignments and double-byte characters

The basic rule for string assignments is that the length of a string assigned to a column must not be greater than the length attribute of the column. (Trailing blanks are normally included in the length of the string. For string assignment, however, trailing blanks are not included in the length of the string.)

If the string contains mixed data, the assignment rules may require truncation within a sequence of double-byte codes. To prevent the loss of the shift-in character that ends the double-byte sequence, additional characters might be truncated from the end of the string, and a shift-in character added. In the truncated result, there is always an even number of bytes between each shift-out character and its matching shift-in character.

Character, DBCS-only, DBCS-open, and DBCS-either are not compatible with graphic types for assignment.

SQL conversion rules:

When two strings are compared, one of the strings is first converted, if necessary, to the coded character set of the other string.

Character conversion is necessary only if all of the following conditions are true:
- The CCSIDs of the two strings are different.
- Neither CCSID is 65535.
- The string selected for conversion is neither null nor empty.
- The CCSID conversion selection table indicates that conversion is necessary.

If one string has an SBCS CCSID and the other is the same type of operand and has a mixed data CCSID, the SBCS data character string is converted. Otherwise, the string selected for conversion depends on the type of each operand. The following table shows which operand is selected for conversion, given the operand types.

First operand	Column value (second operand)	Derived value (second operand)	Special register (second operand)	Constant (second operand)	Host variable (second operand)
Column value	Second	Second	Second	Second	Second
Derived Value	First	Second	Second	Second	Second
Special Register	First	First	Second	Second	Second
Constant	First	First	First	Second	Second
Host Variable	First	First	First	First	Second

A host variable containing data in a foreign encoding scheme is always effectively converted to the native encoding scheme before it is used in any operation. The previous rules are based on the assumption that this conversion has already occurred.

An error occurs if a character of the string cannot be converted or the CCSID Conversion Selection Table is used but does not contain any information about the pair of CCSIDs. A warning occurs if a character of the string is converted to the substitution character.

DBCS code schemes

IBM supports two DBCS code schemes: one for the host systems and the other for personal computers.

The DBCS code scheme for host systems has the following code-range characteristics:
- First byte: hex 41 to hex FE
- Second byte: hex 41 to hex FE
- Double-byte blank: hex 4040

Shift-control characters

When the host code scheme is used, the system uses shift-control characters to identify the beginning and end of a string of double-byte characters. The shift-out (SO) character, hex 0E, indicates the beginning of a double-byte character string. The shift-in (SI) character, hex 0F, indicates the end of a double-byte character string.

Each shift-control character occupies the same amount of space as one alphanumeric character. By contrast, double-byte characters occupy the same amount of space as two alphanumeric characters.

When double-byte characters are stored in a graphic field or a variable of graphic data type, there is no need to use shift control characters to surround the double-byte characters.

Incorrect and undefined double-byte code

Incorrect double-byte code has a double-byte code value that is not in the valid double-byte code range. This is in contrast to undefined double-byte code where the double-byte code is valid, but no graphic symbol has been defined for the code.

Supported DBCS code ranges

The i5/OS operating system supports the following DBCS character-set code ranges:
- Japanese character-set code range
- Korean character-set code range
- Simplified Chinese character-set code range
- Traditional Chinese character-set code range

Related concepts

Double-byte character set support

Database file management

DBCS font tables

DBCS font tables contain the images of the double-byte extended characters used on the i5/OS operating system. The system uses these images to display and print extended characters when they are not resident on the device.

The following DBCS font tables are objects that you can save or restore. These font tables are distributed with the DBCS national language versions of the i5/OS licensed program:

QIGC2424
: A Japanese DBCS font table used to display and print extended characters in a 24-by-24 dot matrix image. The system uses the table with Japanese displays, printers attached to displays, 5227 Model 1 Printer, and the 5327 Model 1 Printer.

QIGC2424C
: A Traditional Chinese DBCS font table used to print extended characters in a 24-by-24 dot matrix image. The system uses the table with the 5227 Model 3 Printer and the 5327 Model 3 Printer.

QIGC2424K
: A Korean DBCS font table used to print extended characters in a 24-by-24 dot matrix image. The system uses the table with the 5227 Model 2 Printer and the 5327 Model 2 Printer.

QIGC2424S
: A Simplified Chinese DBCS font table used to print extended characters in a 24-by-24 dot matrix image. The system uses the table with the 5227 Model 5 Printer.

QIGC3232
A Japanese DBCS font table used to print characters in a 32-by-32 dot matrix image. The system uses the table with the 5583 Printer and the 5337 Model 1 Printer.

QIGC3232S
A Simplified Chinese DBCS font table used to print characters in a 32-by-32 dot matrix image. The system uses the table with the 5337 Model R05 Printer.

All DBCS font tables have an object type of *IGCTBL. You can find instructions for adding user-defined characters to DBCS font tables in the *ADTS/400: Character Generator Utility*, SC09-1769-00 book.

DBCS font table commands

The following commands allow you to manage and use DBCS font tables:
- Check DBCS Font Table (CHKIGCTBL)
- Copy DBCS Font Table (CPYIGCTBL)
- Delete DBCS Font Table (DLTIGCTBL)
- Start Font Management Aid (STRFMA)

Locating an existing font table

Use the Check DBCS Font Table (CHKIGCTBL) command to find out if a DBCS font table exists in your system.

For example, to find out if the table QIGC2424 exists, enter:

```
CHKIGCTBL IGCTBL(QIGC2424)
```

If the table does not exist, the system responds with a message. If the table does exist, the system returns without a message.

Check for the existence of a table when adding a new type of DBCS workstation to make sure that the table used by the device exists in the system.

Related reference

Check DBCS Font Table (CHKIGCTBL) command

Copy DBCS Font Table (CPYIGCTBL) command

Delete DBCS Font Table (DLTIGCTBL) command

Start Font Management Aid (STRFMA) command

Copying a DBCS font table:

You can use the Copy DBCS Font Table (CPYIGCTBL) command to copy a DBCS font table to or from a tape, diskette, or physical file.

The DBCS font tables are saved when you use the Save System (SAVSYS) command so you do not have to use the CPYIGCTBL command when performing normal system backup.

A physical file used to save and restore table information must have a minimum record length of 74 bytes.

Copying a table onto a tape, a diskette, or a physical file

You should copy a DBCS font table onto a tape, a diskette, or a physical file in the following instances:
- Before deleting that table
- After new user-defined characters are added to the tables

- When planning to use the tables on another system

To copy a DBCS font table onto a tape, a diskette, or a physical file, follow these steps:

1. If copying a DBCS font table onto a tape or diskettes, make sure that the tape or diskettes are initialized to the *DATA format. If necessary, initialize the tape or diskettes by specifying the FMT(*DATA) parameter on the Initialize Diskette (INZDKT) command.
2. Load the initialized tape or diskette onto the system.
3. Enter the CPYIGCTBL command as follows:
   a. Choose the value OPTION(*OUT).
   b. Use the DEV parameter to select the device to which you want to copy the table. A value of *FILE specifies that the DBCS font table is saved to a physical file.
   c. Use the SELECT and RANGE parameters to specify which portion of the table you want copied from the system.
4. Press the Enter key. The system copies the DBCS font table onto the specified medium or into a physical file.
5. Remove the tape or the diskette after the system finishes copying the table.

Copying a DBCS font table from a tape, a diskette, or a physical file

Use the Copy DBCS Font Table (CPYIGCTBL) command to copy a DBCS font table from a tape, a diskette, or a physical file onto the system. The system automatically creates the DBCS font table again when copying its contents if the following conditions are true:

- The specified table does not already exist in the system.
- The medium or physical file from which you are copying the table contains all of the IBM-defined double-byte characters.
- SELECT(*ALL) or SELECT(*SYS) is specified on the CPYIGCTBL command.

Related reference

Copy DBCS Font Table (CPYIGCTBL) command

Save System (SAVSYS) command

Deleting a DBCS font table:

You can delete an unused DBCS font table to free storage space on your system. Use the Delete DBCS Font Table (DLTIGCTBL) command to delete a DBCS font table from the i5/OS operating system.

For example, if you do not plan to use Japanese printer 5583 or 5337 with your system, font table QIGC3232 is not needed and can be deleted.

To delete a table, follow these steps:

1. Optional: Copy the table onto a tape, a diskette, or a physical file. If you do not copy the table before deleting it, you will not have a copy of the table for future use.
2. Vary off all devices using that table.
3. Enter the DLTIGCTBL command. For example, to delete the DBCS font table QIGC3232, enter:
   ```
   DLTIGCTBL IGCTBL(QIGC3232)
   ```
4. Press the Enter key. The system sends an inquiry message to the system operator message queue for you to confirm your intention to delete a DBCS table.
5. Respond to the inquiry message. The system sends you a message when it has deleted the table.

Note: Do not delete a DBCS font table if any device using that table is currently varied on. Also, make sure that the affected controller is not varied on. If you try to delete the table while the device and controller are varied on, the system reports any devices attached to the same controller as those devices, and the controller as damaged the next time you try to print or display extended characters on an affected device. If such damage is reported, follow these steps:

a. Vary off the affected devices, using the Vary Configuration (VRYCFG) command.
b. Vary off the affected controller.
c. Vary on the affected controller.
d. Vary on the affected devices.
e. Continue normal work.

Related reference

Delete DBCS Font Table (DLTIGCTBL) command

Vary Configuration (VRYCFG) command

DBCS font files

The i5/OS operating system provides DBCS font files. *DBCS font files* are physical files that contain frequently used double-byte characters. When using the character generator utility, you can use the characters in these files as the base for a new, user-defined character.

These files are supplied with read-only authority as they are not to be changed. If you do not use character generator utility or the Advanced DBCS Printer Support licensed program, you can delete these files to save space. They all exist in the QSYS library.

The following DBCS font files are distributed with the DBCS national language versions of the i5/OS licensed program. They are used as a reference for the character generator utility (CGU) and the Advanced DBCS Printer Support licensed program.

QCGF2424
A Japanese DBCS font file used to store a copy of the Japanese DBCS basic character images.

QCGF2424K
A Korean DBCS font file used to store a copy of the Korean DBCS basic character images.

QCGF2424C
A Traditional Chinese DBCS font file used to store a copy of the Traditional Chinese DBCS basic character images.

QCGF2424S
A Simplified Chinese DBCS font file used to store a copy of the Simplified Chinese DBCS basic character images.

DBCS sort tables

DBCS sort tables contain the sort information and collating sequences of all the double-byte characters used on the system. The sort utility on the system uses these tables to sort double-byte characters.

DBCS sort tables are objects that you can save, restore and delete. Using the character generator utility you can also add, delete and change entries in these tables corresponding to the image entries in the DBCS font tables. For Japanese use only, you can also copy the DBCS master sort table to and from a data file.

The following DBCS sort tables are distributed with the DBCS national language versions of the i5/OS licensed program:

QCGMSTR
A Japanese DBCS master sort table used to store the sort information for the Japanese double-byte character set.

QCGACTV
: A Japanese DBCS active sort table used to store the sort collating sequences for the Japanese double-byte character set.

QCGMSTRC
: A Traditional Chinese DBCS master sort table used to store the sort information for the Traditional Chinese double-byte character set.

QCGACTVC
: A Traditional Chinese DBCS active sort table used to store the sort collating sequences for the Traditional Chinese double-byte character set.

QCGACTVK
: A Korean DBCS active sort table used to map Hanja characters to Hangeul characters with equivalent pronunciation.

QCGMSTRS
: A Simplified Chinese DBCS master sort table used to store the sort information for the Simplified Chinese double-byte character set.

QCGACTVS
: A Simplified Chinese DBCS active sort table used to store the sort collating sequences for the Simplified Chinese double-byte character set.

You can sort Japanese, Korean, Simplified Chinese, and Traditional Chinese double-byte characters. Each of these languages have two DBCS sort tables, a DBCS master sort table and a DBCS active sort table, except for Korean which has only a DBCS active sort table. The DBCS master sort table contains sort information for all defined DBCS characters. The DBCS active sort table for Japanese, Simplified Chinese, and Traditional Chinese is created from the master sort table information and contains the collating sequences for the double-byte characters of that given language. These collating sequences have a purpose similar to the EBCDIC and ASCII collating sequences for the single-byte alphanumeric character set. For Korean characters, the Hangeul characters are assigned both their collating sequence as well as their DBCS codes according to their pronunciation. Hence, a separate collating sequence is not required, and each of the Hanja characters is mapped to a Hangeul character of the same pronunciation using the DBCS active sort table QCGACTVK.

All DBCS sort tables have an object type of *IGCSRT.

Commands for DBCS sort tables

The following commands allow you to manage and use DBCS sort tables.

- Check Object (CHKOBJ)
- Save Object (SAVOBJ)
- Restore Object (RSTOBJ)

Using existing DBCS sort tables

You can save the tables to a tape or diskette, delete them from the system, and restore them to the system. You can also add sort information for each user-defined character, and add that character to the DBCS collating sequence, as you create it using the character generator utility.

Finding existing DBCS sort table

Use the Check Object (CHKOBJ) command to find out if a DBCS sort table exists on your system. For example, to find out if the table QCGMSTR exists, enter the following command:

```
CHKOBJ OBJ(QSYS/QCGMSTR) OBJTYPE(*IGCSRT)
```

If the table does not exist, the system responds with a message. If the table does exist, the system returns without a message.

Check for the existence of a DBCS active sort table when you want to sort double-byte characters for the first time. The DBCS active table for the DBCS language must exist to sort the characters.

Related reference

Check Object (CHKOBJ) command

Save Object (SAVOBJ) command

Restore Object (RSTOBJ) command

Saving and restoring a DBCS sort table:

To save DBCS sort tables to a tape or diskette, use the Save Object (SAVOBJ) command. To restore DBCS sort tables from a tape or diskette, use the Restore Object (RSTOBJ) command.

Saving a DBCS sort table to tape or diskette

Save a DBCS sort table onto tape or diskette in the following instances:
- Before deleting that table
- After information is added, updated, or changed in the tables using the character generator utility
- When planning to use the tables on another system

Use the Save Object (SAVOBJ) command to save a DBCS sort table onto tape or diskette. Specify *IGCSRT for the object type.

The DBCS sort tables are saved when you use the SAVSYS command so you do not have to use the SAVOBJ command when performing normal system backup.

Restoring a DBCS sort table from tape or diskette

Use the RSTOBJ command to restore a DBCS sort table from a tape or a diskette onto the system. The tables on the tape or diskette must previously have been saved using the SAVOBJ command. Specify *IGCSRT for the object type. The system automatically re-creates the DBCS sort table when the specified table does not already exist in the system.

These tables must be restored to the QSYS library for the system to know they exist. For that reason, RSTOBJ restores *IGCSRT objects only to the QSYS library and only if the objects do not already exist there.

Related reference

Save Object (SAVOBJ) command

Save System (SAVSYS) command

Restore Object (RSTOBJ) command

Deleting a DBCS sort table:

You can delete an unused DBCS sort table to free disk space, but you should always first save a copy of the table using the Save Object (SAVOBJ) command. Use the Delete DBCS Sort Table (DLTIGCSRT) command to delete a DBCS sort table from the system.

You should delete the DBCS master sort table for a DBCS language if either of the following conditions are true:
- You will not be creating any new characters for that language using the character generator utility.
- You will not be using the sort utility to sort characters for that language.

You should delete the DBCS active sort table for a DBCS language if you will not be using the sort utility to sort characters for that language. The DBCS active sort table must be on the system to use the sort utility for this language.

To delete a table, follow these steps:
1. Optional: Save the table onto tape or diskettes. If you do not save the table onto removable media before deleting it, you will not have a copy of the table for future use.
2. Enter the DLTIGCSRT command. For example, to delete the DBCS sort table QCGACTV, enter:
   ```
   DLTIGCSRT IGCSRT(QCGACTV)
   ```
3. Press the Enter key. The system sends you a message when it has deleted the table.

Related reference

Save Object (SAVOBJ) command

DBCS field definition

Consider the characteristics of DBCS data when you define a DBCS field.
- Each DBCS character is 2 bytes long.
- The length of a DBCS character string is always even.
- Shift-out (SO) and shift-in (SI) control characters are required at the beginning and end of the DBCS character string, except for graphic-data type fields. Together, these characters are 2 bytes long.
- The system treats DBCS data the same as character data, and therefore cannot perform arithmetic operations on it.
- The following DBCS data types can be used to identify DBCS fields:
  – J (DBCS-only) for fields that can contain only bracketed DBCS data.
  – E (DBCS-either) for fields that can contain bracketed DBCS or SBCS data, but not both.
  – O (DBCS-open) for fields that can contain both SBCS and bracketed DBCS data.
  – G (DBCS-graphic) for fields that can contain graphic data without the SO and SI control characters.

  Data type O is allowed in all types of files. Data types J and E are allowed only in database and display files. Data type G is allowed in database, display, and printer files. In most cases, the operating system automatically inserts shift-out and shift-in characters. An exception is when data is written into a data type G field in a database file.

Related reference

DDS concepts

Related information

Application Display Programming PDF

Working with locales

Locales are used primarily in ILE-based application programs. Additionally, the Retrieve Locale Information (OPM, QLGRTVLC; ILE, QlgRetrieveLocaleInformation) API retrieves one or all categories of a locale.

Benefits of using locales in global applications

Applications can be created independent of language, cultural data, or specific characters. Locales can be accessed to provide this type of support to any integrated language environment-based application.

For example, the LC_TIME category within a locale can be defined in any of the following ways, or in any combination that is convenient for the environment in which the application runs:
- HH:MM:SS
- MM:SS:HH

- SS:MM:HH

Creating locales

Locales are created using the Create Locale (CRTLOCALE) command. The source file used to create the locale is named QLOCALESRC, in the QSYSLOCALE library. This library is loaded with option 21 of the operating system. These source files cannot be changed. Instead, they must be copied and then edited if changes are required.

Related concepts

"Locales" on page 8
A *locale* is an object that can determine how data is processed, printed, and displayed.

"Installing and enabling locales" on page 44
If you are installing a new release, you can request that library QSYSLOCALE be installed on the system at that time.

"System-supplied locales and recommended CCSIDs" on page 360
The system-supplied locale source definition file members are in the optionally installable library QSYSLOCALE in the QLOCALESRC source file. The source file members are encoded in CCSID 37 and are read only.

"Example: Creating and enabling a locale" on page 225
This example contains the steps necessary for creating and enabling a locale.

Related reference

Application programming interfaces (APIs)

Create Locale (CRTLOCALE) command

Related information

Retrieve Locale Information (QLGRTVLC, QlgRetrieveLocaleInformation) API

Locale restrictions

Certain restrictions apply when you use locales to set job attributes.

- The locale CCSID must be an EBCDIC single-byte CCSID for an SBCS system.
- The locale CCSID must be an EBCDIC, single-byte character set (SBCS), or mixed-byte CCSID for a DBCS-capable system.
- The locale object must exist in the QSYS file system.
- The DATFMT, DATSEP, TIMSEP, and DECFMT parameters within the locale must be valid values supported as job attributes.
- If you want sort sequence support from the locale, you must use the CPYSYSCOL keyword.

Related concepts

Work management

"LC_COLLATE category" on page 203
The LC_COLLATE category defines character or string collation information. Within LC_COLLATE you can specify a sort sequence to use using the cpysyscol keyword. The *cpysyscol* keyword value is used in place of the LC_COLLATE category definitions.

Locale categories

This table describes the locale categories that are supported on the i5/OS operating system.

Locale category	Description
LC_COLLATE	Defines character or string collation information
LC_CTYPE	Defines character classification, case conversion, and other character attributes.
LC_MESSAGES	Defines the format for affirmative and negative responses.

Locale category	Description
LC_MONETARY	Defines rules and symbols for formatting monetary numeric information.
LC_NUMERIC	Defines a list of rules and symbols for formatting non-monetary numeric information.
LC_TIME	Defines a list of rules and symbols for formatting time and date information.
LC_TOD	Defines rules for daylight saving time and time zone information.

Note: A locale source file cannot contain duplicate categories.

Locale category source definitions

The category source definition consists of:

- The category header (*category name*), where the category name must be all uppercase characters.
- The associated keyword/value pairs that comprise the category body. Keywords can be all uppercase, all lowercase, or mixed case characters.
- The category trailer (which consists of END *category-name*)

For example:

```
LC_CTYPE
source for LC_CTYPE category
END LC_CTYPE
```

Lines preceding the first category header can be used to change the comment character and the escape character. The comment_char (the default is #) and escape_char (the default is \) keywords can be used to change these characters. The following example shows how to change the comment character and escape character to * and / :

```
comment_char <asterisk>
escape_char  <slash>
```

Note: This example uses symbolic names to represent the '*' and '/' characters.

The source for all categories is specified using keywords, strings, character literals, and character symbols:

Keywords

Each keyword identifies either a definition or a rule. The remainder of the statement containing the keyword contains the operands to the keyword. Operands are separated from the keyword by one or more blank characters. A statement may be continued on the next line by placing an escape_char as the last character before the newline or linefeed character that ends the line.

Lines containing the comment_char in the first column are treated as comment lines. Comment lines cannot be continued on a subsequent line using an escape character. \ is the default escape character. However, the escape character can be defined to be any character by the user.

Strings

Strings must be enclosed in quotation marks. Quotation marks within strings can be represented in two ways:

- Literally. The escape character can be followed by quotation marks.
- A symbolic name. For example, <quotation-mark>.

A string can be continued on the next line by placing an escape_char as the last character before the newline or linefeed character that ends the line.

A string is a sequence of character symbols or literals enclosed by quotation ("") characters. For example:

```
"<A><B><C>"
```

Character literals
A character literal is the character itself.

Character symbols
A character symbol begins with the < (less-than) character, followed by non-control characters, and ends with the > (greater-than) character. For example, <A> is a valid character symbol (symbolic name). Any character symbol referenced in the source file should be one of the predefined system-supplied symbols. The system-supplied symbolic names are in the source file member QLGSYMBOL in the QLOCALESRC source file in the optionally installable library QSYSLOCALE.

In the event that the system does not contain a predefined symbolic name for a character, the UCS-2 level-1 format is allowed. The UCS-2 format is based on the character set defined in ISO/IEC 10646. The UCS-2 format may also be used in place of the predefined symbolic names. Here is an example of the UCS-2 symbolic name format:

```
<Uxxxx>
```

where 'xxxx' are 4 hexadecimal digits. For example, <U0041>. The hexadecimal number 0041 within this symbolic name is the UCS-2 code point that represents the character 'A'.

Each locale category must be explicitly defined in a locale definition source file.

Related concepts

"System-supplied locales and recommended CCSIDs" on page 360
The system-supplied locale source definition file members are in the optionally installable library QSYSLOCALE in the QLOCALESRC source file. The source file members are encoded in CCSID 37 and are read only.

"Example: POSIX locale" on page 233
This example shows the POSIX (or C) locale categories and the source.

"Examples: Locale programming" on page 221
These examples illustrate how locales work in the i5/OS operating system, and how you can work with locales and use locales for a multilingual system environment.

LC_COLLATE category:

The LC_COLLATE category defines character or string collation information. Within LC_COLLATE you can specify a sort sequence to use using the cpysyscol keyword. The *cpysyscol* keyword value is used in place of the LC_COLLATE category definitions.

A collation element is the unit of comparison for collation. A collation element may be a character or a sequence of characters. Every collation element in the locale has a set of weights, which determine if the collation element collates before, equal to, or after the other collation elements in the locale. Each collation element is assigned collation weights by the CRTLOCALE command when the locale definition source file is created. These collation weights are then used by applications programs that compare strings.

Every character defined in the CCSID that is specified in the CRTLOCALE command is itself a collating element. Additional collating elements can be defined using the collating-element statement. The syntax is:

```
collating-element character-symbol from string
```

The LC_COLLATE category begins with the LC_COLLATE keyword and ends with the END LC_COLLATE keyword.

The following keywords are recognized in the LC_COLLATE category:

cpysyscol
This statement specifies that a system collating sequence table is to be used for the collation information for the category. If the locale is intended to be used to set the sort sequence table for the job, then it is required that the CPYSYSCOL keyword be used. If the CPYSYSCOL keyword is specified, no other keyword can be specified. The syntax for the CPYSYSCOL keyword is:

```
CPYSYSCOL sort sequence path name;langid
```

The *sort sequence path name* is a string specifying a fully expanded path name of an existing sort sequence table to use as the definition for this category. The path name delimiter must be a slash (/). Other valid values are strings containing one of the following sort sequence tables:

***JOB** The sort sequence of the job.

***LANGIDUNQ**
The unique-weighted sort sequence table that is associated with the language identifier requested parameter.

***LANGIDSHR**
The shared-weighted sort sequence table that is associated with the language identifier requested parameter.

***HEX** The sort sequence according to the hexadecimal value of the characters.

The *langid* is a string specifying the language identifier of the sort sequence table to be used. All *langid* must be in uppercase. Valid values are strings containing one of the following language identifiers:

***JOB** Use the language identifier of the job.

language id
A valid 3-character language identifier. For example, Danish can be DAN.

Collating-element
The collating-element statement specifies multi-character collating elements. The syntax for the collating-element statement is:

```
collating-element symbolic-name from string
```

The symbolic-name value defines a collating element that is a string of one or more characters as a single collating element. The symbolic-name value cannot duplicate any system predefined symbolic name, or any other symbolic name defined in this collation definition. The string value specifies a string of two or more characters or character symbols that define the symbolic-name value. The following are examples of the syntax for the collating-element statement:

```
collating-element <ch> from "<c><h>"
collating-element <e-acute> from "<acute><e>"
collating-element <11> from "<1><1>"
```

A symbolic-name value defined by the collating-element statement is recognized only with the LC_COLLATE category.

Order_start
The order_start statement may be followed by one or more collation order statements, assigning collation weights to collating elements. This statement is required. The syntax for the order_start statement is:

```
order_start  sort-rules;sort-rules;...sort-rules collation-order-statements order_end
```

The sort-rules have the following syntax:

```
directive, directive,...directive
```

where directive is one of the directives; **forward, backward,** and **position.**

The sort-rules directives are optional. If present, they define the rules to apply during string comparison. The number of specified sort-rules directives defines the number of weights each collating element is assigned (that is, the number of collation orders in the locale). If no sort-rules directives are present, one forward directive is assumed.

If present, the first sort-rules directive applies when comparing strings using primary weight, the second when comparing strings using the secondary weight, and so on. Each set of sort-rules directives is separated by a ; (semicolon). A sort-rules directive consists of one or more comma-separated directives. The following directives are supported:

Forward
: Specifies that collation weight comparisons proceed from the beginning of a string toward the end of the string.

Backward
: Specifies that collation weight comparisons proceed from the end of a string toward the beginning of the string.

Position
: Specifies that collation weight comparisons consider the relative position of non-ignored elements in the string. That is, if strings compare equal, the element with the shortest distance from the starting point of the string collates first.

The forward and backward directives are mutually exclusive. The following example shows the syntax for the sort-rules directives:

```
order_start  forward;backward
```

Order_end
: This keyword ends collating order entries introduced by the order_start keyword.

The order of the characters and elements specified between the order_start and order_end keywords defines the character order used in range expressions and regular expressions. If no weights are assigned to the characters, then the character order also becomes the collation sequence weight.

Special symbols

Special symbols are required to be all upper-case characters. The following special symbols can be used in the LC_COLLATE category:

- IGNORE

  The optional operands for each collation element are used to define the primary, secondary, or subsequent weights for the collating element. The special symbol IGNORE is used to indicate a collating element that is to be ignored when strings are compared.

- UNDEFINED

  All characters in the character set must be placed in the collation order, either explicitly or implicitly, by using the Undefined symbol. The UNDEFINED symbol includes all coded character set values not specified explicitly. These characters are inserted in the character collation order at the point indicated by the Undefined symbol in the order of their character code page values. If a collating weight is not explicitly specified for the UNDEFINED symbol, then by default, all of the undefined characters are assigned the same collating weight equal to the relative order of the first undefined character in the collating sequence. If no UNDEFINED special symbol exists and the collation order does not specify all collation elements from the coded character set, a warning is issued and all undefined characters are placed at the end of the character collation order and be given the same collating weight.

Example 1

Here is an example of a collation order statement in the LC_COLLATE locale definition source file category.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

The text that follows the LC_COLLATE keywords has been added for clarity and does not appear in the locale source file.

```
order_start  forward;backward
#           The order_start has two sort rules specified:
#           forward and backward

UNDEFINED   IGNORE;IGNORE
#           The UNDEFINED special symbol indicates that
#           all characters in the CCSID of the locale
#           that are not specified in the definition
#           are ignored for collation purposes.

<LOW>
#           <LOW> is a collating symbol that is ordered
#           after all undefined characters. For example, if there
#           were only two undefined characters, then the <LOW> symbol
#           would be third in the order.

#           All collating elements between <space> and <a> have the
#           same primary equivalence class and individual secondary
#           weights based on their coded character set values.

<a>      <a>;<a>
<a-acute> <a>;<a-acute>
<a-grave> <a>;<a-grave>
<A>       <a>;<A>
<A-acute> <a>;<A-acute>
<A-grave> <a>;<A-grave>
#           All characters between <a> and <A-grave> belong to the
#           same primary equivalence class because they have the same
#           primary weight.

<ch>      <ch>;<ch>
<Ch>      <ch>;<Ch>
#           The <c><h> multi-character collating element is
#           represented by the <ch> collating symbol and belongs to the
#           same primary equivalence class as the <Ch> multi-character
#           collating element.

<s>       <s>;<s>
<eszet>   "<s><s>";<s>
#           A one-to-many mapping is indicated by the <eszet>
#           character collated as an <s><s> string. That is, one
#           <eszet> character is expanded to <s><s> characters before
#           comparing.

<HIGH>
order_end
```

Example 2

Here is an example of a CPYSYSCOL statement in the LC_COLLATE locale definition source file category.

```
LC_COLLATE

CPYSYSCOL "//QSYS.LIB//QLA10025S.TBL";"ENU"

END LC_COLLATE
```

Related concepts

"Character sorting" on page 155
Traditionally, information is displayed in sorted order to enable users to easily find the items they are looking for. However, users of different languages might have very different expectations of what a sorted list should look like.

"Locale restrictions" on page 201
Certain restrictions apply when you use locales to set job attributes.

"Language identifiers and associated default CCSIDs" on page 358
This table shows the language identifiers and the job default CCSID (DFTCCSID) values associated with those identifiers.

LC_CTYPE category:

The LC_CTYPE category defines character classification, case conversion, and other character attributes.

The LC_CTYPE category begins with an LC_CTYPE category header and ends with an END LC_CTYPE category trailer.

All operands for LC_CTYPE category statements are defined as lists of characters. Each list consists of one or more semicolon-separated characters or symbolic character names.

The following keywords are recognized in the LC_CTYPE category. In the descriptions, the term *automatically included* means that an error does not occur if the referenced characters are included or omitted. The characters are provided if they are missing, and are accepted if they are present. In the event that the *automatically included* characters do not exist in the CCSID that you want to create the locale, a warning is issued by the CRTLOCALE command.

upper Defines uppercase letter characters. No character defined by the cntrl, digit, punct, or space keyword can be specified. At a minimum, the uppercase letters A through Z are automatically included.

lower Defines lowercase letter characters. No character defined by the cntrl, digit, punct, or space keyword can be specified. At a minimum, the lowercase letters a through z are automatically included.

alpha Defines all letter characters. No character defined by the cntrl, digit, punct, or space keyword can be specified. Characters defined by the upper and lower keywords are automatically included in this character class.

digit Defines numeric digit characters. Only the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 can be specified.

space Defines white space characters. No character defined by the upper, lower, alpha, digit, graph, or xdigit keyword can be specified. At a minimum, the <space>, <form-feed>, <newline>, <carriage return>, <tab>, <vertical-tab> characters, and any characters defined by the blank keyword, are automatically included.

cntrl Defines control characters. No character defined by the upper, lower, alpha, digit, punct, graph, print, or xdigit keywords can be specified.

punct Defines punctuation characters. A character defined as the <space> character and characters defined by the upper, lower, alpha, digit, cntrl, or xdigit keyword cannot be specified.

graph Defines printable characters, excluding the <space> character. If this keyword is not specified, characters defined by the upper, lower, alpha, digit, xdigit, and punct keywords are automatically included in this character class. No character defined by the cntrl keyword can be specified.

print Defines printable characters, including the <space> character. If this keyword is not specified, the <space> character and characters defined by the upper, lower, alpha, digit, xdigit, and punct keywords are automatically included in this character class. No character defined by the cntrl keyword can be specified.

xdigit Defines hexadecimal digit characters. Only the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 and the letters A, B, C, D, E, F, a, b, c, d, e, and f can be specified. If not specified, the xdigit class defaults to the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 and the letters A, B, C, D, E, F, a, b, c, d, e, and f.

blank Defines blank characters. If this keyword is not specified, the <space> and <horizontal-tab> characters are included in this character class.

toupper
Defines the mapping of lowercase characters to uppercase characters. Operands for this keyword consist of semicolon-separated character pairs. Each character pair is enclosed in ( ) (parentheses) and separated from the next pair by a , (comma). The first character in each pair is considered lowercase; the second character is considered uppercase. Only characters defined by the lower and upper keywords can be specified.

tolower
Defines the mapping of uppercase characters to lowercase characters. Operands for this keyword consist of semicolon-separated character pairs. Each character pair is enclosed in ( ) (parentheses) and separated from the next pair by a , (comma). The first character in each pair is considered uppercase; the second character is considered lowercase. Only characters defined by the lower and upper keywords can be specified.

Note: The **tolower** keyword is optional. If this keyword is not specified, the mapping defaults to the reverse mapping of the **toupper** keyword, if specified. If the **toupper** keyword is not specified, the mapping defaults to the **C** locale.

Example

Here is an example of the LC_CTYPE category in a locale definition source file.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
LC_CTYPE

#"alpha" is by default "upper" and "lower"
#"print" is by default "alpha", "digit", "punct", and the space character
#"graph" is by default "alnum" and "punct"
#"tolower" is by default the reverse mapping of "toupper"
#
upper   <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>
#
lower <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>
#
digit <zero>;<one>;<two>;<three>;<four>;<five>;<six>;\
<seven>;<eight>;<nine>
#
space <tab>;<newline>;<vertical-tab>;<form-feed>;\
<carriage-return>;<space>
#
cntrl <alert>;<backspace>;<tab>;<newline>;<vertical-tab>;\
<form-feed>;<carriage-return>;<NUL>;<SOH>;<STX>;\
<ETX>;<EOT>;<ENQ>;<ACK>;<SO>;<SI>;<DLE>;<DC1>;<DC2>;\
<DC3>;<DC4>;<NAK>;<SYN>;<ETB>;<CAN>;<EM>;<SUB>;\
<ESC>;<IS4>;<IS3>;<IS2>;<IS1>;<DEL>
#
```

```
punct <exclamation-mark>;<quotation-mark>;<number-sign>;\
<dollar-sign>;<percent-sign>;<ampersand>;<asterisk>;\
<apostrophe>;<left-parenthesis>;<right-parenthesis>;\
<plus-sign>;<comma>;<hyphen>;<period>;<slash>;\
<colon>;<semicolon>;<less-than-sign>;<equals-sign>;\
<greater-than-sign>;<question-mark>;<commercial-at>;\
<left-square-bracket>;<backslash>;<circumflex>;\
<right-square-bracket>;<underline>;<grave-accent>;\
<left-curly-bracket>;<vertical-line>;<tilde>;\
<right-curly-bracket>
#
xdigit <zero>;<one>;<two>;<three>;<four>;<five>;<six>;\
<seven>;<eight>;<nine>;<A>;<B>;<C>;<D>;<E>;<F>;\
<a>;<b>;<c>;<d>;<e>;<f>
#
blank <space>;<tab>
#
toupper (<a>,<A>);(<b>,<B>);(<c>,<C>);(<d>,<D>);(<e>,<E>);\
(<f>,<F>);(<g>,<G>);(<h>,<H>);(<i>,<I>);(<j>,<J>);\
(<k>,<K>);(<l>,<L>);(<m>,<M>);(<n>,<N>);(<o>,<O>);\
(<p>,<P>);(<q>,<Q>);(<r>,<R>);(<s>,<S>);(<t>,<T>);\
(<u>,<U>);(<v>,<V>);(<w>,<W>);(<x>,<X>);(<y>,<Y>);\
(<z>,<Z>)
#
END LC_CTYPE
```

LC_MESSAGES category:

The LC_MESSAGES category of a locale definition source file defines the format for affirmative and negative system responses. This category begins with an LC_MESSAGES category header and ends with an END LC_MESSAGES category trailer.

All operands for the LC_MESSAGES category are defined as strings or *extended regular expressions* enclosed by quotation marks ("").

Note: These operands are separated from the keyword they define by one or more blanks. Two adjacent quotation marks ("") indicate an undefined value. The following keywords are recognized in the LC_MESSAGES category:

yesexpr
Specifies an extended regular expression that describes the acceptable affirmative response to a question expecting an affirmative or negative response.

noexpr
Specifies an extended regular expression that describes the acceptable negative response to a question expecting an affirmative or negative response.

yesstr A fixed string of acceptable affirmative response.

nostr A fixed string of acceptable negative response.

Extended regular expressions

The following special characters are used to form extended regular expressions:

Character	Function
+	Specifies that a string matches if one or more occurrences of the character or extended regular expression that precedes the + (plus) are within the string.
?	Specifies that a string matches if zero or one occurrences of the character or extended regular expression that precedes the ? (question mark) are within the string.

Character	Function
|	Specifies that a string matches if either of the strings separated by the | (vertical line) are within the string.
( )	Groups strings together in regular expressions.
{m}	Specifies that a string matches if exactly m occurrences of the pattern are within the string.
{m,}	Specifies that a string matches if at least m occurrences of the pattern are within the string.
{m, n}	Specifies that a string matches if between m and n, inclusive, occurrences of the pattern are within the string (where m <= n).
[String]	Signifies that the regular expression matches any characters specified by the string variable within the square brackets.
[^ String]	A ^ (caret) within the [ ] (square brackets) and at the beginning of the specified string indicates that the regular expression does not match any characters within the square brackets.
^	Signifies the beginning of a field or record.
$	Signifies the end of a field or record.
. (period)	Signifies any one character except the terminal new-line character at the end of a space.
* (asterisk)	Signifies zero or more of any characters.
\ (backslash)	The escape character. When preceding any of the characters that have special meaning in extended regular expressions, the escape character removes any special meaning for the character.

Character class expressions may also be specified in the extended regular expression. The following character class expressions are supported in all locales:

[:alnum:]
[:alpha:]
[:blank:]
[:cntrl:]
[:digit:]
[:graph:]
[:lower:]
[:print:]
[:punct:]
[:space:]
[:upper:]
[:xdigit:]

Example

Here is an example of the LC_MESSAGES category in a locale definition source file.

```
LC_MESSAGES
#
yesexpr  "[yY]"
noexpr   "[nN]"
yesstr   "yes"
nostr    "no"
#
END LC_MESSAGES
```

LC_MONETARY category:

The LC_MONETARY category of a locale definition source file defines rules and symbols for formatting monetary numeric information. This category begins with an LC_MONETARY category header and ends with an END LC_MONETARY category trailer.

All operands for the LC_MONETARY category keywords are defined as string or integer values. String values are bounded by quotation marks (""). All values are separated from the keyword they define by one or more spaces. Two adjacent quotation marks indicate an undefined string value. A -1 indicates an undefined integer value. The following keywords are recognized in the LC_MONETARY category:

int_curr_symbol
: Specifies the string used for the international currency symbol. The operand for the int_curr_symbol keyword is a four-character string. The first three characters contain the alphabetic international-currency symbol. The fourth character specifies a character separator between the international currency symbol and a monetary quantity.

currency_symbol
: Specifies the string used for the local currency symbol.

mon_decimal_point
: Specifies the string used for the decimal delimiter used to format monetary quantities.

mon_thousands_sep
: Specifies the string used for grouping digits to the left of the decimal delimiter in formatted monetary quantities.

mon_grouping
: Defines the size of each group of digits in formatted monetary quantities. The operand for the mon_grouping keyword consists of a sequence of semicolon-separated integers. Each integer specifies the number of digits in a group. The initial integer defines the size of the group immediately to the left of the decimal delimiter. The following integers define succeeding groups to the left of the previous group. If the last digit is not -1, subsequent grouping is performed using the previous digit. If the last digit is -1, grouping is only performed for the number of groups specified.

  Here is an example of the interpretation of the mon_grouping keyword. Assuming that the value to be formatted is 123456789 and the operand for the mon_thousands_sep keyword is comma (,), the following results occur:

mon_grouping Value	Formatted Value
3;-1	123456,789
3	123,456,789
3;2	12,34,56,789
3;2;-1	1234,56,789

  positive_sign
  : Specifies the string used to indicate a nonnegative-valued formatted monetary quantity.

  negative_sign
  : Specifies the string used to indicate a negative-valued formatted monetary quantity.

  int_frac_digits
  : Specifies an integer value representing the number of fractional digits (those after the decimal delimiter) to be displayed in a formatted monetary quantity using the int_curr_symbol value.

frac_digits
Specifies an integer value representing the number of fractional digits (those after the decimal delimiter) to be displayed in a formatted monetary quantity using the currency_symbol value.

p_cs_precedes
Specifies an integer value indicating whether the int_curr_symbol or currency_symbol string precedes or follows the value for a non-negative formatted monetary quantity. The following integer values are recognized:

0 Indicates that the currency symbol follows the monetary quantity.

1 Indicates that the currency symbol precedes the monetary quantity.

p_sep_by_space
Specifies an integer value indicating whether the int_curr_symbol or currency_symbol string is separated by a space from a non-negative formatted monetary quantity. The following integer values are recognized:

0 Indicates that no space separates the currency symbol from the monetary quantity.

1 Indicates that a space separates the currency symbol from the monetary quantity.

2 Indicates that a space separates the currency symbol and the positive_sign string, if adjacent.

n_cs_precedes
Specifies an integer value indicating whether the int_curr_symbol or currency_symbol string precedes or follows the value for a negative formatted monetary quantity. The following integer values are recognized:

0 Indicates that the currency symbol follows the monetary quantity.

1 Indicates that the currency symbol precedes the monetary quantity.

n_sep_by_space
Specifies an integer value indicating whether the int_curr_symbol or currency_symbol string is separated by a space from a negative formatted monetary quantity. The following integer values are recognized:

0 Indicates that no space separates the currency symbol from the monetary quantity.

1 Indicates that a space separates the currency symbol from the monetary quantity.

2 Indicates that a space separates the currency symbol and the **negative_sign** string, if adjacent.

p_sign_posn
Specifies an integer value indicating the positioning of the positive_sign string for a non-negative formatted monetary quantity. The following integer values are recognized:

0 Indicates that parenthesis enclose both the monetary quantity and the int_curr_symbol or currency_symbol string.

1 Indicates that the positive_sign string precedes the quantity and the int_curr_symbol or currency_symbol string.

2 Indicates that the positive_sign string follows the quantity and the int_curr_symbol or currency_symbol string.

3 Indicates that the positive_sign string immediately precedes the int_curr_symbol or currency_symbol string.

4 Indicates that the positive_sign string immediately follows the int_curr_symbol or currency_symbol string.

n_sign_posn
Specifies an integer value indicating the positioning of the negative_sign string for a negative formatted monetary quantity. The following integer values are recognized:

0 Indicates that parenthesis enclose both the monetary quantity and the int_curr_symbol or currency_symbol string.

1 Indicates that the negative_sign string precedes the quantity and the int_curr_symbol or currency_symbol string.

2 Indicates that the negative_sign string follows the quantity and the int_curr_symbol or currency_symbol string.

3 Indicates that the negative_sign string immediately precedes the int_curr_symbol or currency_symbol string.

4 Indicates that the negative_sign string immediately follows the int_curr_symbol or currency_symbol string.

Example

Here is an example of the LC_MONETARY category listed in a locale definition source file.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
LC_MONETARY
#
int_curr_symbol    "<U><S><D>"
currency_symbol    "<dollar-sign>"
mon_decimal_point  "<period>"
mon_thousands_sep  "<comma>"
mon_grouping       3;-1
positive_sign      "<plus-sign>"
negative_sign      "<hyphen>"
int_frac_digits    2
frac_digits        2
p_cs_precedes      1
p_sep_by_space     2
n_cs_precedes      1
n_sep_by_space     2
p_sign_posn        3
n_sign_posn        3
#
END LC_MONETARY
```

Related concepts

"Example: Producing unique monetary formats" on page 230
A unique customized monetary format can be produced by changing the value of a single statement.

LC_NUMERIC category:

LC_NUMERIC category of a locale definition source file defines rules and symbols for formatting non-monetary numeric information.

The LC_NUMERIC category of a locale definition source file defines rules and symbols for formatting non-monetary numeric information. This category begins with an LC_NUMERIC category header and terminates with an END LC_NUMERIC category trailer.

All operands for the LC_NUMERIC category keywords are defined as string or integer values. String values are bounded by quotation marks (""). All values are separated from the keyword they define by one or more spaces. Two adjacent quotation marks indicate an undefined string value. A -1 indicates an undefined integer value. The following keywords are recognized in the LC_NUMERIC category:

decimal_point
Specifies a string containing the decimal delimiter character used to format numeric, non-monetary quantities.

thousands_sep
Specifies the string separator used for grouping digits to the left of the decimal delimiter in formatted numeric, non-monetary quantities.

grouping
Defines the size of each group of digits in formatted monetary quantities. The operand for the grouping keyword consists of a sequence of semicolon-separated integers. Each integer specifies the number of digits in a group. The initial integer defines the size of the group immediately to the left of the decimal delimiter. The following integers define succeeding groups to the left of the previous group. Grouping is performed for each integer specified for the grouping keyword. If the last digit is not -1, subsequent grouping is performed using the previous digit. If the last digit is -1, grouping is only performed for the number of groups specified.

Here is an example of the interpretation of the grouping statement. Assuming that the value to be formatted is 123456789 and the operand for the thousands_sep keyword is comma (,), the following results occur.

Grouping value	Formatted value
3	123,456,789
3;-1	123456,789
3;2	12,34,56,789
3;2;-1	1234,56,789

Example

Here is an example of the LC_NUMERIC category in a locale definition source file.

```
LC_NUMERIC
#
decimal_point "<period>"
thousands_sep "<comma>"
grouping      3;-1
#
END LC_NUMERIC
```

LC_TIME category:

The LC_TIME category of a locale definition source file defines rules and symbols for formatting time and date information. This category begins with an LC_TIME category header and ends with an END LC_TIME category trailer.

All operands for the LC_TIME category keywords are defined as string or integer values. String values are bounded by quotation marks (""). All values are separated from the keyword they define by one or more spaces. Two adjacent quotation marks indicate an undefined string value. A -1 indicates an undefined integer value. Field descriptors are used by commands and subroutines that query the LC_TIME category to represent elements of time and date formats. The following keywords are recognized in the LC_TIME category:

abday Defines the abbreviated weekday names corresponding to the %a field descriptor. Recognized values consist of seven semicolon-separated strings. The first string corresponds to the abbreviated name for the first day of the week (Sun), the second to the abbreviated name for the second day of the week, and so on.

day Defines the full spelling of the weekday names corresponding to the %A field descriptor. Recognized values consist of seven semicolon-separated strings. The first string corresponds to the full spelling of the name of the first day of the week (Sunday), the second to the name of the second day of the week, and so on.

abmon
Defines the abbreviated month names corresponding to the %b field descriptor. Recognized values consist of 12 semicolon-separated strings. The first string corresponds to the abbreviated name for the first month of the year (Jan), the second to the abbreviated name for the second month of the year, and so on.

mon Defines the full spelling of the month names corresponding to the %B field descriptor. Recognized values consist of 12 semicolon-separated strings. The first string corresponds to the full spelling of the name for the first month of the year (January), the second to the full spelling of the name for the second month of the year, and so on.

d_t_fmt
Defines the string used for the standard date and time format corresponding to the %c field descriptor. The string can contain any combination of characters, field descriptors, or escape sequences. See “Escape sequences” on page 217 for additional information.

d_fmt Defines the string used for the standard date format corresponding to the %x field descriptor. The string can contain any combination of characters, field descriptors, or escape sequences. Here is an example of how the d_fmt keyword can be constructed:

%D The %D indicates a %m/%d/%y date format. If you are using this format and have chosen to set the job attribute from the locale, then a slash (/) is extracted for the DATSEP job attribute. *MDY is extracted for the DATFMT job attribute.

%j The %j indicates a Julian date format. If you are using this format and have chosen to set the job attribute from the locale, then no DATSEP job is extracted. However, *JUL is extracted for the DATFMT job attribute.

%d-%m-%y
If you are using this format and have chosen to set the job attribute from the locale, then the compiler extracts the hyphen (-) for the DATSEP job attribute and *DMY for the DATFMT job attribute.

%y.%m.%d
If you are using this format and have chosen to set the job attribute from the locale, then the compiler extracts the period (.) for the DATSEP job attribute and *YMD for the DATFMT job attribute.

%m/%d/%Y
If you are using this format and have chosen to set the job attribute from the locale, then the compiler extracts the slash (/) for the DATSEP job attribute. No DATFMT job attribute is extracted.

Note: If the locale is to contain a valid i5/OS date format and date separator, then the d_fmt value must be defined such that it contains valid i5/OS date format and date separators. For example, if the value is specified as *%m/%d/%y*, then *MDY will be extracted for the i5/OS date format and a slash (/) will be extracted for the i5/OS date format. A warning is issued by the CRTLOCALE command if an i5/OS date format or date separator cannot be extracted.

t_fmt Defines the string used for the standard time format corresponding to the %X field descriptor.

The string can contain any combination of characters, field descriptors, or escape sequences. Here is an example of how the t_fmt keyword can be constructed:

%H:%M:%S
The compiler extracts a colon (:) for the TIMSEP job attribute.

%H.%M.%S
The compiler extracts a period (.) for the TIMSEP job attribute.

%H %M %S
The compiler extracts a blank space for the TIMSEP job attribute.

%H,%M,%S
The compiler extracts a comma (,) for the TIMSEP job attribute.

%T %T implies a %H:%M:%S (hours, minutes, seconds) time format with a colon (:) as the TIMSEP job attribute.

%H&%M&%S;
A valid TIMSEP job attribute cannot be determined.

Note: If the locale is to contain a valid i5/OS time separator, then the t_fmt value must be defined such that it contains a valid i5/OS time separator. For example, if the value is specified as *%H:%M:%S*, then a : (colon) will be extracted for the i5/OS date format. A warning is issued by the CRTLOCALE command if an i5/OS time separator cannot be extracted.

am_pm
Defines the strings used to represent *ante meridiem* (before noon) and *post meridiem* (after noon) corresponding to the %p field descriptor. Recognized values consist of two strings separated by a ; (semicolon). The first string corresponds to the *ante meridiem* designation, the last string to the *post meridiem* designation.

t_fmt_ampm
Defines the string used for the standard 12-hour time format that includes an am_pm value (%p field descriptor). This statement corresponds to the %r field descriptor. The string can contain any combination of characters and field descriptors.

era Defines how the years are counted and displayed for each era in a locale, corresponding to the %E field descriptor modifier. For each era, there must be one string in the following format:

```
direction:offset:start_date:end_date:era_name:era_format
```

The variables for the era-string format are defined as follows:

direction
Specifies a minus sign (-) or plus sign (+) character. The plus character indicates that years count in the positive direction when moving from the start date to the end date. The minus character indicates that years count in the negative direction when moving from the start date to the end date.

offset Specifies a number representing the first year of the era.

start_date
Specifies the starting date of the era in the *yyyy/mm/dd* format, where *yyyy* is the year, *mm* is the month, and *dd* is the day. Years before the year 1 AD are represented as negative numbers. For example, an era beginning March 5th in the year 100 BC is represented as -100/03/05.

end_date
Specifies the ending date of the era in the same form used for the *start_date* variable or one of the two special values -* or +*. A -* value indicates that the ending date of the era extends backward to the beginning of time. A +* value indicates that the ending date of

the era extends forward to the end of time. Therefore, the ending date can be chronologically before or after the starting date of the era. For example, the strings for the Christian eras AD and BC are entered as follows:

```
+:0:0000/01/01:+*:AD:%o %N
+:1:-0001/12/31:-*:BC:%o %N
```

era_name
Specifies a string representing the name of the era that is substituted for the %EC field descriptor.

era_format
Specifies a string for formatting the %EY field descriptor.

An **era** value consists of one string for each era. If more than one era is specified, each era string is separated by a ; (semicolon).

era_d_fmt
Defines the string used to represent the date in alternate-era format corresponding to the %Ex field descriptor. The string can contain any combination of characters and field descriptors.

era_t_fmt
Defines the string used to represent the time in alternate-era format corresponding to the %EX field descriptor. The string can contain any combination of characters and field descriptors.

era_d_t_fmt
Defines the string used to represent the date and time in alternate-era format corresponding to the %Ec field descriptor. The string can contain any combination of characters and field descriptors.

alt_digits
Defines alternate strings for digits corresponding to the %O field descriptor. Recognized values consist of a group of strings separated by ; (semicolons). The first string represents the alternate string for zero, the second string represents the alternate string for one, and so on. A maximum of 100 alternate strings can be specified.

Escape sequences

Here are escape sequences allowed for the d_t_fmt, d_fmt, and t_fmt keyword values:

>

\\	Represents the backslash character.
\a	Represents the alert character.
\b	Represents the backspace character.
\f	Represents the form-feed character.
\n	Represents the newline character.
\r	Represents the carriage-return character.
\t	Represents the tab character.
\v	Represents the vertical-tab character.

Example

Here is an example of the LC_TIME category in a locale definition source file.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
LC_TIME
#
#Abbreviated weekday names (%a)
```

```
abday
"<S><u><n>";"<M><o><n>";"<T><u><e>";"<W><e><d>";\
        "<T><h><u>";"<F><r><i>";"<S><a><t>"
#
#Full weekday names (%A)
day "<S><u><n><d><a><y>";"<M><o><n><d><a><y>";\
"<T><u><e><s><d><a><y>";"<W><e><d><n><e><s><d><a><y>";\
"<T><h><u><r><s><d><a><y>";"<F><r><i><d><a><y>";\
"<S><a><t><u><r><d><a><y>"
#
#Abbreviated month names (%b)
abmon  "<J><a><n>";"<F><e><b>";"<M><a><r>";"<A><p><r>";\
       "<M><a><y>";"<J><u><n>";"<J><u><l>";"<A><u><g>";\
       "<S><e><p>";"<O><c><t>";"<N><o><v>";"<D><e><c>"
#
#Full month names (%B)
mon "<J><a><n><u><a><r><y>";"<F><e><b><r><u><a><r><y>";\
"<M><a><r><c><h>";"<A><p><r><i><l>";"<M><a><y>";\
"<J><u><n><e>";"<J><u><l><y>";"<A><u><g><u><s><t>";\
"<S><e><p><t><e><m><b><e><r>";"<O><c><t><o><b><e><r>";\
"<N><o><v><e><m><b><e><r>";"<D><e><c><e><m><b><e><r>"
#
#Date and time format (%c)
d_t_fmt "%a %b %d %H:%M:%S %Y"
#
#Date format (%x)
d_fmt       "%m/%d/%y"
#
#Time format (%X)
t_fmt       "%H:%M:%S"
#
#Equivalent of AM/PM (%p)
am_pm       "<A><M>";"<P><M>"
#
#12-hour time format (%r)
t_fmt_ampm  "%I:%M:%S %p"
#
era      "+:0:0000/01/01:+*:AD:%EC";\
"+:1:-0001/12/31:-*:BC:%Ey";
era_d_fmt  ""
alt_digits
"<0><t><h>";"<1><s><t>";"<2><n><d>";"<3><r><d>";\
"<4><t><h>";"<5><t><h>";"<6><t><h>";"<7><t><h>";\
"<8><t><h>";"<9><t><h>";"<1><0><t><h>"
#
END LC_TIME
```

Related concepts

"Example: Creating and enabling a locale" on page 225
This example contains the steps necessary for creating and enabling a locale.

LC_TOD category:

The LC_TOD category defines the rules used to define the start and end time of daylight saving time (DST), the difference between local time and Greenwich Mean time, the time zone name, and the DST name. This category is an IBM extension and must appear after all other category definitions in the source file.

Notes:

1. System locale files (such as /QSYS.LIB/EN_US.LOCALE) that IBM provides do not have LC_TOD information filled in. This is because the same locale source (for example, EN_US) can be used for many different time zones and, therefore, no sets of values can be used as the default for every time zone.

2. The values specified in the LC_TOD category are only reflected in the C functions that process time. These values do not affect a job's time, which is set by a time zone object.

The LC_TOD locale time definitions are used only by applications that use locale-sensitive APIs, which are often application code that is based on C and C++ programming languages. Traditional system code (for example, DSPJOB) does not use the locale LC_TOD information.

RPG and COBOL applications have several ways if they need to use the LC_TOD information in a program:
- They can use the locale-sensitive APIs found in C run time.
- They can use the QlgRetrieveLocaleInformation API to retrieve locale information and do their own date and time formatting independent of the C runtime support.
- They can use system APIs, such as Convert Date and Time Format (QWCCVTDT), to map system time to local time. The QWCCVTDT API supports the specification of a time zone object when performing date and time conversions.

All the operands for the LC_TOD category are defined as string or integer values. String values are bounded by quotation marks (""). All values are separated from the keyword they define by one or more spaces. Two adjacent quotation marks indicate an undefined string value. A 0 (zero) indicates an undefined integer value. The following keywords are recognized in the LC_TOD category.

tzdiff The time zone difference in minutes. It is the difference between the local time and Greenwich mean time.

tname The string used for the time zone name.

dstname
The string used for the DST name.

dststart
A set of 4 integers that represents the DST start date. The operand for the dststart keyword consists of a sequence of four integers separated by comma.

dststart keyword format

month,*week*,*day*,*time*

The variables for the dststart format are defined as follows:

month An integer value that represents the month of the year when DST takes effect. This value ranges from 1 to 12, with 1 corresponding to January, and 12 corresponding to December.

week An integer value that represents the week of the month when DST takes effect. This value ranges from -4 to 4, with -4 corresponding to the fourth week of the month counting from the end of the month and 4 corresponding to the fourth week of the month counting from the beginning of the month.

Note: You can specify zero for *week*. This means that the day indicates the day of the month rather than the day of the week. For example, the string 4,0,23,0 means that DST starts the fourth month of the year, the twenty-third day of the month, and 0 seconds after midnight local standard time.

day An integer value that represents the day of the month when DST takes effect. If the week keyword is not 0 (zero), then this is the day of the week when DST takes effect. This value ranges from 1 to the last day of the month or 1 to the last day of the week.

time An integer value that represents the number of seconds after 12 midnight, local standard time, when DST takes effect. This value ranges from 0 to 86399.

dstend

A set of 4 integers that represents the DST end date. The operand for the dstend keyword consists of a sequence of four integers that are separated by comma.

dstend keyword format

month,*week*,*day*,*time*

The variables for the dstend format are defined as follows:

month An integer value that represents the month of the year when DST ends. This value ranges from 1 to 12, with 1 corresponding to January, and 12 corresponding to December.

week An integer value that represents the week of the month when DST ends. This value ranges from -4 to 4, with -4 corresponding to the fourth week of the month counting from the end of the month and 4 corresponding to the fourth week of the month counting from the beginning of the month.

day An integer value that represents the day of the month when DST ends. If the week keyword is not 0 (zero), then this is the day of the week when DST ends. This value ranges from 1 to the last day of the month or 1 to the last day of the week.

time An integer value that represents the number of seconds after 12 midnight, local standard time, when DST takes effect. This value ranges from 0 to 86399.

dstshift

An integer value that represents the DST shift value in seconds.

Example

Here is an example of the LC_TOD category in a locale definition source file.

Note: By using the code examples, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
LC_TOD
#
tzdiff    360
tname     "<C><S><T>"
dstname   "<C><D><T>"

#Set daylight saving time to start on 3rd week of October at
#midnight on Saturday.
dststart  10,3,6,0

#Set daylight saving time to end on April 23, at midnight.

dstend    4,0,23,0
dstshift  3600
#
END LC_TOD
```

Locale symbolic names

The i5/OS operating system supports locale symbolic names based on predefined names from the X/Open Standard portable character set.

In addition, the i5/OS operating system supports a 5-character alphanumeric symbolic name for all characters, in the following situations:

- The first character of the symbolic name is a Latin capital letter U. This character identifies that the name is derived from the ISO/IEC 10646 Universal Coded Character Set.

- The second through fifth characters of the symbolic name represent the code point of the character in the ISO/IEC 10646 Universal Coded Character Set 2 Level 1. This portion of the symbolic name is assigned by code point for ease of creating and changing locales.

As an example, the question mark (?) character provides the following correlation between symbolic naming, UCS2-1 code point, and an IBM-assigned code point:

- The ? character is symbolically represented by `<question-mark>`
- It is at code point `U003F` in the ISO 10646 code page
- It is at code point `6F` in IBM code page 500.

Mapping of locale symbolic names provides a list of all symbolic names supported on i5/OS. The table also provides the UCS2-1 (ISO 10646) code points, their corresponding IBM code page or code points, and a graphic representation of each character.

Related concepts

"Portable character set" on page 328
The X/Open portable character set is a superset of the IBM invariant character set (00640), which includes symbols not represented in the IBM invariant character set 00640.

"Mapping of locale symbolic names" on page 365
This table lists locale symbolic names.

Examples: Locale programming

These examples illustrate how locales work in the i5/OS operating system, and how you can work with locales and use locales for a multilingual system environment.

In addition to these examples, Locale categories provides programming examples for each of the different locale categories.

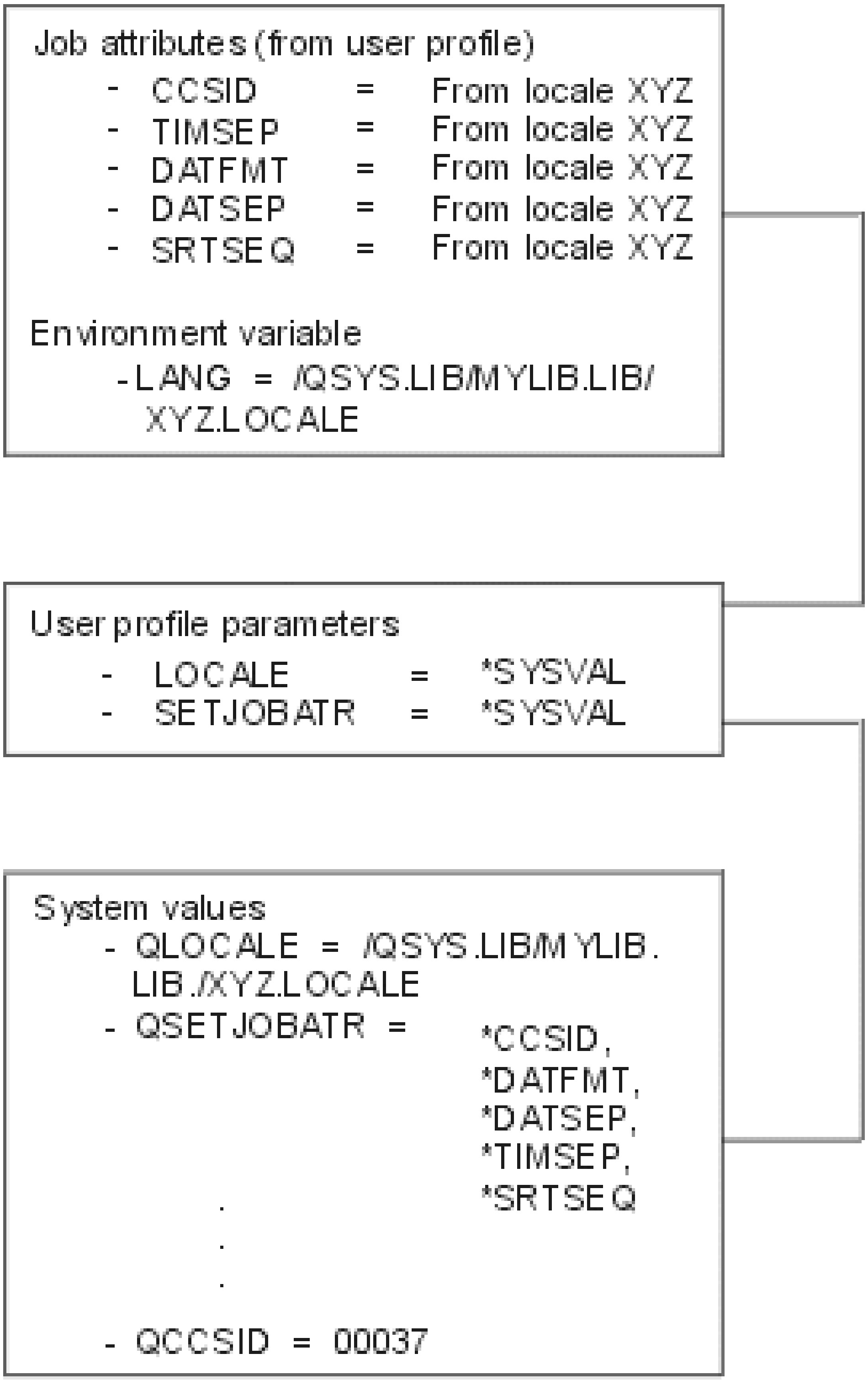
Related concepts

"Locale categories" on page 201
This table describes the locale categories that are supported on the i5/OS operating system.

Example: How locales work:

These two examples focus on the LOCALE and SETJOBATR parameter values specified on the user profile.

The first example illustrates using locales to establish job attributes. The user profile parameters LOCALE and SETJOBATR have values of *SYSVAL. This means that the job attributes at job start time come from the QLOCALE value based on the values in QSETJOBATR.



If a job runs based on the information in the figure, the following conditions are true:

- The locale used is XYZ.

  This is because the user profile parameter value for LOCALE was *SYSVAL. The *SYSVAL value is XYZ.
- The CCSID is based on the value specified when locale XYZ was created.

This value is specified when the LOCALE object is created using the CRTLOCALE command.

- The time separator is derived from locale XYZ.

  This value is derived from the LC_TIME category specified in LOCALE XYZ.
- The date format separator is derived from locale XYZ.

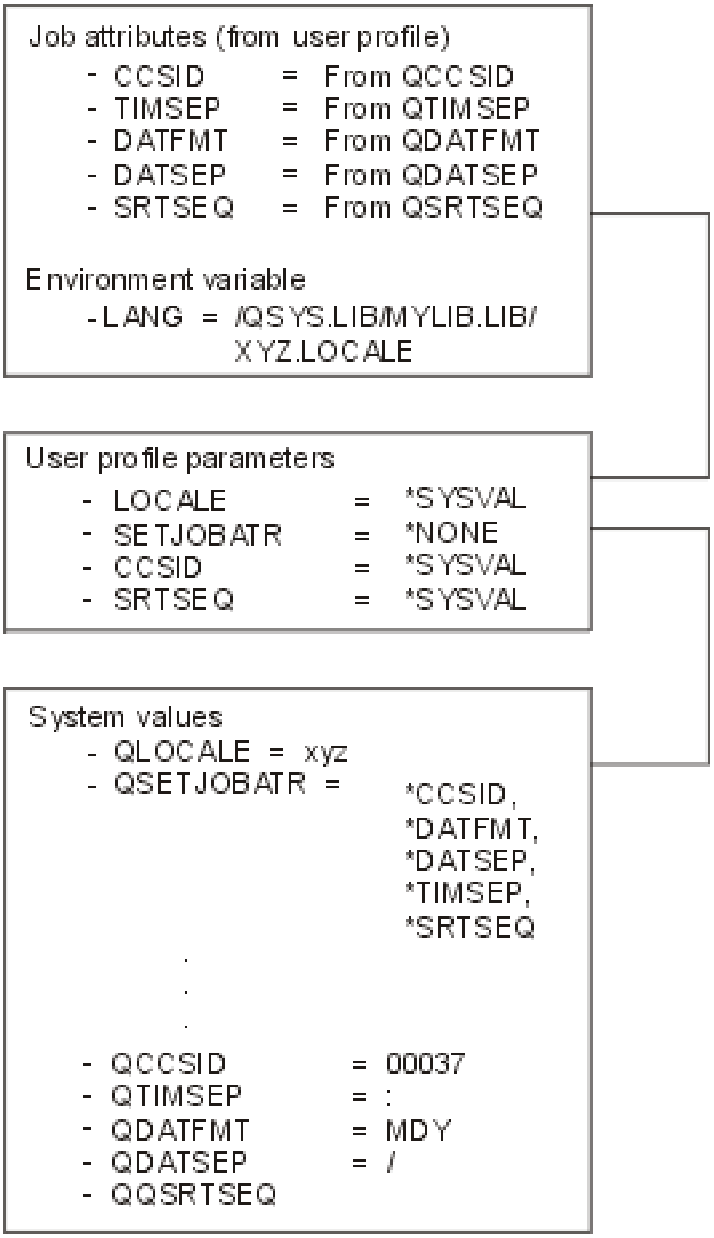
  This value is derived from the LC_TIME category specified in LOCALE XYZ.
- The data separator is derived from locale XYZ.

  This value is derived from the LC_TIME category specified in LOCALE XYZ.
- The decimal format character is derived from locale XYZ.

  This value is derived from the LC_NUMERIC category specified in LOCALE XYZ.

In the second example the user profile LOCALE parameter value is *SYSVAL and the SETJOBATR parameter values is *NONE. This means that the LOCALE value is determined by looking at the system value QLOCALE. When the SETJOBATR value is *NONE, job attributes are determined by the values in the user profile.

Remember, because the user profile SETJOBATR parameter was *NONE, the system's search resulted in using the values specified for QCCSID, QTIMSEP, QDATFMT, QSRTSEQ, and QDATSEP.



If a job runs based on the information in this example, the following conditions are true:

- The locale used is XYZ.

  This is because the user profile parameter value for LOCALE was *SYSVAL. The *SYSVAL value is XYZ.
- The CCSID is 00037.

  This is because the user profile SETJOBATR parameter value was *NONE. The system search ended with the value for QCCSID being used.
- The time separator is a colon (:).

  This is because the user profile SETJOBATR parameter value was *NONE. The system search ended with the value for QTIMSEP being used.
- The date format separator is a slash (/).

  This is because the user profile SETJOBATR parameter value was *NONE. The system search ended with the value for QDATSEP being used.
- The date format is month/day/year (MDY).

This is because the user profile SETJOBATR parameter value was *NONE. The system search ended with the value for QDATFMT being used.

- The decimal format character is a period. Zero suppression is performed.

  This is because the user profile SETJOBATR parameter value was *NONE. The system search ended with the value for QDECFMT being used.

Example: Creating and enabling a locale:

This example contains the steps necessary for creating and enabling a locale.

To create and enable a locale, follow these steps:

1. Create (or have) a library and a source physical file.
2. Copy an existing locale source file definition member (to a library and source physical file).
3. Edit the copied locale source file member if you need to customize any of the categories within the locale source.
4. Create the locale object.
5. Enable the locale object by using system values or parameters on the user profile.

Note: By using the code examples, you agree to the terms of the “Code license and disclaimer information” on page 445.

Step 1. Creating a library and a source physical file

A library and a source physical file are needed to store the locale source file member. Locale-source-definition files that are system supplied include a list of the locale source file members that are included in the i5/OS operating system.

To create a library called localelib, follow these steps:

1. On a command line, type CRTLIB and press F4 (Prompt).
2. Type `localelib` for the name of the library and press Enter.

To create a source physical file (localesrc) in library localelib, follow these steps:

1. On a command line, type CRTSRCPF and press F4 (Prompt).
2. Type `localesrc` for the file name and press Enter.

Step 2. Copying an existing locale source definition

Locale-source-definition file members that are IBM supplied are located in library QSYSLOCALE, source physical file QLOCALESRC. See System-supplied locale source definition files for a list of the IBM-supplied locale source files.

IBM provides sources for many sample locales. Besides the locales that IBM provides, an open source repository that contains locale sources is available. See the POSIX type files at the Unicode Common Locale Data Repository (CLDR) Web site for more information.

In this example, member EN_US, a locale for the English language, is copied.

To copy a locale source definition, follow these steps:

1. On a command line, type CPYF and press F4 (Prompt).
2. Type the values shown on the following display.

```
Copy File (CPYF)

Type choices, press Enter.

From file  . . . . . . . . . . .   QLOCALESRC    Name
Library  . . . . . . . . . . .     QSYSLOCALE  Name, *LIBL,*CURLIB
To file  . . . . . . . . . . . .   localesrc_    Name, *PRINT
Library  . . . . . . . . . . .     localelib_  Name, *LIBL,*CURLIB
From member  . . . . . . . . . .   EN_US         Name, generic*,*FIRST, *ALL
To member or label . . . . . . .   EN_US_____    Name, *FIRST,*FROMMBR
Replace or add records . . . . .   *ADD____      *NONE, *ADD,*REPLACE
Create file  . . . . . . . . . .   *YES          *NO, *YES
Record format field mapping  . .   *MAP__        *NONE, *NOCHK,*CVTSRC
```

The values entered copy the EN_US member to the source physical file localesrc in library localelib.

Note: When you copy a file that is tagged with a CCSID, you need to use the FMTOPT(*MAP) parameter to ensure that the copied source is converted to the CCSID of the to-file. Scroll ahead to see the FMTOPT parameter.

Step 3. Editing the copied locale source definition

If you want to use the IBM-supplied locale as it is included, you do not need to change it. You can continue with step 4, Creating the locale object.

In this example, edit the EN_US member to set the time-of-day keywords used in the LC_TOD category. To know more about the LC_TOD category, see “LC_TOD category” on page 218.

This example shows how to use Source Entry Utility (SEU) to edit the locale. You can also use an equivalent editor.

To edit the locale using SEU, follow these steps:

1. On a command line, type STRSEU (Start Source Entry Utility) and press F4 (Prompt).
2. Type the source file name (localesrc), library name (localelib), and source member name (EN_US) as shown on the following display.

```
Start Source Entry Utility (STRSEU)

Type choices, press Enter.

Source file  . . . . . . . . . .   localesrc     Name, *PRV
Library  . . . . . . . . . . .     localelib_  Name, *LIBL, *CURLIB, *PRV
Source member  . . . . . . . . .   EN_US_____    Name, *PRV, *SELECT
```

3. Press Enter. The following display appears.

```
COLUMNS . . . :    1  71           EDIT      localelib/localesrc
SEU==>  F_LC_TOD      EN_US
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+...6 ...+... 7
*************** BEGINNING OF DATA************************************
5967.00 comment_char <percent-sign>
5968.00 escape_char  <slash>
5969.00
5970.00 %
5971.00 % 5761SS1         (C) COPYRIGHT IBM CORP. 1991,2008
5972.00 % ALL RIGHTS RESERVED.
5973.00 % US GOVERNMENT USERS RESTRICTED RIGHTS -
5974.00 % USE, DUPLICATION OR DISCLOSURE RESTRICTED
5975.00 % BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
5976.00 %
5977.00 % LICENSED MATERIALS-PROPERTY OF IBM
5978.00 %
5979.00 % FILE NAME  :    EN_US
5980.00 %
5981.00 % COUNTRY/REGION: UNITED STATES
5982.00 %

F3=EXIT   F4=PROMPT   F5=REFRESH   F9=RETRIEVE   F10=CURSOR   F11=TOGGLE
F16=REPEAT FIND       F17=REPEAT CHANGE          F24=MORE KEYS
                                (C) COPYRIGHT IBM CORP.1991, 2008.
```

4. Use the SEU search function to locate LC_TOD. After the search is completed, the following display appears.

   As you can see, all LC_TOD category keywords have values of 0 and no descriptive names are declared for tname and dstname.

```
COLUMNS . . . :    1  71           EDIT      localelib/localesrc
SEU==>       EN_US
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+...6 ...+... 7
6519.00
6520.00 LC_TOD
6521.00
6522.00 tzdiff   0
6523.00 tname    ""
6524.00 dstname  ""
6525.00 dststart 0,0,0,0
6526.00 dstend   0,0,0,0
6527.00 dstshift 0
6528.00
6529.00 END LC_TOD
****************** END OF DATA****************************************






F3=EXIT   F4=PROMPT   F5=REFRESH   F9=RETRIEVE   F10=CURSOR  F11=TOGGLE
F16=REPEAT FIND       F17=REPEAT CHANGE          F24=MORE KEYS
STRING LC_TOD FOUND.
```

5. To create the correct LC_TOD values for CST and CDT in the locale (for 2007 and later years), type the following values for the LC_TOD keywords:

tzdiff Type 360.

This is the difference in the number of minutes between Greenwich meantime and the central time zone of the United States.

tname Type CST.

dstname
Type CDT.

dststart

Type 3,2,1,7200.

This string of integers means that daylight saving time starts the third month of the year, the beginning of the second week, the first day of the week, and 7200 seconds (120 minutes) after midnight local standard time.

dstend

Type 10,3,6,0.

This string of integers means that daylight saving time ends the tenth month of the year, the beginning of the third week, the sixth day of the week, and 0 seconds after midnight local standard time.

dstshift

Type 3600.

For detailed information about the LC_TOD keywords, see "LC_TOD category" on page 218.

The SEU edit screen should match the following display.

6. Press Enter to make the changes to the EN_US locale member.

```
COLUMNS . . . :    1  71           EDIT      localelib/localesrc
SEU==>       EN_US
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+...6 ...+... 7
6519.00
6520.00 LC_TOD
6521.00
6522.00 tzdiff   360
6523.00 tname   "CST"
6524.00 dstname  "CDT"
6525.00 dststart 3,2,1,7200
6526.00 dstend   11,1,1,7200
6527.00 dstshift 3600
6528.00
6529.00 END LC_TOD
****************** END OF DATA ****************************************







F3=EXIT   F4=PROMPT   F5=REFRESH   F9=RETRIEVE   F10=CURSOR   F11=TOGGLE
F16=REPEAT FIND      F17=REPEAT CHANGE          F24=MORE KEYS
STRING LC_TOD FOUND.
```

7. Save the member and exit SEU.

You have now copied the IBM-supplied locale source member and customized the LC_TOD category.

Step 4. Creating the locale object

To create the locale object, follow these steps:

1. On a command line, type CRTLOCALE and press F4 (Prompt). The Create Locale (CRTLOCALE) display appears.
2. In the Locale name field, type the locale path name (the path name includes the location and the name of the locale source member).
3. In the Source file path name field, type the source file path name (the location and name of the source physical file and the member name).

   Notes:

   a. Make sure that the locale source file you are using has the same characters defined as does the CCSID you specify on the CRTLOCALE command. If they do not have the same characters available, unpredictable results can occur. For example, the EN_US source file has lowercase letters defined. However, the lowercase letters are not available in CCSID 290.

b. When you create locale objects, it is preferred to make the CCSID part of the locale object name. For example, EN_US created with CCSID 37 can be named EN_US37.

```
Create Locale (CRTLOCALE)

Type choices, press Enter.

Locale name  . . . . . . . . . . > '/qsys.lib/localelib.lib/en_us.locale'____
______
Source file path name  . . . . . > '/qsys.lib/localelib.lib/localesrc.file/en
_us.mbr'
Coded character set ID . . . . . > 37________    1-65533, *JOB
Generation severity level  . . .   10____        10, 20
Text 'description' . . . . . . .   my version of locale EN_US - contains my c
hanges_








Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters  F12=Cancel
F13=How to use this display        F24=More keys
```

4. Press Enter to complete the creation of the locale object named EN_US in the library LOCALELIB.

Step 5. Enabling the locale object

You can enable locales system-wide by using the QLOCALE system value or for individual users by changing their user profile. To enable the locale system-wide, make EN_US the value for the QLOCALE system value. In this example, the locale support is enabled for one user.

To enable the locale for one user, follow these steps:

1. On a command line, type CHGUSRPRF and press F4 (Prompt).
2. Specify the user ID of the user and press Enter.

In the following portion of the Change User Profile display, the LOCALE parameter now has a value indicating that EN_US is the specified locale to be used by the user ID.

```
For more values
Locale . . . . . . . . . . . . .  QSYS.LIB/LOCALELIB.LIB/EN_US.LOCALE______
_______
```

After the user profile has been changed, any jobs initiated by the user ID have the EN_US locale associated with those jobs. The LANG environment variable is also initialized to the name of the locale.

Related concepts

"Working with locales" on page 200
Locales are used primarily in ILE-based application programs. Additionally, the Retrieve Locale Information (OPM, QLGRTVLC; ILE, QlgRetrieveLocaleInformation) API retrieves one or all categories of a locale.

"System-supplied locales and recommended CCSIDs" on page 360
The system-supplied locale source definition file members are in the optionally installable library QSYSLOCALE in the QLOCALESRC source file. The source file members are encoded in CCSID 37 and are read only.

"LC_TIME category" on page 214
The LC_TIME category of a locale definition source file defines rules and symbols for formatting time and date information. This category begins with an LC_TIME category header and ends with an END LC_TIME category trailer.

Example: Producing unique monetary formats:

A unique customized monetary format can be produced by changing the value of a single statement.

For example, the following table shows the results of using all combinations of defined values for the p_cs_precedes, p_sep_by_space, and p_sign_posn statements:

p_cs_precedes value	p_sign_posn value	p_sep_by_space=2	p_sep_by_space=3	p_sep_by_space=4
p_cs_precedes = 1	p_sign_posn = 0	($1.25)	($ 1.25)	($1.25)
	p_sign_posn = 1	+ $1.25	+$ 1.25	+$1.25
	p_sign_posn = 2	$1.25 +	$ 1.25+	$1.25+
	p_sign_posn = 3	+ $1.25	+$ 1.25	+$1.25
	p_sign_posn = 4	$ +1.25	$+ 1.25	$+1.25
p_cs_precedes = 0	p_sign_posn = 0	(1.25 $)	(1.25 $)	(1.25$)
	p_sign_posn = 1	+1.25 $	+1.25 $	+1.25$
	p_sign_posn = 2	1.25$ +	1.25 $+	1.25$+
	p_sign_posn = 3	1.25+ $	1.25 +$	1.25+$
	p_sign_posn = 4	1.25$ +	1.25 $+	1.25$+

Related concepts

"LC_MONETARY category" on page 210
The LC_MONETARY category of a locale definition source file defines rules and symbols for formatting monetary numeric information. This category begins with an LC_MONETARY category header and ends with an END LC_MONETARY category trailer.

Example: Locales as part of a multilingual environment:

The i5/OS operating system, through the use of locales, user profiles, and subsystems, can provide a multilingual environment. Users of a system that is set up for multilingual environments can work with their national language and all of its cultural conventions (for example, the character used to separate hours, minutes, and seconds).

Assume that the system used in this example has its primary language defined as English and the secondary national language versions (NLVs) for French and Spanish have been installed.

Follow the steps in this example to perform the following tasks:
- Create the locales for English, French, and Spanish.
- Create user profiles for users named: English, French, and Spanish.
- Create separate subsystems for French and Spanish language users.

Step 1. Creating locales
1. Type CRTLOCALE and press the Prompt key (F4).
2. Enter these values for the following fields:
   - Locale name: qsys.lib/localelib.lib/en_us.locale
   - Source file path name: qsys.lib/qsyslocale.lib/qlocalesrc.file/en_us.mbr
   - Coded character set ID: 37

- Generation severity level: 20
  - Text ’description’: US English locale
3. Press Enter.

Repeat the CRTLOCALE command for the FRENCH and SPANISH locales, using these values for the following fields.

For the French locale:
- Locale name: qsys.lib/localelib.lib/fr_fr.locale
- Source file path name: qsys.lib/qsyslocale.lib/qlocalesrc.file/fr_fr.mbr
- Coded character set ID: 297
- Generation severity level: 20
- Text ’description’: French locale

For the Spanish locale:
- Locale name: qsys.lib/localelib.lib/es_es.locale
- Source file path name: qsys.lib/qsyslocale.lib/qlocalesrc.file/es_es.mbr
- Coded character set ID: 284
- Generation severity level: 20
- Text ’description’: Spanish locale

You have created three locales (EN_US (English US), FR_FR (French), and ES_ES (Spanish). They are stored in library localelib.lib.

Step 2. Creating the user profile

In this example, three user profiles are created; each one will use one of the locales that was just created. The user profile names are English, French, and Spanish.
1. Type CRTUSRPRF and prompt (F4).
2. Type ENGLISH for the User profile parameter value.
3. Scroll forward until you see the Locale job attributes parameter and the Locale parameter.
4. Type:
   - /qsys.lib/localelib.lib/en_us.locale for the Locale parameter value.
   - Type + for the Locale job attributes parameter value and press Enter. Type:
     ```
     *CCSID
     *DATFMT
     *DATSEP
     *TIMSEP
     *SRTSEQ
     *DECFMT
     ```

   Note: At job start, the system finds the actual job attribute values defined in the locale object. The job attributes found in the locale override the values specified in the user profile fields for the CCSID and SRTSEQ parameters. They also override the date format, date separator, and time separator job attributes specified in any system value.
5. Press Enter. You have now created the user profile for a user named ENGLISH.

Repeat the CRTUSRPRF command for user IDs FRENCH and SPANISH. The next two displays provide the correct Locale parameter and Locale job attribute information for creating the user profiles for FRENCH and SPANISH.

```
Locale job attributes  . . . . . > *CCSID        *SYSVAL,*NONE, *CCSID...
> *DATFMT
> *DATSEP
> *TIMSEP
> *SRTSEQ
Locale . . . . . . . . . . . . . >'/qsys.lib/localelib.lib/fr_fr.locale'
```

```
Locale job attributes  . . . . . > *CCSID        *SYSVAL,*NONE, *CCSID...
> *DATFMT
> *DATSEP
> *TIMSEP
> *SRTSEQ
Locale . . . . . . . . . . . . . >'/qsys.lib/localelib.lib/es_es.locale'
```

Step 3. Creating subsystems for each national language version

Subsystems can be tailored to provide users an environment in which they see their own national language with data presented in the cultural format and conventions they are used to seeing.

Note: Because the primary language of the system is English, it is not necessary to create a subsystem for English.

1. Type CRTSBSD and prompt (F4).
2. Specify values for the following parameters to ensure that the subsystem is enabled for a specific national language (such as French and Spanish in our example).
   - Subsystem description

     This can be any name you choose.
   - Text 'description'

     The description can be anything you want it to be.
   - Sign-on display file and Library

     This often is QDSIGNON. The important information here is to know the name of the library where the national language version (French in this example) is stored.
   - Subsystem library

     It specifies a library that is entered ahead of other libraries in the library list of jobs started in this subsystem. This parameter allows you to use a secondary language library causing messages and displays to appear in your spoken language.

     Note: The correct values for Sign-on display file library and Subsystem library parameters are determined by adding QSYS to the national language version feature code. For example, the French national language library is named QSYS2928.

   The following screen shows the correct values to ensure that users of the FRENCH subsystem interact with the computer in the French language.

   ```
   Create Subsystem Description (CRTSBSD)

   Type choices, press Enter.

   Subsystem description  . . . . . SBSD         > FRENCH
   Library  . . . . . . . . . . .                  *CURLIB
   ```

```
Text 'description' . . . . . . . TEXT        > 'Subsystem for French users'

Additional Parameters

Sign-on display file . . . . . . SGNDSPF      > QDSIGNON
Library  . . . . . . . . . . .              >   QSYS2928
Subsystem library  . . . . . . . SYSLIBLE     > QSYS2928
More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

3. Press Enter.

Step 4. More information about subsystems

Creation of a subsystem requires additional work such as:
- Setting subsystem attributes
- Adding workstation entries
- Adding job queue entries
- Adding communications entries (if your national language users are attached over communications lines)
- Adding autostart job entries if you want to use this feature
- Adding prestart job entries if you want to use this feature
- Creating a class
- Adding routing entries

How to perform the tasks in the previous list is not described in this example.

Related concepts

"National language version feature codes" on page 247
This table lists the available national language version feature codes on the i5/OS operating system. When you order an i5/OS licensed program, you identify the national language version you want by specifying a language feature code.

Work management

Example: POSIX locale:

This example shows the POSIX (or C) locale categories and the source.

It is published in its entirety because:
- It provides a locale example with source provided for all categories.
- If you have not set a locale value in your C application program, the default POSIX locale is then used.

In either case, in the following listing, you are able to look at the locale categories and view the source.

Note: By using the code example, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
comment_char <percent-sign>
escape_char  <slash>


%
% 5716SS1 (C) COPYRIGHT IBM(R) CORP. 1991,1996
% ALL RIGHTS RESERVED.
% US GOVERNMENT USERS RESTRICTED RIGHTS -
% USE, DUPLICATION OR DISCLOSURE RESTRICTED
```

```
% BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
%
% LICENSED MATERIALS-PROPERTY OF IBM
%
% FILE NAME : POSIX
%
% COUNTRY/REGION: POSIX DEFAULT LOCALE
%
% LANGUAGES(S): NOT SPECIFIED
%
% DESCRIPTION: LOCALE SOURCE DEFINITION FILE.
%

LC_CTYPE

upper <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>


lower <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>


space <tab>;<newline>;<vertical-tab>;<form-feed>;<carriage-return>;/
<space>

cntrl <NUL>;<SOH>;<STX>;<ETX>;<EOT>;<ENQ>;<ACK>;<alert>;<backspace>;/
<tab>;<newline>;<vertical-tab>;<form-feed>;<carriage-return>;/
<SO>;<SI>;<DLE>;<DC1>;<DC2>;<DC3>;<DC4>;<NAK>;<SYN>;<ETB>;/
<CAN>;<EM>;<SUB>;<ESC>;<IS4>;<IS3>;<IS2>;<IS1>;<DEL>


punct <exclamation-mark>;<quotation-mark>;<number-sign>;/
<dollar-sign>;<percent-sign>;<ampersand>;<apostrophe>;/
<left-parenthesis>;<right-parenthesis>;<asterisk>;<plus-sign>;/
<comma>;<hyphen>;<period>;<slash>;/
<colon>;<semicolon>;<less-than-sign>;/
<equals-sign>;<greater-than-sign>;<question-mark>;/
<commercial-at>;/
<left-square-bracket>;<backslash>;/
<right-square-bracket>;<circumflex>;/
<underscore>;<grave-accent>;/
<left-curly-bracket>;<vertical-line>;<right-curly-bracket>;/
<tilde>

digit <zero>;<one>;<two>;<three>;<four>;/
<five>;<six>;<seven>;<eight>;<nine>

xdigit <zero>;<one>;<two>;<three>;<four>;/
<five>;<six>;<seven>;<eight>;<nine>;/
<A>;<B>;<C>;<D>;<E>;<F>;/
<a>;<b>;<c>;<d>;<e>;<f>

blank <space>;/
<tab>

toupper (<a>,<A>);(<b>,<B>);(<c>,<C>);(<d>,<D>);(<e>,<E>);/
(<f>,<F>);(<g>,<G>);(<h>,<H>);(<i>,<I>);(<j>,<J>);/
(<k>,<K>);(<l>,<L>);(<m>,<M>);(<n>,<N>);(<o>,<O>);/
(<p>,<P>);(<q>,<Q>);(<r>,<R>);(<s>,<S>);(<t>,<T>);/
(<u>,<U>);(<v>,<V>);(<w>,<W>);(<x>,<X>);(<y>,<Y>);/
(<z>,<Z>)

tolower (<A>,<a>);(<B>,<b>);(<C>,<c>);(<D>,<d>);(<E>,<e>);/
(<F>,<f>);(<G>,<g>);(<H>,<h>);(<I>,<i>);(<J>,<j>);/
(<K>,<k>);(<L>,<l>);(<M>,<m>);(<N>,<n>);(<O>,<o>);/
(<P>,<p>);(<Q>,<q>);(<R>,<r>);(<S>,<s>);(<T>,<t>);/
```

```
(<U>,<u>);(<V>,<v>);(<W>,<w>);(<X>,<x>);(<Y>,<y>);/
(<Z>,<z>)

END LC_CTYPE

LC_COLLATE

order_start

<NUL>
<SOH>
<STX>
<ETX>
<EOT>
<ENQ>
<ACK>
<alert>
<backspace>
<tab>
<newline>
<vertical-tab>
<form-feed>
<carriage-return>
<SO>
<SI>
<DLE>
<DC1>
<DC2>
<DC3>
<DC4>
<NAK>
<SYN>
<ETB>
<CAN>
<EM>
<SUB>
<ESC>
<IS4>
<IS3>
<IS2>
<IS1>
<space>
<exclamation-mark>
<quotation-mark>
<number-sign>
<dollar-sign>
<percent-sign>
<ampersand>
<apostrophe>
<left-parenthesis>
<right-parenthesis>
<asterisk>
<plus-sign>
<comma>
<hyphen>
<period>
<slash>
<zero>
<one>
<two>
<three>
<four>
<five>
<six>
<seven>
<eight>
<nine>
```

<colon>
<semicolon>
<less-than-sign>
<equals-sign>
<greater-than-sign>
<question-mark>
<commercial-at>
<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>
<left-square-bracket>
<backslash>
<right-square-bracket>
<circumflex>
<underscore>
<grave-accent>
<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>
<left-curly-bracket>
<vertical-line>

```
<right-curly-bracket>
<tilde>
<DEL>
UNDEFINED

order_end

END LC_COLLATE

LC_MONETARY

int_curr_symbol ""
currency_symbol ""
mon_decimal_point ""
mon_thousands_sep ""
mon_grouping -1
positive_sign ""
negative_sign ""
int_frac_digits -1
frac_digits -1
p_cs_precedes -1
p_sep_by_space -1
n_cs_precedes -1
n_sep_by_space -1
p_sign_posn -1
n_sign_posn -1

END LC_MONETARY

LC_NUMERIC

decimal_point "<period>" thousands_sep
"" grouping -1

END LC_NUMERIC

LC_TIME

abday "<S><u><n>";/
"<M><o><n>";/
"<T><u><e>";/
"<W><e><d>";/
"<T><h><u>";/
"<F><r><i>";/
"<S><a><t>"

day "<S><u><n><d><a><y>";/
"<M><o><n><d><a><y>";/
"<T><u><e><s><d><a><y>";/
"<W><e><d><n><e><s><d><a><y>";/
"<T><h><u><r><s><d><a><y>";/
"<F><r><i><d><a><y>";/
"<S><a><t><u><r><d><a><y>"


abmon "<J><a><n>";/
"<F><e><b>";/
"<M><a><r>";/
"<A><p><r>";/
"<M><a><y>";/
"<J><u><n>";/
"<J><u><l>";/
"<A><u><g>";/
"<S><e><p>";/
"<O><c><t>";/
"<N><o><v>";/
"<D><e><c>"
```

```
mon "<J><a><n><u><a><r><y>";/
"<F><e><b><r><u><a><r><y>";/
"<M><a><r><c><h>";/
"<A><p><r><i><l>";/
"<M><a><y>";/
"<J><u><n><e>";/
"<J><u><l><y>";/
"<A><u><g><u><s><t>";/
"<S><e><p><t><e><m><b><e><r>";/
"<O><c><t><o><b><e><r>";/
"<N><o><v><e><m><b><e><r>";/
"<D><e><c><e><m><b><e><r>"


d_t_fmt "%a %b %d %H:%M:%S %Z %Y"

d_fmt "%m//%d//%y"

t_fmt "%H:%M:%S"

am_pm "<A><M>";"<P><M>"

t_fmt_ampm "%I:%M:%S %p"

END LC_TIME

LC_MESSAGES

yesexpr "[yY][eE][sS]|[yY]"
noexpr "[nN][oO]|[nN]"
yesstr "yes"
nostr "no"

END LC_MESSAGES

LC_TOD

tzdiff 0
tname ""
dstname ""
dststart 0,0,0,0
dstend 0,0,0,0
dstshift 0

END LC_TOD
```

Related concepts

"Locale categories" on page 201
This table describes the locale categories that are supported on the i5/OS operating system.

Example: EN_US locale:

In this example, you can find the locale categories and see the source.

Note: By using the code example, you agree to the terms of the "Code license and disclaimer information" on page 445.

```
comment_char <percent-sign>
escape_char  <slash>

%
% 5716SS1         (C) COPYRIGHT IBM(R) CORP. 1991,1996
% ALL RIGHTS RESERVED.
% US GOVERNMENT USERS RESTRICTED RIGHTS -
```

```
% USE, DUPLICATION OR DISCLOSURE RESTRICTED
% BY GSA ADP SCHEDULE CONTRACT WITH IBM CORP.
%
% LICENSED MATERIALS-PROPERTY OF IBM
%
% FILE NAME  :    EN_US
%
% COUNTRY/REGION: UNITED STATES
%
% LANGUAGES(S):   ENGLISH
%
% DESCRIPTION:    LOCALE SOURCE DEFINITION FILE.
%


LC_CTYPE

upper   <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-acute>;<A-grave>;<A-circumflex>;<A-diaresis>;/
<A-tilde>;<A-ring>;<AE>;<C-cedilla>;<Eth>;<E-acute>;/
<E-grave>;<E-circumflex>;<E-diaresis>;<I-acute>;/
<I-grave>;<I-circumflex>;<I-diaresis>;<N-tilde>;/
<O-acute>;<O-grave>;<O-circumflex>;<O-diaresis>;/
<O-tilde>;<O-slash>;<Thorn>;<U-acute>;<U-grave>;/
<U-circumflex>;<U-diaresis>;<Y-acute>

lower   <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-acute>;<a-grave>;<a-circumflex>;<a-diaresis>;/
<a-tilde>;<a-ring>;<ae>;<c-cedilla>;<eth>;<e-acute>;/
<e-grave>;<e-circumflex>;<e-diaresis>;<i-acute>;/
<i-grave>;<i-circumflex>;<i-diaresis>;<n-tilde>;/
<o-acute>;<o-grave>;<o-circumflex>;<o-diaresis>;/
<o-tilde>;<o-slash>;<s-sharp>;<thorn>;<u-acute>;/
<u-grave>;<u-circumflex>;<u-diaresis>;<y-acute>;/
<y-diaresis>

space   <tab>;<newline>;<vertical-tab>;<form-feed>;<carriage-return>;/
<space>

cntrl   <NUL>;<SOH>;<STX>;<ETX>;<EOT>;<ENQ>;<ACK>;<alert>;<backspace>;/
<tab>;<newline>;<vertical-tab>;<form-feed>;<carriage-return>;/
<SO>;<SI>;<DLE>;<DC1>;<DC2>;<DC3>;<DC4>;<NAK>;<SYN>;<ETB>;/
<CAN>;<EM>;<SUB>;<ESC>;<IS4>;<IS3>;<IS2>;<IS1>;<DEL>;/
<DS>;<SOS>;<FS>;<WUS>;<BYP>;<NL>;<RNL>;<POC>;<SA>;<SFE>;<SM>;/
<CSP>;<MFA>;<SPS>;<RPT>;<CU1>;<DCS>;<PU1>;<UBS>;<IR>;<PP>;/
<TRN>;<NBS>;<GE>;<SBS>;<IT>;<RFF>;<CU3>;<SEL>;<RES>;<PM>;<EO>

graph   <exclamation-mark>;<quotation-mark>;<number-sign>; /
<dollar-sign>;<percent-sign>;<ampersand>;<apostrophe>;    /
<left-parenthesis>;<right-parenthesis>;<asterisk>;<plus-sign>;/
<comma>;<hyphen-minus>;<period>;<slash>;/
<zero>;<one>;<two>;<three>;<four>;<five>;<six>;<seven>;/
<eight>;<nine>;<colon>;<semicolon>;<less-than-sign>;  /
<equals-sign>;<greater-than-sign>;<question-mark>;/
<commercial-at>;<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>;/
<left-square-bracket>;<backslash>;/
<right-square-bracket>;<circumflex>;/
<underscore>;<grave-accent>;/
<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>;/
<left-brace>;<vertical-line>;<right-brace>;/
<tilde>;<C-cedilla>;<u-diaresis>;<e-acute>;<a-circumflex>;/
<a-diaresis>;<a-grave>;<a-ring>;<c-cedilla>;<e-circumflex>;/
<e-diaresis>;<e-grave>;<i-diaresis>;<i-circumflex>;/
```

```
<i-grave>;<A-diaresis>;<A-ring>;<E-acute>;<ae>;<AE>;/
<o-circumflex>;<o-diaresis>;<o-grave>;<u-circumflex>;/
<u-grave>;<y-diaresis>;<O-diaresis>;<U-diaresis>;<o-slash>;/
<sterling>;<O-slash>;<multiply>;<a-acute>;<i-acute>;/
<o-acute>;<u-acute>;<n-tilde>;<N-tilde>;<feminine>;/
<masculine>;<question-down>;<registered>;<not>;<one-half>;/
<one-quarter>;<exclamation-down>;<guillemot-left>;/
<guillemot-right>;<A-acute>;<A-circumflex>;<A-grave>;/
<copyright>;<cent>;<yen>;<a-tilde>;<A-tilde>;<currency>;/
<eth>;<Eth>;<E-circumflex>;<E-diaresis>;<E-grave>;/
<I-acute>;<I-circumflex>;<I-diaresis>;<broken-bar>;/
<I-grave>;<O-acute>;<s-sharp>;<O-circumflex>;/
<O-grave>;<o-tilde>;<O-tilde>;<mu>;<thorn>;<Thorn>;<U-acute>;/
<U-circumflex>;<U-grave>;<y-acute>;<Y-acute>;<macron>;/
<acute>;<hyphen>;<plus-minus>;<three-quarters>;<paragraph>;/
<section>;<divide>;<cedilla>;<degree>;<diaresis>;<dot>;/
<one-superior>;<three-superior>;<two-superior>

print   <space>;<exclamation-mark>;<quotation-mark>;<number-sign>; /
<dollar-sign>;<percent-sign>;<ampersand>;<apostrophe>;    /
<left-parenthesis>;<right-parenthesis>;<asterisk>;<plus-sign>;/
<comma>;<hyphen-minus>;<period>;<slash>;/
<zero>;<one>;<two>;<three>;<four>;<five>;<six>;<seven>;/
<eight>;<nine>;<colon>;<semicolon>;<less-than-sign>;  /
<equals-sign>;<greater-than-sign>;<question-mark>;/
<commercial-at>;<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>;/
<left-square-bracket>;<backslash>;/
<right-square-bracket>;<circumflex>;/
<underscore>;<grave-accent>;/
<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>;/
<left-brace>;<vertical-line>;<right-brace>;/
<tilde>;<C-cedilla>;<u-diaresis>;<e-acute>;<a-circumflex>;/
<a-diaresis>;<a-grave>;<a-ring>;<c-cedilla>;<e-circumflex>;/
<e-diaresis>;<e-grave>;<i-diaresis>;<i-circumflex>;/
<i-grave>;<A-diaresis>;<A-ring>;<E-acute>;<ae>;<AE>;/
<o-circumflex>;<o-diaresis>;<o-grave>;<u-circumflex>;/
<u-grave>;<y-diaresis>;<O-diaresis>;<U-diaresis>;<o-slash>;/
<sterling>;<O-slash>;<multiply>;<a-acute>;<i-acute>;/
<o-acute>;<u-acute>;<n-tilde>;<N-tilde>;<feminine>;/
<masculine>;<question-down>;<registered>;<not>;<one-half>;/
<one-quarter>;<exclamation-down>;<guillemot-left>;/
<guillemot-right>;<A-acute>;<A-circumflex>;<A-grave>;/
<copyright>;<cent>;<yen>;<a-tilde>;<A-tilde>;<currency>;/
<eth>;<Eth>;<E-circumflex>;<E-diaresis>;<E-grave>;/
<I-acute>;<I-circumflex>;<I-diaresis>;<broken-bar>;/
<I-grave>;<O-acute>;<s-sharp>;<O-circumflex>;/
<O-grave>;<o-tilde>;<O-tilde>;<mu>;<thorn>;<Thorn>;<U-acute>;/
<U-circumflex>;<U-grave>;<y-acute>;<Y-acute>;<macron>;/
<acute>;<hyphen>;<plus-minus>;<three-quarters>;<paragraph>;/
<section>;<divide>;<cedilla>;<degree>;<diaresis>;<dot>;/
<one-superior>;<three-superior>;<two-superior>

punct   <exclamation-mark>;<quotation-mark>;<number-sign>; /
<dollar-sign>;<percent-sign>;<ampersand>;<apostrophe>;     /
<left-parenthesis>;<right-parenthesis>;<asterisk>;<plus-sign>;/
<comma>;<hyphen-minus>;<period>;<slash>;/
<colon>;<semicolon>;<less-than-sign>;   /
<equals-sign>;<greater-than-sign>;<question-mark>;/
<commercial-at>;/
<left-square-bracket>;<backslash>;/
<right-square-bracket>;<circumflex>;/
<underscore>;<grave-accent>;/
<left-brace>;<vertical-line>;<right-brace>;/
<tilde>
```

```
digit   <zero>;<one>;<two>;<three>;<four>;/
<five>;<six>;<seven>;<eight>;<nine>

xdigit  <zero>;<one>;<two>;<three>;<four>;/
<five>;<six>;<seven>;<eight>;<nine>;/
<A>;<B>;<C>;<D>;<E>;<F>;/
<a>;<b>;<c>;<d>;<e>;<f>

blank    <space>;/
<tab>

toupper (<a>,<A>);(<b>,<B>);(<c>,<C>);(<d>,<D>);(<e>,<E>);/
(<f>,<F>);(<g>,<G>);(<h>,<H>);(<i>,<I>);(<j>,<J>);/
(<k>,<K>);(<l>,<L>);(<m>,<M>);(<n>,<N>);(<o>,<O>);/
(<p>,<P>);(<q>,<Q>);(<r>,<R>);(<s>,<S>);(<t>,<T>);/
(<u>,<U>);(<v>,<V>);(<w>,<W>);(<x>,<X>);(<y>,<Y>);/
(<z>,<Z>);(<a-acute>,<A-acute>);(<a-grave>,<A-grave>);/
(<a-circumflex>,<A-circumflex>);(<a-diaresis>,<A-diaresis>);/
(<a-tilde>,<A-tilde>);(<a-ring>,<A-ring>);(<ae>,<AE>);/
(<c-cedilla>,<C-cedilla>);(<eth>,<Eth>);(<e-acute>,<E-acute>);/
(<e-grave>,<E-grave>);(<e-circumflex>,<E-circumflex>);/
(<e-diaresis>,<E-diaresis>);(<i-acute>,<I-acute>);/
(<i-grave>,<I-grave>);(<i-circumflex>,<I-circumflex>);/
(<i-diaresis>,<I-diaresis>);(<n-tilde>,<N-tilde>);/
(<o-acute>,<O-acute>);(<o-grave>,<O-grave>);/
(<o-circumflex>,<O-circumflex>);(<o-diaresis>,<O-diaresis>);/
(<o-tilde>,<O-tilde>);(<o-slash>,<O-slash>);(<thorn>,<Thorn>);/
(<u-acute>,<U-acute>);(<u-grave>,<U-grave>);/
(<u-circumflex>,<U-circumflex>);(<u-diaresis>,<U-diaresis>);/
(<y-acute>,<Y-acute>);(<y-diaresis>,<Y>)

tolower (<A>,<a>);(<B>,<b>);(<C>,<c>);(<D>,<d>);(<E>,<e>);/
(<F>,<f>);(<G>,<g>);(<H>,<h>);(<I>,<i>);(<J>,<j>);/
(<K>,<k>);(<L>,<l>);(<M>,<m>);(<N>,<n>);(<O>,<o>);/
(<P>,<p>);(<Q>,<q>);(<R>,<r>);(<S>,<s>);(<T>,<t>);/
(<U>,<u>);(<V>,<v>);(<W>,<w>);(<X>,<x>);(<Y>,<y>);/
(<Z>,<z>);(<A-acute>,<a-acute>);(<A-grave>,<a-grave>);/
(<A-circumflex>,<a-circumflex>);(<A-diaresis>,<a-diaresis>);/
(<A-tilde>,<a-tilde>);(<A-ring>,<a-ring>);(<AE>,<ae>);/
(<C-cedilla>,<c-cedilla>);(<Eth>,<eth>);(<E-acute>,<e-acute>);/
(<E-grave>,<e-grave>);(<E-circumflex>,<e-circumflex>);/
(<E-diaresis>,<e-diaresis>);(<I-acute>,<i-acute>);/
(<I-grave>,<i-grave>);(<I-circumflex>,<i-circumflex>);/
(<I-diaresis>,<i-diaresis>);(<N-tilde>,<n-tilde>);/
(<O-acute>,<o-acute>);(<O-grave>,<o-grave>);/
(<O-circumflex>,<o-circumflex>);(<O-diaresis>,<o-diaresis>);/
(<O-tilde>,<o-tilde>);(<O-slash>,<o-slash>);(<Thorn>,<thorn>);/
(<U-acute>,<u-acute>);(<U-grave>,<u-grave>);/
(<U-circumflex>,<u-circumflex>);(<U-diaresis>,<u-diaresis>);/
(<Y-acute>,<y-acute>)

END LC_CTYPE


LC_COLLATE

order_start

<NUL>
<SOH>
<STX>
<ETX>
<SEL>
<tab>
<RNL>
<DEL>
<GE>
```

```
<SPS>
<RPT>
<vertical-tab>
<form-feed>
<carriage-return>
<SO>
<SI>
<DLE>
<DC1>
<DC2>
<DC3>
<RES>
<NL>
<backspace>
<POC>
<CAN>
<EM>
<UBS>
<CU1>
<IS4>
<IS3>
<IS2>
<IS1>
<DS>
<SOS>
<FS>
<WUS>
<BYP>
<newline>
<ETB>
<ESC>
<SA>
<SFE>
<SM>
<CSP>
<MFA>
<ENQ>
<ACK>
<alert>
<SYN>
<IR>
<PP>
<TRN>
<NBS>
<EOT>
<SBS>
<IT>
<RFF>
<CU3>
<DC4>
<NAK>
<SUB>
<EO>
<space>
<underscore>
<macron>
<hyphen>
<hyphen-minus>
<comma>
<semicolon>
<colon>
<exclamation-mark>
<exclamation-down>
<question-mark>
<question-down>
<slash>
<period>
```

```
<acute>
<grave-accent>
<circumflex>
<diaresis>
<tilde>
<dot>
<cedilla>
<apostrophe>
<quotation-mark>
<guillemot-left>
<guillemot-right>
<left-parenthesis>
<right-parenthesis>
<left-square-bracket>
<right-square-bracket>
<left-brace>
<right-brace>
<section>
<paragraph>
<copyright>
<registered>
<commercial-at>
<currency>
<cent>
<dollar-sign>
<sterling>
<yen>
<asterisk>
<backslash>
<ampersand>
<number-sign>
<percent-sign>
<plus-sign>
<plus-minus>
<divide>
<multiply>
<less-than-sign>
<equals-sign>
<greater-than-sign>
<not>
<vertical-line>
<broken-bar>
<degree>
<mu>
<nobreakspace>
<zero>
<one-quarter>
<one-half>
<three-quarters>
<one>
<one-superior>
<two>
<two-superior>
<three>
<three-superior>
<four>
<five>
<six>
<seven>
<eight>
<nine>
<a>
<A>
<a-acute>
<A-acute>
<feminine>
<a-grave>
```

```
<A-grave>
<a-circumflex>
<A-circumflex>
<a-ring>
<A-ring>
<a-diaresis>
<A-diaresis>
<a-tilde>
<A-tilde>
<ae>
<AE>
<b>
<B>
<c>
<C>
<c-cedilla>
<C-cedilla>
<d>
<D>
<eth>
<Eth>
<e>
<E>
<e-acute>
<E-acute>
<e-grave>
<E-grave>
<e-circumflex>
<E-circumflex>
<e-diaresis>
<E-diaresis>
<f>
<F>
<g>
<G>
<h>
<H>
<i-dotless>
<i>
<I>
<i-acute>
<I-acute>
<i-grave>
<I-grave>
<i-circumflex>
<I-circumflex>
<i-diaresis>
<I-diaresis>
<j>
<J>
<k>
<K>
<l>
<L>
<m>
<M>
<n>
<N>
<n-tilde>
<N-tilde>
<o>
<O>
<masculine>
<o-acute>
<O-acute>
<o-grave>
<O-grave>
```

```
<o-circumflex>
<O-circumflex>
<o-diaresis>
<O-diaresis>
<o-tilde>
<O-tilde>
<o-slash>
<O-slash>
<p>
<q>
<Q>
<r>
<R>
<s>
<S>
<s-sharp>
<t>
<T>
<thorn>
<Thorn>
<u>
<U>
<u-acute>
<U-acute>
<u-grave>
<U-grave>
<u-circumflex>
<U-circumflex>
<u-diaresis>
<U-diaresis>
<v>
<V>
<w>
<W>
<x>
<X>
<y>
<Y>
<y-acute>
<Y-acute>
<y-diaresis>
<z>
<Z>
UNDEFINED

order_end

END LC_COLLATE


LC_MONETARY

int_curr_symbol   "<U><S><D><space>"
currency_symbol   "<dollar-sign>"
mon_decimal_point "<period>"
mon_thousands_sep "<comma>"
mon_grouping      3
positive_sign     ""
negative_sign     "<hyphen-minus>"
int_frac_digits   2
frac_digits       2
p_cs_precedes     1
p_sep_by_space    0
n_cs_precedes     1
n_sep_by_space    0
p_sign_posn       2
n_sign_posn       2
```

```
END LC_MONETARY


LC_NUMERIC

decimal_point     "<period>"
thousands_sep     "<comma>"
grouping          3

END LC_NUMERIC


LC_TIME

abday   "<S><u><n>";/
"<M><o><n>";/
"<T><u><e>";/
"<W><e><d>";/
"<T><h><u>";/
"<F><r><i>";/
"<S><a><t>"

day     "<S><u><n><d><a><y>";/
"<M><o><n><d><a><y>";/
"<T><u><e><s><d><a><y>";/
"<W><e><d><n><e><s><d><a><y>";/
"<T><h><u><r><s><d><a><y>";/
"<F><r><i><d><a><y>";/
"<S><a><t><u><r><d><a><y>"

abmon   "<J><a><n>";/
"<F><e><b>";/
"<M><a><r>";/
"<A><p><r>";/
"<M><a><y>";/
"<J><u><n>";/
"<J><u><l>";/
"<A><u><g>";/
"<S><e><p>";/
"<O><c><t>";/
"<N><o><v>";/
"<D><e><c>"

mon
"<J><a><n><u><a><r><y>";/
"<F><e><b><r><u><a><r><y>";/
"<M><a><r><c><h>";/
"<A><p><r><i><l>";/
"<M><a><y>";/
"<J><u><n><e>";/
"<J><u><l><y>";/
"<A><u><g><u><s><t>";/
"<S><e><p><t><e><m><b><e><r>";/
"<O><c><t><o><b><e><r>";/
"<N><o><v><e><m><b><e><r>";/
"<D><e><c><e><m><b><e><r>"

d_t_fmt "%a %b %e %H:%M:%S %Z %Y"

d_fmt   "%m//%d//%y"

t_fmt   "%H:%M:%S"

am_pm   "<A><M>";"<P><M>"

END LC_TIME
```

```
LC_MESSAGES

yesexpr "[yY][eE][sS]|[yY]"
noexpr  "[nN][oO]|[nN]"
yesstr  "yes:y:Y"
nostr   "no:n:N"

END LC_MESSAGES

LC_TOD

tzdiff   0
tname    ""
dstname  ""
dststart 0,0,0,0
dstend   0,0,0,0
dstshift 0

END LC_TOD
```

Globalization reference information

Here is a comprehensive collection of supporting information about the concepts and tasks described in the Globalization topic collection.

National language version feature codes

This table lists the available national language version feature codes on the i5/OS operating system. When you order an i5/OS licensed program, you identify the national language version you want by specifying a language feature code.

National language version	Primary language feature code	Secondary language feature code
Albanian	2995	5595
Arabic	2954	5554
Belgian Dutch	2963	5563
Belgian English	2909	5509
Belgian French	2966	5566
Brazilian Portuguese	2980	5580
Bulgarian	2974	5574
Canadian French	2981	5581
Croatian	2912	5512
Czech	2975	5575
Danish	2926	5526
Dutch Netherlands	2923	5523
English	2924	5524
English Uppercase Support for Double-Byte Character Set (DBCS)	2938	5538
English Uppercase and Lowercase Support for Double-Byte Character Set (DBCS)	2984	5584
Estonian	2902	5502
Farsi	2998	5598
Finnish	2925	5525

National language version	Primary language feature code	Secondary language feature code
French	2928	5528
French Multinational Character Set	2940	5540
German	2929	5529
German Multinational Character Set	2939	5539
Greek	2957	5557
Hebrew	2961	5561
Hungarian	2976	5576
Icelandic	2958	5558
Italian	2932	5532
Italian Multinational Character Set	2942	5542
Japanese Universal	2930	5530
Japanese Kanji	2962	5562
Korean	2986	5586
Lao	2906	5506
Latvian	2904	5504
Lithuanian	2903	5503
Macedonian	2913	5513
Norwegian	2933	5533
Polish	2978	5578
Portuguese	2922	5522
Portuguese Multinational Character Set	2996	5596
Romanian	2992	5592
Russian	2979	5579
Serbian Cyrillic	2914	5514
Simplified Chinese	2989	5589
Slovakian	2994	5594
Slovenian	2911	5511
Spanish	2931	5531
Swedish	2937	5537
Thai	2972	5572
Traditional Chinese	2987	5587
Turkish	2956	5556
Vietnamese	2905	5505

Related concepts

"National language version" on page 3
A *national language version (NLV)* is a version of the i5/OS operating system that contains a predefined set of language-dependent values, such as date format, time format, and sort sequence, for a particular language.

"Setting up i5/OS with a national language version" on page 29
The steps to install and configure a national language version on the i5/OS operating system include selecting and installing hardware, installing software, and configuring your environment to run in a globalized setting.

"Configuring secondary languages" on page 39
A secondary language consists of textual data for all licensed programs supported for a national language version.

"Example: Locales as part of a multilingual environment" on page 230
The i5/OS operating system, through the use of locales, user profiles, and subsystems, can provide a multilingual environment. Users of a system that is set up for multilingual environments can work with their national language and all of its cultural conventions (for example, the character used to separate hours, minutes, and seconds).

Related information

Installing, upgrading, or deleting i5/OS and related software

Country and region identifiers

This table lists the country and region identifiers.

Country and region name	**Country and region ID**
Afghanistan	AF
Albania	AL
Algeria	DZ
American Samoa	AS
Andorra	AD
Angola	AO
Anguilla	AI
Antarctica	AQ
Antigua and Barbuda	AG
Arabic speaking countries	AA
Argentina	AR
Armenia	AM
Aruba	AW
Australia	AU
Austria	AT
Azerbaijan	AZ
Bahamas	BS
Bahrain	BH
Bangladesh	BD
Barbados	BB
Belarus	BY
Belgium	BE
Belize	BZ
Benin	BJ
Bermuda	BM
Bhutan	BT

Country and region name	Country and region ID
Bolivia	BO
Bosnia/Herzegovina	BA
Botswana	BW
Bouvet island	BV
Brazil	BR
British Indian ocean territory	IO
Brunei Darussalam	BN
Bulgaria	BG
Burkina Faso	BF
Burundi	BI
Burma	BU
Cambodia	KH
Cameroon, United Republic of	CM
Canada	CA
Cape Verde	CV
Cayman islands	KY
Central African Republic	CF
Chad	TD
Chile	CL
China	CN
China (Hong Kong S.A.R.)	HK
China (Macao S.A.R.)	MO
Christmas island	CX
Cocos (Keeling) islands	CC
Colombia	CO
Comoros	KM
Congo	CG
Cook islands	CK
Costa Rica	CR
Ivory coast	CI
Croatia	HR
Cuba	CU
Cyprus	CY
Czech Republic	CZ
Denmark	DK
Djibouti	DJ
Dominica	DM
Dominican Republic	DO
East Timor	TP
Ecuador	EC
Egypt	EG

Country and region name	Country and region ID
El Salvador	SV
Equatorial Guinea	GQ
Estonia	EE
Ethiopia	ET
Falkland islands (Malvinas)	FK
Faroe islands	FO
Fiji	FJ
Finland	FI
France	FR
French Guiana	GF
French Polynesia	PF
French southern territories	TF
Gabon	GA
Gambia	GM
Georgia	GE
Germany	DE
Ghana	GH
Gibraltar	GI
Greece	GR
Greenland	GL
Grenada	GD
Guadeloupe	GP
Guam	GU
Guatemala	GT
Guinea	GN
Guinea-Bissau	GW
Guyana	GY
Haiti	HT
Heard and McDonald islands	HM
Honduras	HN
Hungary	HU
Iceland	IS
India	IN
Indonesia	ID
Iran (Islamic Republic of)	IR
Iraq	IQ
Ireland	IE
Israel	IL
Italy	IT
Jamaica	JM
Japan	JP

Country and region name	Country and region ID
Jordan	JO
Kazakhstan	KK
Kenya	KE
Kiribati	KI
Korea, Democratic People's Republic of	KP
Korea, Republic of	KR
Kuwait	KW
Kyrgyzstan	KG
Lao People's Democratic Republic	LA
Latvia	LV
Lebanon	LB
Lesotho	LS
Liberia	LR
Libyan Arab Jamahiriya	LY
Liechtenstein	LI
Lithuania	LT
Luxembourg	LU
Macedonia	MK
Madagascar	MG
Malawi	MW
Malaysia	MY
Maldives	MV
Mali	ML
Malta	MT
Marshall islands	MH
Martinique	MQ
Mauritania	MR
Mauritius	MU
Mexico	MX
Micronesia	FM
Moldava, Republic of	MD
Monaco	MC
Mongolia	MN
Montenegro	ME
Montserrat	MS
Morocco	MA
Mozambique	MZ
Myanmar	MM
Namibia	NA
Nauru	NR
Nepal	NP

Country and region name	Country and region ID
Netherlands	NL
Netherlands Antilles	AN
New Caledonia	NC
Neutral zone	NT
New Zealand	NZ
Nicaragua	NI
Niger	NE
Nigeria	NG
Niue	NU
Norfolk island	NF
Northern Mariana islands	MP
Norway	NO
Oman	OM
Pakistan	PK
Palau	PW
Panama	PA
Papua New Guinea	PG
Paraguay	PY
Peru	PE
Philippines	PH
Pitcairn	PN
Poland	PL
Portugal	PT
Puerto Rico	PR
Qatar	QA
Reunion	RE
Romania	RO
Russia	RU
Rwanda	RW
Saint. Helena	SH
Saint Kitts and Nevis	KN
Saint Lucia	LC
Saint Pierre and Miquelon	PM
Saint Vincent and the Grenadines.	VC
Western Samoa	WS
San Marino	SM
Sao Tome and Principe	ST
Saudi Arabia	SA
Senegal	SN
Seychelles	SC
Sierra Leone	SL

Country and region name	Country and region ID
Serbia	SQ
Singapore	SG
Slovakia	SK
Slovenia	SI
Solomon islands	SB
Somalia	SO
South Africa	ZA
Spain	ES
Sri Lanka	LK
Sudan	SD
Suriname	SR
Svalbard and Jan Mayen islands	SJ
Swaziland	SZ
Sweden	SE
Switzerland	CH
Syrian Arab Republic	SY
Taiwan	TW
Tajikistan	TJ
Tanzania, United Republic of	TZ
Thailand	TH
Togo	TG
Tokelau	TK
Tonga	TO
Trinidad and Tobago	TT
Tunisia	TN
Turkmenistan	TM
Turkey	TR
Turks and Caicos islands	TC
Tuvalu	TV
Uganda	UG
Ukraine	UA
United Arab Emirates	AE
United Kingdom	GB
United States Minor Outlying Islands	UM
United States of America	US
Uruguay	UY
Uzbekistan	UZ
Vanuatu	VU
Vatican City State	VA
Venezuela	VE
Vietnam	VN

Country and region name	Country and region ID
Virgin islands (British)	VG
Virgin islands (U.S.)	VI
Wallis and Futuna islands	WF
Western Sahara	EH
Yemen, Republic of	YE
Countries of the former Yugoslavia	YU
Zaire	ZR
Zambia	ZM
Zimbabwe	ZW

Related concepts

"Country or region identifier (QCNTRYID) system value" on page 17
The country or region identifier (QCNTRYID) system value indicates the default country or region identifier for the system.

"ILE RPG sort sequence" on page 100
The ILE RPG feature, an option of the IBM WebSphere Development Studio for System i licensed program, provides the possibility for a user to specify a sort sequence table and to use the table in comparison operations that are performed with nonnumeric data.

Default system values for national language versions

Jobs and functions on the i5/OS operating system use system values as default values.

The values for each national language version are listed by keyword. These are the values you see when you display the CPX8416 message for a particular language library. The tables contain Internet values that are needed to configure a Web browser for a specific language. The Internet values are also listed by keyword.

Values that are provided in the tables

Keyword	Description
QCCSID	Character set identifier. This is the recommended QCCSID value if you want to use CDRA support. For all national language versions (NLVs), the default QCCSID value is 65535 unless otherwise indicated.
QCHRID	Character set and code page
QCNTRYID	Country or region identifier
QCURSYM	Currency symbol. The values given are accurate; however, the system supports only 1 character in that return field.
QDATFMT	Date format
QDATSEP	Date separator
QDECFMT	Decimal format
QIGC	DBCS version indicator
QIGCCDEFNT	DBCS font name
QKBDTYPE	Keyboard type
QLANGID	Language identifier
QLEAPADJ	Leap year adjustment

Keyword	Description
QLOCALE	Locale. On newly-installed systems, the system sets the QLOCALE system value to a default locale based on the primary NLV that is installed. For example, the value will be set to `/QSYS.LIB/DA_DK.LOCALE` if you install the 2926 NLV.
QSRTSEQ	Sort sequence
QTIMSEP	Time separator
Internet CCSID	Client character set environment.
Client encoding nomenclature	NLTC (National Language Technical Center) value and Document Encoding. Client encoding nomenclature provides a guideline for configuring a client for a specific language and setting up your internet Web browser.

Related concepts

"System values" on page 14
The system values of the primary language on the system are used as system-wide cultural and linguistic defaults. Therefore, if you change the primary language on the system, each varying system value resets to the default system value of the new primary language.

"Configuring the primary language" on page 38
A primary language consists of program code, textual data for each licensed program ordered, and default national language cultural values.

"Packaging and installation process" on page 55
You need to consider the running code, translated textual data, and installation documents when packaging applications. Here are some suggestions for simplifying the packaging and installation of your application.

"Cultural-dependent design" on page 79
Different countries might have different standards, which you must consider when developing an NLS-enabled application. This culturally sensitive information must be placed outside the program the same way as the textual data is handled.

"System values for other languages with no national language version" on page 270
Some of the system values are associated with languages and countries that do not have a national language version. You should set these values immediately after initially installing the i5/OS operating system.

Albanian (Feature 2995)

The table shows the default system values for the Albanian (Feature 2995) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	ALI	Lek	-	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	colon (:)	00500	AL	SQ_AL	SQI	ISO-8859-1 Latin 1

Arabic (Feature 2954)

The table shows the default system values for the Arabic (Feature 2954) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00235 00420	J	CLB	Dollar ($)	Slash (/)	01089

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Comma (,)	00420	AA	AR_AA	ARA	ISO-8859-6

Belgian Dutch MNCS (Feature 2963)

The table shows the default system values for the Belgian Dutch multinational character set (MNCS) (Feature 2963) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	BLI	F	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00500	BE	NL_BE _M	NLB	ISO-8859-1 Latin 1

Belgium English (Feature 2909)

The table shows the default system values for the Belgian English (Feature 2909) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	BLI	F	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00500	BE	NE_BE	ENB	ISO-8859-1 Latin 1

Belgian French MNCS (Feature 2966)

The table shows the default system values for the Belgian French multinational character set (MNCS) (Feature 2966) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	BLI	F	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client Eencoding nomenclature
DMY	Colon (:)	00500	BE	FR_BE_M	FRB	ISO-8859-1 Latin 1

Brazilian Portuguese (Feature 2980)

The table shows the default system values for the Brazilian Portuguese (Feature 2980) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00037	J	BRB	Cruzeiro ($)	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00037	BR	PT_BR	PTB	ISO-8859-1 Latin 1

Bulgarian (Feature 2974)

The table shows the default system values for the Bulgarian (Feature 2974) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01150 01025	J	BGB	Lv	-	00915

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	colon (:)	01025	BG	BG_BG	BGR	ISO-8859-5

Canadian French MNCS (Feature 2981)

The table shows the default system values for the Canadian French multinational character set (MNCS) (Feature 2981) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	CAI	Dollar ($)	Hyphen (-)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00500	CA	FR_CA_M	FRC	ISO-8859-1 Latin 1

Croatian (Feature 2912)

The table shows the default system values for the Croatian (Feature 2912) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	Blank	YGI	Croatian Kuna (K)	Hyphen (-)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	HR	HR_HR	HRV	ISO-8859-2 Latin 2

Czech (Feature 2975)

The table shows the default system values for the Czech (Feature 2975) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	J	CSB	Ceske Koruna (K)	Hyphen (-)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	CZ	CS_CZ	CSY	ISO-8859-2 Latin 2

Danish (Feature 2926)

The table shows the default system values for the Danish (Feature 2926) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00277	J	DMB	Colon (:)	Hyphen (-)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Period (.)	00277	DK	DA_DK	DAN	ISO-8859-1 Latin 1

Dutch Netherlands (Feature 2923)

The table shows the default system values for the Dutch Netherlands (Feature 2923) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00037	J	NEB	Dollar $	Hyphen (-)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00037	NL	NL_NL	NLD	ISO-8859-1 Latin 1

English Uppercase and Lowercase (Feature 2924)

The table shows the default system values for the English Uppercase and Lowercase (Feature 2924) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00037	Blank	USB	Dollar ($)	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
MDY	Colon (:)	00037	US	EN_US	ENU	ISO-8859-1 Latin 1

Related concepts

"Information in message CPX8416" on page 81
If your application is translated into other languages, use message CPX8416 from the QCPFMSG message file to get the correct setting for some cultural values for the other languages. The message exists for your primary language and all installed secondary language libraries.

English Uppercase DBCS (Feature 2938)

The table shows the default system values for the English Uppercase DBCS (Feature 2938) national language version.

If English Uppercase is installed, you must respond to messages in uppercase rather than lowercase.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00037	Blank	JKB	Yen sign	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID [1]	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
MDY	Colon (:)	65535	US	EN_US	ENP	ISO-8859-1 Latin 1

[1] Because this national language version is not specific to any country or region, 65535 is used. You should change this system value to an appropriate CCSID after installing your system, or change the job or user profile CCSID attribute. Here are the recommended CCSID values and CHRID changes that occur.

- 05026 is the CCSID for Japanese with no lowercase characters. The CHRID is set to 1172 290.
- 01399 is the CCSID for Japanese with both uppercase and lowercase characters. The CHRID is set to 01172 01027.

System and product files not explicitly assigned a CCSID value will be assigned CCSID 05035 if DBCS capable and CCSID 01027 for SBCS only files.

English Uppercase and Lowercase DBCS (Feature 2984)

The table shows the default system values for the English Uppercase and Lowercase DBCS (Feature 2984) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01175 00037	Blank	TAB	Dollar ($)	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID [1]	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
MDY	Colon (:)	65535	US	EN_US	ENU	ISO-8859-1 Latin 1

[1] Because this national language version is not specific to any country or region, 65535 is used. You should change this system value to an appropriate CCSID after installing your system, or change the job or user profile CCSID attribute. Here are the recommended CCSID values and CHRID changes.

- 00937 is the CCSID for Traditional Chinese. The CHRID is set to 1175 00037
- 00935 is the CCSID for Simplified Chinese. The CHRID is set to 01174 00836.
- 00933 is the CCSID for Korean. The CHRID is set to 01173 00833.

Estonian (Feature 2902)

The table shows the default system values for the Estonian (Feature 2902) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01307 01122	J	ESB	kr	-	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	01122	EE	ET_EE	EST	ISO-8859-1 Latin 1

Farsi (Feature 2998)

The table shows the default system values for the Farsi (Feature 2998) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01219 01097	J	IRB	Farsi Riyal	Slash (/)	

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	01097	IR	AR_AA	FAR	

Finnish (Feature 2925)

The table shows the default system values for the Finnish (Feature 2925) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00278	J	FNB	F	Period (.)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Period (.)	00278	FI	FI_FI	FIN	ISO-8859-1 Latin 1

French (Feature 2928)

The table shows the default system values for the French (Feature 2928) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00297	J	FAB	F	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00297	FR	FR_FR	FRA	ISO-8859-1 Latin 1

French MNCS (Feature 2940)

The table shows the default system values for the French multinational character set (MNCS) (Feature 2940) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	SFI	F	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00500	CH	FR_FR_M	FRS	ISO-8859-1 Latin 1

German (Feature 2929)

The table shows the default system values for the German (Feature 2929) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00273	J	AGB	Dollar ($)	Period (.)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00273	DE	DE_DE	DEU	ISO-8859-1 Latin 1

German MNCS (Feature 2939)

The table shows the default system values for the German multinational character set (MNCS) (Feature 2939) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	AGI	Dollar ($)	Period (.)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Comma (,)	00500	CH	DE_DE_M	DES	ISO-8859-1 Latin 1

Greek (Feature 2957)

The table shows the default system values for the Greek (Feature 2957) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00925 00875	J	GNB	Dollar ($)	Slash (/)	00813

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00875	GR	EL_GR	ELL	ISO-8859-7 Greek

Hebrew (Feature 2961)

The table shows the default system values for the Hebrew (Feature 2961) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00941 00424	Blank	NCB	Shin (GCGID HS210000, code point X'69' on code page 00424)	Slash (/)	00916

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00424	IL	HE_IL	HEB	ISO-8859-8

Hungarian (Feature 2976)

The table shows the default system values for the Hungarian (Feature 2976) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	J	HNB	Forint (F)	Hyphen (-)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	HU	HU_HU	HUN	ISO-8859-2 Latin 2

Icelandic (Feature 2958)

The table shows the default system values for the Icelandic (Feature 2958) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00871	J	ICB	Dollar ($)	Hyphen (-)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00871	IS	IS_IS	ISL	ISO-8859-1 Latin 1

Italian (Feature 2932)

The table shows the default system values for the Italian (Feature 2932) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00280	Blank	ITB	Lira sign	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Period (.)	00280	IT	IT_IT	ITA	ISO-8859-1 Latin 1

Italian MNCS (Feature 2942)

The table shows the default system values for the Italian MNCS (Feature 2942) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	Blank	ITI	Lira sign	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Comma (,)	00500	CH	IT_IT_M	ITS	ISO-8859-1 Latin 1

Japanese Universal (Feature 2930)

The table shows the default system values for the Japanese Universal (Feature 2930) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
1172 1027	Blank	JPB	Yen sign	Hyphen (-)	00942

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	01399	JP	JA_JP_1399	JPN	ShiftJIS

Japanese (Katakana) (Feature 2962)

The table shows the default system values for the Japanese (Katakana) (Feature 2962) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01172 00290	Blank	JKB	Yen sign	Hyphen (-)	00942

QDATFMT	QTIMSEP	QCCSID[1]	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	05026	JP	JA_JP_5026	JPN	ShiftJIS
[1] This is the recommended QCCSID value if you want to use CDRA support. For all NLVs, the default QCCSID value is 65535.						

Korean (Feature 2986)

The table shows the default system values for the Korean (Feature 2986) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01173 00833	Blank	KOB	WON sign	Period (.)	00949

QDATFMT	QTIMSEP	QCCSID[1]	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00933	KR	KO_KR	KOR	EUC Korean
[1] This is the recommended QCCSID value if you want to use CDRA support. For all NLVs, the default QCCSID value is 65535.						

Laotian (Feature 2906)

The table shows the default system values for the Laotian (Feature 2906) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
1341 1132	J	LAB	KIP	Slash (/)	

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	01132	LA	EN_US	LAO	

Latvian (Feature 2904)

The table shows the default system values for the Latvian (Feature 2904) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01305 01112	J	LVB	Ls	Period (.)	00921

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	01112	LV	LV_LV	LVA	ISO-8859-4

Lithuanian (Feature 2903)

The table shows the default system values for the Lithuanian (Feature 2903) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01305 01112	J	LTB	Lt	Period (.)	00921

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	01112	LT	LT_LT	LTU	ISO-8859-4

Macedonian (Feature 2913)

The table shows the default system values for the Macedonian (Feature 2913) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01150 01025	J	MKB	Den	-	00915

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	01025	MK	MK_MK	MKD	Cyrillic (Win1251)

Norwegian (Feature 2933)

The table shows the default system values for the Norwegian (Feature 2933) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00277	J	NWB	Dollar ($)	Period (.)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00277	NO	NO_NO	NON	ISO-8859-1 Latin 1

Polish (Feature 2978)

The table shows the default system values for the Polish (Feature 2978) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	J	PLB	Polish Zloty (Z)	Hyphen (-)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	PL	PL_PL	PLK	ISO-8859-2 Latin 2

Portuguese (Feature 2922)

The table shows the default system values for the Portuguese (Feature 2922) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00037	J	PRB	Escudo ($)	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00037	PT	PT_PT	PTG	ISO-8859-1 Latin 1

Portuguese MNCS (Feature 2996)

The table shows the default system values for the Portuguese multinational character set (MNCS) (Feature 2996) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00500	J	PRI	Escudo ($)	Hyphen (-)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00500	PT	PT_PT_M	PTG	ISO-8859-1 Latin 1

Romanian (Feature 2992)

The table shows the default system values for the Romanian (Feature 2992) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	J	RMB	lei	Period (.)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	RO	RO_RO	ROM	ISO-8859-2 Latin 2

Russian (Feature 2979)

The table shows the default system values for the Russian (Feature 2979) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01150 01025	J	RUB	Ruble (GCGID KR02000 - character X'DE' on code page 1025)	Slash (/)	1251

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Period (.)	01025	RU	RU_RU	RUS	Cyrillic (Win1251)

Serbian (Feature 2914)

The table shows the default system values for the Serbian (Feature 2914) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01150 01025	J	SQB	Din	-	00915

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Period (.)	01025	SQ	SR_SP	SRB	ISO-8859-5

Simplified Chinese (Feature 2989)

The table shows the default system values for the Simplified Chinese (Feature 2989) national language version.

QCHRID[1]	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01174 00836	Blank	RCB	Dollar ($)	Period (.)	01381

QDATFMT	QTIMSEP	QCCSID[2]	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00935	CN	ZH_CN	CHS	GB SimpChin

[1] For Version 1 Release 2 and Release 3, the character set and code page is 00101 00037.

[2] This is the recommended QCCSID value if you want to use CDRA support.

Slovakian (Feature 2994)

The table shows the default system values for the Slovakian (Feature 2994) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	J	SKB	Slovak Koruna (K)	Hyphen (-)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	SK	SK_SK	SKY	ISO-8859-2 Latin 2

Slovenian (Feature 2911)

The table shows the default system values for the Slovenian (Feature 2911) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00959 00870	Blank	YGI	Slovenian Tolar (T)	Hyphen (-)	00912

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00870	SI	SL_SI	SLO	ISO-8859-2 Latin 2

Spanish (Feature 2931)

The table shows the default system values for the Spanish (Feature 2931) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00284	J	SPB	Dollar ($)	Slash (/)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	00284	ES	ES_ES	ESP	ISO-8859-1 Latin 1

Swedish (Feature 2937)

The table shows the default system values for the Swedish (Feature 2937) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
00697 00278	J	SWB	Dollar ($)	Hyphen (-)	00819

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Period (.)	00278	SE	SV_SE	SVE	ISO-8859-1 Latin 1

Thai (Feature 2972)

The table shows the default system values for the Thai (Feature 2972) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01279 00838	Blank	THB	Baht sign	Slash (/)	01066

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	09030	TH	TH_TH	THA	IBM 874

Traditional Chinese (Feature 2987)

The table shows the default system values for the Traditional Chinese (Feature 2987) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01175 00037	Blank	TAB	Dollar ($)	Slash (/)	00950

QDATFMT	QTIMSEP	QCCSID[1]	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
YMD	Colon (:)	00937	TW	ZH_TW	CHT	BIG5
[1] This is the recommended QCCSID value if you want to use CDRA support. For all NLVs, the default QCCSID value is 65535.						

Turkish (Feature 2956)

The table shows the default system values for the Turkish (Feature 2956) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
01152 01026	J	TKB	Dollar ($)	Slash (/)	00920

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	01026	TR	TR_TR	TRK	ISO-8859-9 Turkish

Vietnamese (Feature 2905)

The table shows the default system values for the Vietnamese (Feature 2905) national language version.

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP	Internet CCSID
1336 1130	J	VNB	DONG	Slash (/)	1258

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLOCALE	QLANGID	Client encoding nomenclature
DMY	Colon (:)	01130	VN	VI_VN	VNM	Windows® 1258

System values for other languages with no national language version

Some of the system values are associated with languages and countries that do not have a national language version. You should set these values immediately after initially installing the i5/OS operating system.

You must also change the message, CPX8416, in the QCPFMSG message file in library QSYS to reflect the changed system values.

- Afrikaans (South Africa)
- Australian English (Australia)
- Byelorussia (Belarus)
- Irish Gaelic (Ireland)
- Serbian Latin (Serbia)
- Spanish (Argentina)
- UK English (United Kingdom)

Afrikaans (South Africa)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
00697 00037	J	USB	rand	hyphen (-)

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
YMD	colon (:)	00037	ZA	AFR

Australian English (Australia)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
00697 00037	J	USB	dollar	slash (/)

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
DMY	period (.)	00037	AU	ENA

Byelorussia (Belarus)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
01150 01025		RUB		

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
		01025	BY	BEL

Irish Gaelic (Ireland)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
00697 00285	J	UKB	punt	slash (/)

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
DMY	colon (:)	00285	IE	GAE

Serbian Latin (Serbia)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
00859 00870	blank	YGI		

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
		00870	SQ	SRL

Spanish (Argentina)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
00697 00284	J	SSB	Austral ($)	slash (/)

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
DMY	period (.)	00284	AR	ESP

UK English (United Kingdom)

QCHRID	QDECFMT	QKBDTYPE	QCURSYM	QDATSEP
00697 00285	J	UKB	pound	slash (/)

QDATFMT	QTIMSEP	QCCSID	QCNTRYID	QLANGID
DMY	colon (:)	00285	GB	ENG

Related concepts

"Default system values for national language versions" on page 255
Jobs and functions on the i5/OS operating system use system values as default values.

Related tasks

"Handling languages with no NLV support" on page 96
If you need to support a language that does not have a supported national language version, follow these steps.

Workstation controllers reference

Workstation controllers that attach to the System i platform can support a number of different languages simultaneously. The characteristics of multilingual support depend on the type of workstation controller.

The types of workstation controllers that can be attached to the System i platform are:

- Twinaxial
- ASCII
- Remote

Multilingual support for the System i workstation controller

There are limitations to the amount of support provided by a workstation controller in a multilingual support.

- The workstation must be able to transmit and receive data in the required language.
- Each workstation controller supports a maximum number of languages at the same time, regardless of the keyboard types. This maximum number is in addition to the U.S. English support.
  - Twinaxial workstation controllers support a maximum of 3 languages.
  - ASCII workstation controllers support a maximum of 14 national language versions, in addition to U.S. English.
- The number of keyboard types allowed on a workstation controller is dependent on the different keyboard types required.

Twinaxial workstation controller requirements

The local twinaxial workstation controller maps keyboard data received from a workstation into EBCDIC values corresponding to the keyboard ID configured for that workstation on the system.

Multiple languages are supported by using a separate keyboard mapping table to handle each unique combination of national language version and keyboard for each workstation attached to the system.

The amount of storage available within the twinaxial workstation controller for mapping tables is limited and restricts the total number of national language versions that the workstation controller can simultaneously support. Depending on the mix of languages and types of keyboards, the twinaxial workstation controller can support several different national language versions simultaneously (in addition to U.S. English, which is always available).

Keyboard types on the twinaxial workstation controller

The twinaxial workstation controller supports the following types of IBM keyboards:

- 5250 typewriter keyboard
- 5250 data entry keyboard
- 122-key typewriter keyboard
- 122-key data entry keyboard
- Enhanced keyboard

The sum of the computational factors for a language and keyboard type must not exceed 22 for the twinaxial workstation controller. The following table shows the computational factor for each language, KBDTYPE parameter, and keyboard type.

Each workstation controller supports a maximum number of languages at the same time, regardless of the keyboard types. This maximum number is in addition to the US English support.

Note: The actual number of keyboards of each type does not have any impact on the sum. For example, the computational factor is 3, whether there are 1 or 30 enhanced keyboards running Austrian/German.

Language and keyboard computational factor table

1. Identify the language down the first column of the table.
2. Identify the keyboard type across the row for the selected language.
3. Record the computational factor listed for each keyboard type.
4. Repeat the first three steps for all the required keyboard types.
5. Add the computational factor for all required keyboard types.

a. If the sum of all computational factors does not exceed 22, then the twinaxial workstation controller can support all the required keyboard types for the languages.

b. If the sum of the computational factors exceeds 22, then the twinaxial workstation controller might not support all the required keyboard types for the languages.

Language	KBDTYPE	5250 typewriter keyboard	5250 data entry keyboard	122-key typewriter keyboard	122-key data entry keyboard	Enhanced keyboard
Albanian	ALI	N/A	N/A	1	N/A	3
Arabic	CLB	2	N/A	2	N/A	2.5
Austrian / German	AGB	1	1	1	N/A	3
Austrian / German MNCS	AGI	1	1	1	N/A	3
Belgian Dutch MNCS	BLI	1	1	1	N/A	3
Brazilian Portuguese	BRB	1	1	1	N/A	3
Bulgarian	BGB	N/A	N/A	N/A	N/A	3
Canadian French	CAB	1	1	1	1	3
Canadian French MNCS	CAI	1	1	1	1	3
Croatian	YGI	1	1	1	N/A	3
Cyrillic	CYB	2	N/A	2	N/A	2.5
Czech	CSB	N/A	N/A	N/A	N/A	3
Danish	DMB	1	1	1	N/A	3
Danish MNCS	DMI	1	1	1	N/A	3
Estonia	ESB	N/A	N/A	N/A	N/A	1.5
Finnish / Swedish	FNB	1	1	1	N/A	3
Finnish / Swedish MNCS	FNI	1	1	1	N/A	3
French (Azerty)	FAB	1	1	1	N/A	3
French (Azerty) MNCS	FAI	1	1	1	N/A	3
French (Qwerty)	FQB	1	1	N/A	N/A	N/A
French (Qwerty) MNCS	FQI	1	1	N/A	N/A	N/A
Greek	GNB	2	N/A	2	N/A	2.5
Hebrew	NCB	2	N/A	2	N/A	2.5
Hungarian	HNB	N/A	N/A	1	N/A	3
Icelandic	ICB	1	N/A	1	N/A	3

Language	KBDTYPE	5250 typewriter keyboard	5250 data entry keyboard	122-key typewriter keyboard	122-key data entry keyboard	Enhanced keyboard
Icelandic MNCS	ICI	1	N/A	1	N/A	3
International	INB	1	1	N/A	N/A	N/A
International MNCS	INI	1	1	N/A	N/A	N/A
Iran (Farsi)	IRB	N/A	N/A	2	N/A	2.5
Italian	ITB	1	1	1	N/A	3
Italian MNCS	ITI	1	1	1	N/A	3
Japanese - English	JEB	1	1	N/A	N/A	N/A
Japanese - English MNCS	JKB	1	1	N/A	N/A	N/A
Japanese - Kanji and Katakana	JKB	N/A	N/A	N/A	N/A	2.5
Japanese - Kanji and US English	JUB	N/A	N/A	1	N/A	N/A
Japanese Katakana	KAB	2	2	2	2	2.5
Japanese Latin Extended	JPB	1	1	N/A	N/A	N/A
Korean	KOB	1	1	1	1	1
Latin 2	ROB	1	N/A	1	N/A	1.5
Latvia	LVB	N/A	N/A	N/A	N/A	2.5
Lithuania	LTB	N/A	N/A	N/A	N/A	2.5
Macedonian	MKB	N/A	N/A	N/A	N/A	2.5
Netherlands Dutch	NEB	1	1	1	1	3
Netherlands Dutch MNCS	NEI	1	1	1	1	3
Norwegian	NWB	1	1	1	1	3
Norwegian MNCS	NWI	1	1	1	1	3
Polish	PLB	N/A	N/A	1	N/A	3
Portuguese	PRB	1	1	1	N/A	3
Portuguese MNCS	PRI	1	1	1	N/A	3
Romanian	RMB	1	1	1	N/A	3
Russian	RUB	N/A	N/A	N/A	N/A	3
Serbian (Latin)	YGI	1	1	1	N/A	3
Serbian (Cyrillic)	SQB	1	1	1	N/A	3

Language	KBDTYPE	5250 typewriter keyboard	5250 data entry keyboard	122-key typewriter keyboard	122-key data entry keyboard	Enhanced keyboard
Simplified Chinese	RCB	1	1	1	1	N/A
Slovakian	SKB	N/A	N/A	1	N/A	3
Slovenian	YGI	1	1	1	N/A	3
Spanish	SPB	1	1	1	N/A	3
Spanish MNCS	SPI	1	1	1	N/A	3
Spanish Speaking	SSB	1	1	1	1	3
Spanish Speaking MNCS	SSI	1	1	1	1	3
Swedish	SWB	1	1	1	N/A	3
Swedish MNCS	SWI	1	1	1	N/A	3
Swiss / French MNCS	SFI	1	N/A	1	N/A	3
Swiss / German MNCS	SGI	1	N/A	1	N/A	3
Thai	THB	N/A	N/A	N/A	N/A	2.5
Traditional Chinese	TAB	1	1	1	1	1
Turkish (QWERTY)	TKB	1	N/A	1	N/A	1.5
Turkish (F)	TRB	1	N/A	1	N/A	1.5
United Kingdom English	UKB	1	1	1	1	3
United Kingdom English MNCS	UKI	1	1	1	1	3
United States / Canada English	USB	See note.				
United States / Canada MNCS	USI	1	1	1	1	3

Note: Information about USB is not included, because it is always available and does not take up any additional space in the workstation controller.

ASCII workstation controller requirements

Like twinaxial devices, ASCII devices for different national language versions support different code pages. The ASCII workstation controller handles conversion of data back and forth between a particular EBCDIC code page for a language and an ASCII code page for that same language by using a set of mapping tables.

The ASCII workstation controller can simultaneously support 14 national language versions (in addition to U.S. English, which is always available).

The set of languages that can be selected for the ASCII workstation controller is a subset of the language types that can be selected for the twinaxial workstation controller. For the list of languages supported by the ASCII workstation controller, see "Multilingual support for the System i workstation controller" on page 272.

Keyboard types on an ASCII workstation controller

The maximum number of country and keyboard types on the ASCII workstation controller is 14. Depending on the display or printer device type, the controller might be able to fully support all graphic characters in a language. If the display or printer does not support all graphic characters in your language, you can use the workstation customization function to display and print unsupported graphic characters.

Languages supported without workstation customization

- Arabic
- Austrian/German
- Austrian/German MNCS
- Belgian Dutch MNCS
- Canadian French
- Canadian French MNCS
- Danish
- Danish MNCS
- Finnish/Swedish
- Finnish/Swedish MNCS
- French (Azerty)
- French (Azerty) MNCS
- Hebrew
- Italian
- Italian MNCS
- Norwegian
- Norwegian MNCS
- Portuguese
- Portuguese MNCS
- Spanish
- Spanish MNCS
- Spanish Speaking
- Spanish Speaking MNCS
- Swedish
- Swedish MNCS
- Swiss/French
- Swiss/French MNCS
- Swiss/German
- Swiss/German MNCS
- United Kingdom English
- United Kingdom English MNCS
- United States and Canada English
- United States and Canada English MNCS

Languages supported using workstation customization

Using the workstation customization functions, the following countries or languages can also be supported by the ASCII workstation controller.

- Brazilian Portuguese
- Croatian
- Cyrillic
- Czech
- Estonian
- Greek
- Hungarian
- Icelandic
- Icelandic MNCS
- Latin 2
- Latvian
- Lithuanian
- Polish
- Russian
- Slovakian
- Slovenian
- Turkish
- Thai
- Ukrainian

5394 remote workstation controller requirements

The 5394 remote workstation controller is a twinaxial workstation controller and uses conversion tables to map the data between the devices.

The 5394 remote workstation controller can support up to 4 MNCS languages at a time and only one language that is not a MNCS language. When changing languages, you must change the keyboard language code. Changing the keyboard language code is explained further in the *5394 Remote Control Unit Setup Guide* and the *5394 Remote Control Unit Introduction and Installation Planning* book.

5494 remote workstation controller requirements

The 5494 remote workstation controller has 4 ports for twinaxial workstations, a port for communication networks, and on the Model 002, a port for attaching to a token-ring network.

The 5494 remote workstation controller can support up to 4 MNCS languages at a time and only one language that is not a MNCS language. When changing languages, you must change the keyboard language code. Changing the keyboard language codes is explained further in the *5494 Remote Control Unit Planning Guide* and the *5494 Remote Control Unit User's Guide*.

Determining the number of supported keyboard types

To determine the number of keyboard types that the twinaxial or ASCII controllers can support, you must understand this information.

- A twinaxial or ASCII workstation controller can support several types of keyboards and languages.
- Each of the supported languages can be available on more than one of the supported keyboards.
- For each workstation controller, the size of the conversion table for each keyboard is different.

- The matrix for languages used on each supported keyboard type is called a keyboard and language computational factor.
- The sum of the computational factors for each required keyboard type determines whether all required keyboard types can be supported on one workstation controller.
- If the sum of the computational factors for the required keyboard types exceeds the maximum limit, the language for the first workstation that caused the overflow and any additional workstations set to U.S. English as the default.
- To recover from a keyboard-type overflow, you can do one of the following tasks:
  - Attach the workstation causing the overflow to a second workstation controller.
  - On the same workstation controller, configure the workstations causing the overflow to some other keyboard type that reduces the sum of the computational factor to within the maximum limits.

Related concepts

"Keyboard layouts"
These keyboard layout samples are provided for your information. The special-character keyboard set is available only with the enhanced keyboard.

Keyboard reference information

This reference information includes keyboard layouts, SBCS keyboard and display part numbers, the special-character keyboard set, and the keyboard types and SBCS code pages for each national language.

Keyboard layouts

These keyboard layout samples are provided for your information. The special-character keyboard set is available only with the enhanced keyboard.

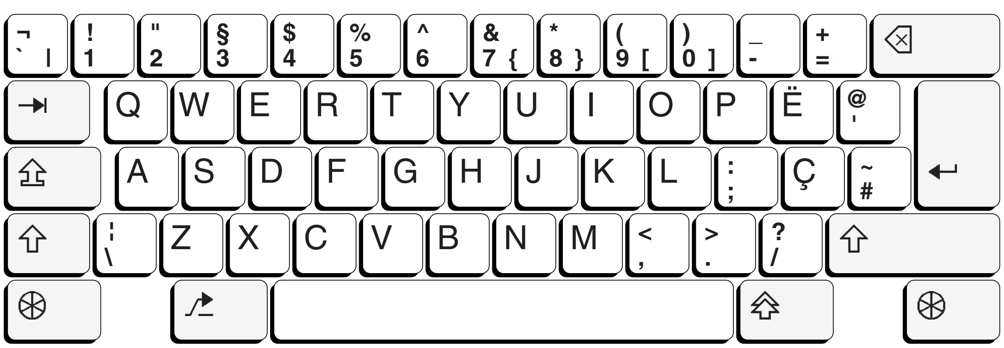
Related concepts

"Determining the number of supported keyboard types" on page 277
To determine the number of keyboard types that the twinaxial or ASCII controllers can support, you must understand this information.

"Workstation considerations" on page 34
In a multilingual environment, different workstations support different languages on the same system.

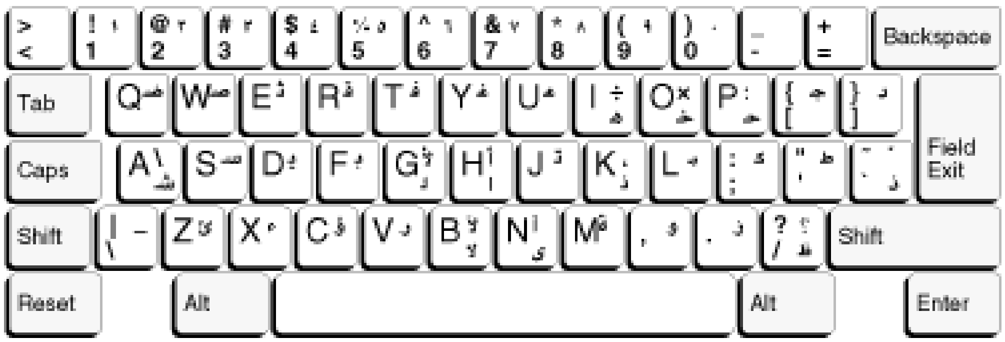
Albanian IBM Enhanced Keyboard:

The figure shows the Albanian IBM Enhanced Keyboard.



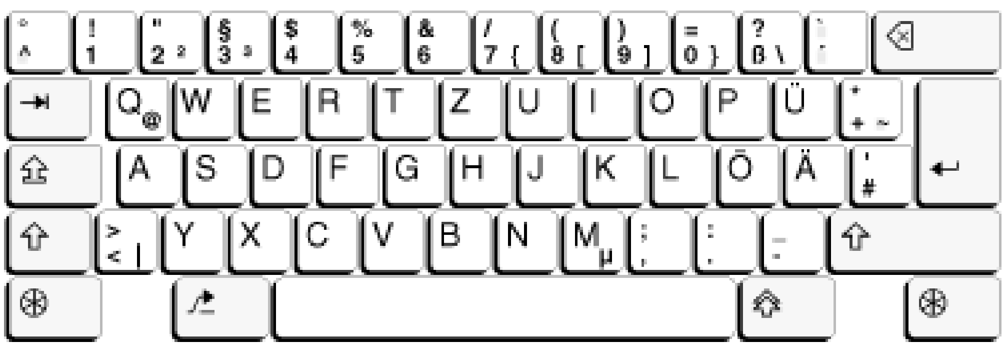
Arabic IBM Enhanced Keyboard:

The figure shows the Arabic IBM Enhanced Keyboard.



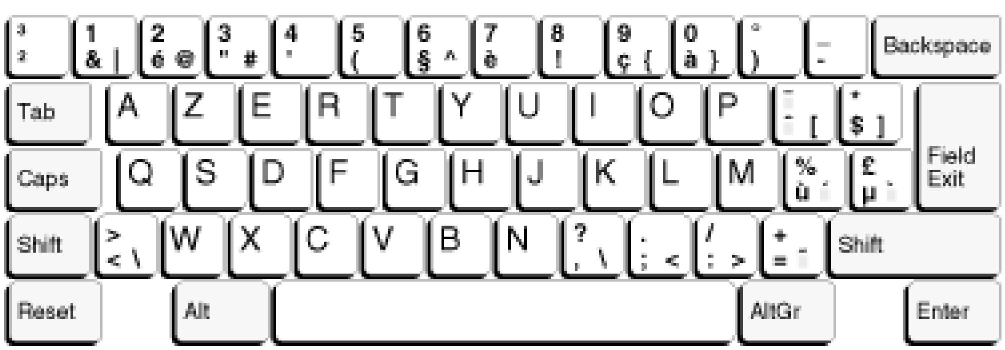
Austrian German IBM Enhanced Keyboard:

The figure shows the Austrian German IBM Enhanced Keyboard.



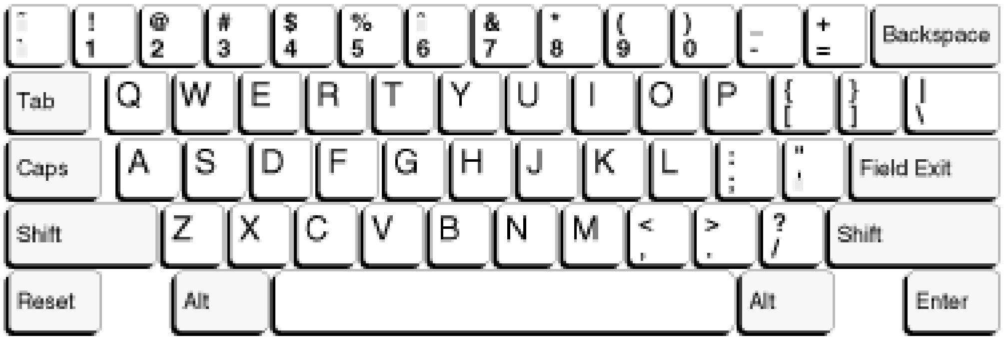
Belgian Multinational IBM Enhanced Keyboard:

The figure shows the Belgian Multinational IBM Enhanced Keyboard.



Brazilian Portuguese IBM Enhanced Keyboard:

The figure shows the Brazilian Portuguese IBM Enhanced Keyboard.



This U.S. keyboard is used to enter Brazilian Portuguese characters. Accented characters for use in Brazil are generated using key combinations as shown in the following table. To produce the accented characters, press the key and character at the same time.

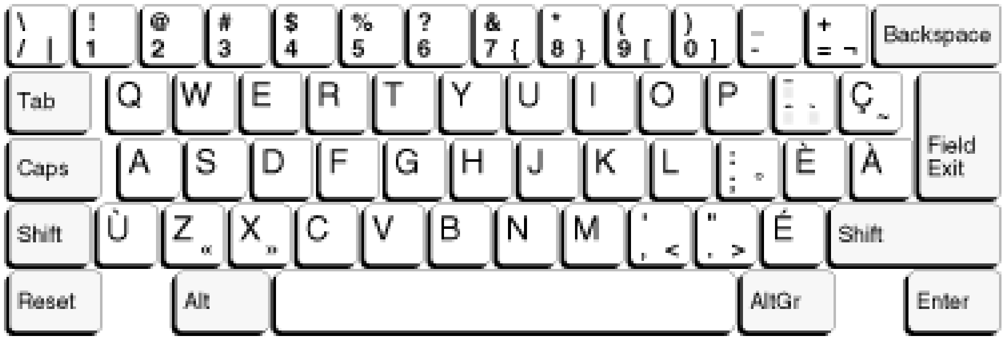
Accent	Key	Valid characters	Example result
Accent acute ´	′ (apostrophe)	a, e, i, o, u, A, E, I, O, U	á
Accent grave `	` (accent grave)	a, e, i, o, u, A, E, I, O, U	à
Diaeresis ¨	″ (quotation mark)	a, e, i, o, u, A, E, I, O, U	ä
Tilde ~	~ (tilde)	a, o, n, A, O, N	ã
Circumflex ^	^ (caret)	a, e, i, o, u, A, E, I, O, U	â

Bulgarian Cyrillic IBM Enhanced Keyboard:

The figure shows the Bulgarian Cyrillic IBM Enhanced Keyboard.

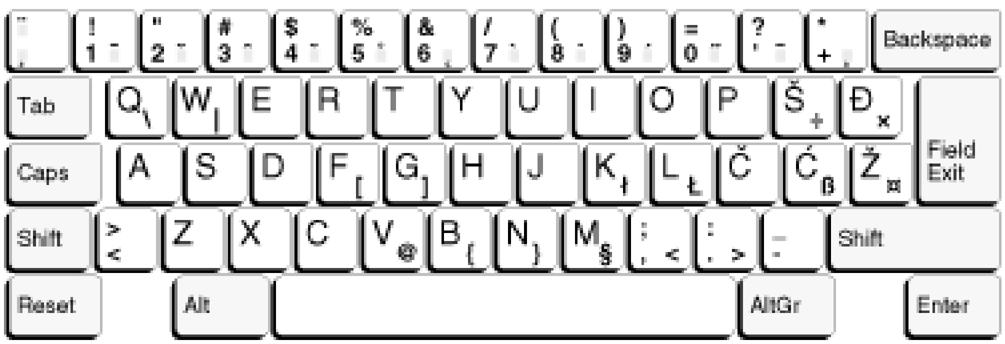
Canadian French IBM Enhanced Keyboard:

The figure shows the Canadian French IBM Enhanced Keyboard.



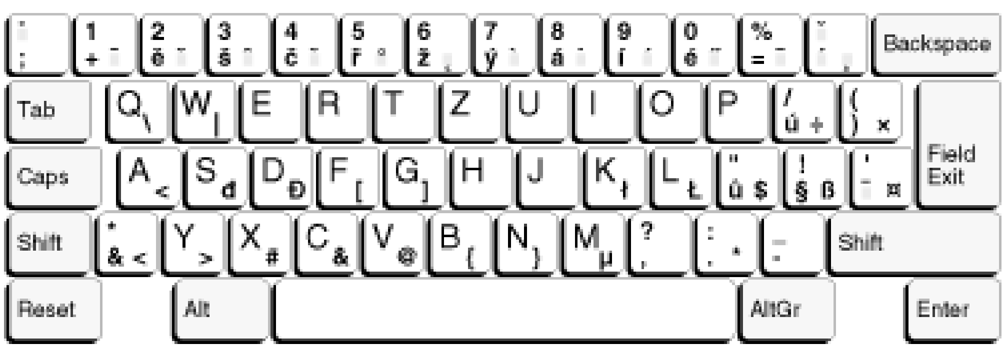
Croatian IBM Enhanced Keyboard:

The figure shows the Croatian IBM Enhanced Keyboard.



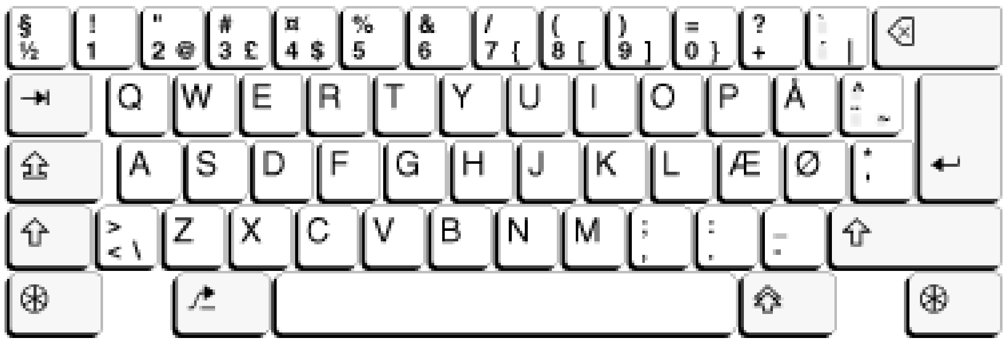
Czech IBM Enhanced Keyboard:

The figure shows the Czech IBM Enhanced Keyboard.



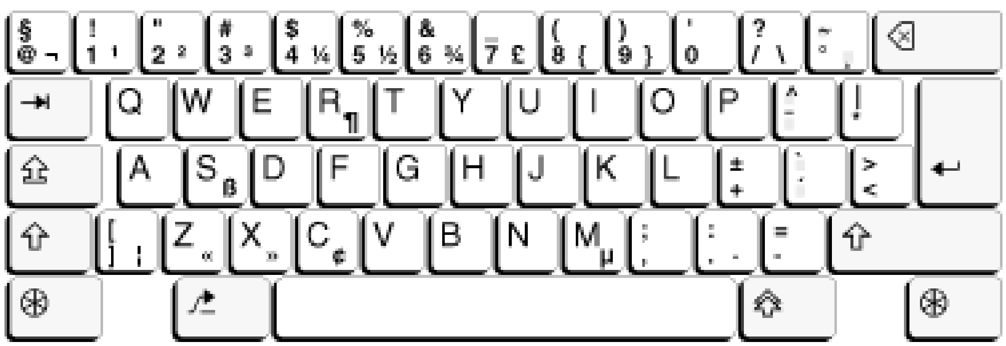
Danish IBM Enhanced Keyboard:

The figure shows the Danish IBM Enhanced Keyboard.



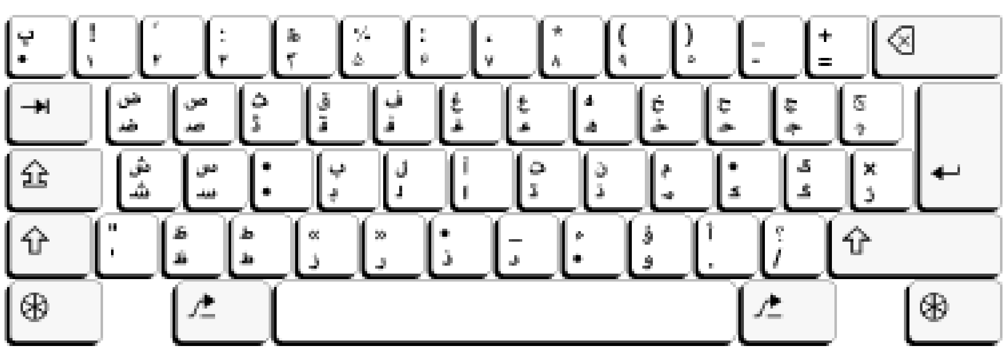
Dutch IBM Enhanced Keyboard:

The figure shows the Dutch IBM Enhanced Keyboard.



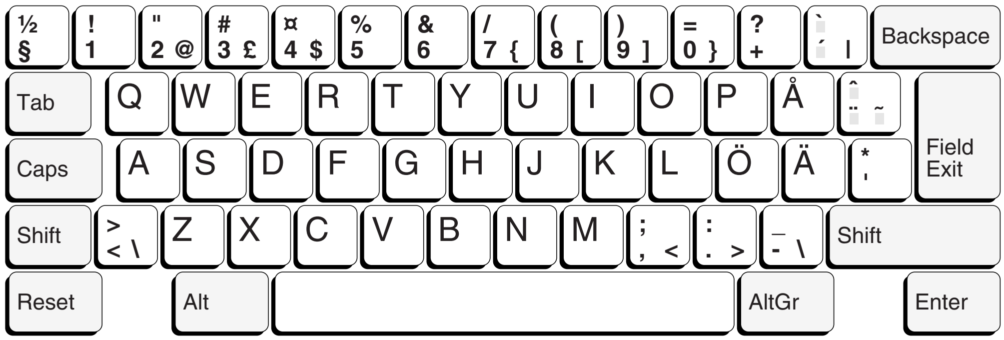
Farsi IBM Enhanced Keyboard:

The figure shows the Farsi IBM Enhanced Keyboard.



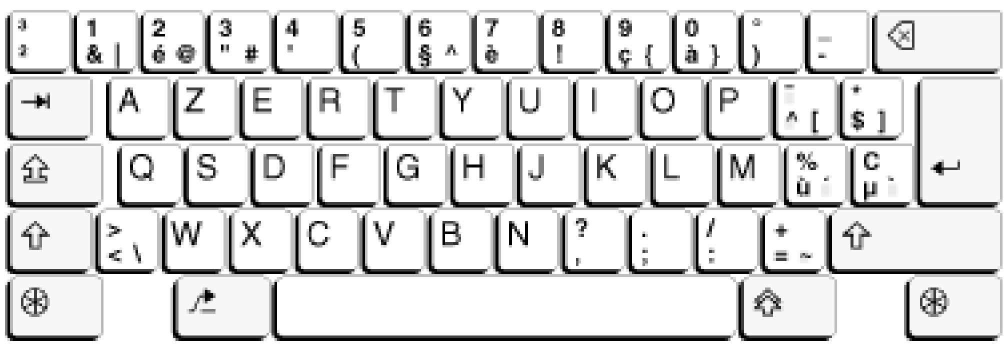
Finnish/Swedish IBM Enhanced Keyboard:

The figure shows the Finnish/Swedish IBM Enhanced Keyboard.



French (AZERTY) IBM Enhanced Keyboard:

The figure shows the French IBM Enhanced Keyboard.

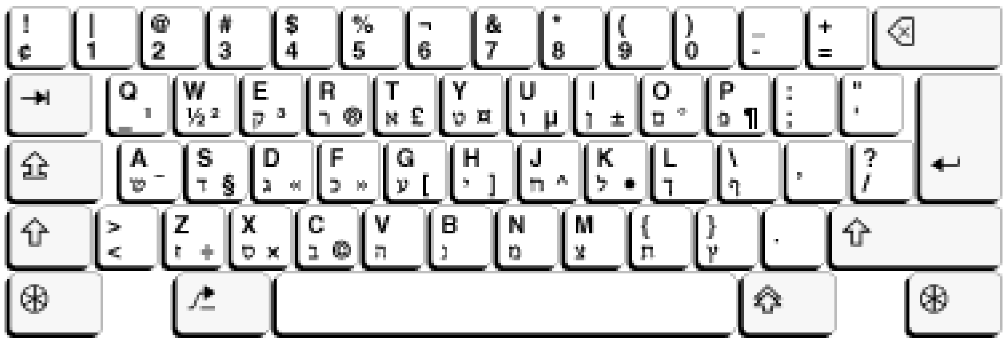


Greek IBM Enhanced Keyboard:

The figure shows the Greek IBM Enhanced Keyboard.

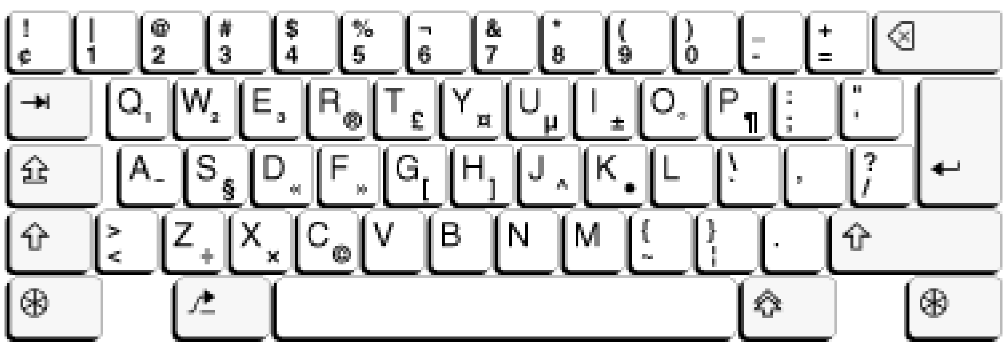
Hebrew IBM Enhanced Keyboard:

The figure shows the Hebrew IBM Enhanced Keyboard.



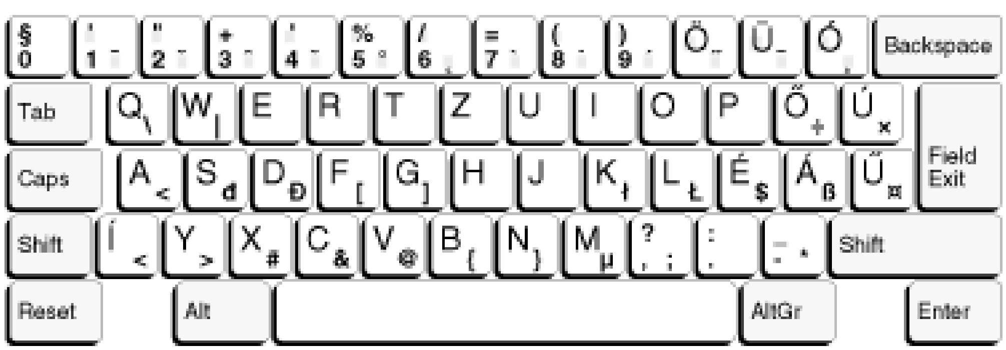
Hebrew, Latin IBM Enhanced Keyboard:

The figure shows the Hebrew, Latin IBM Enhanced Keyboard.



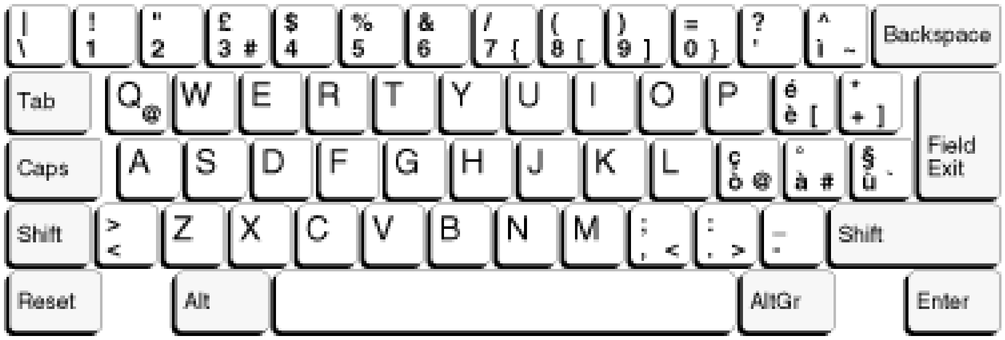
Hungarian IBM Enhanced Keyboard:

The figure shows the Hungarian IBM Enhanced Keyboard.



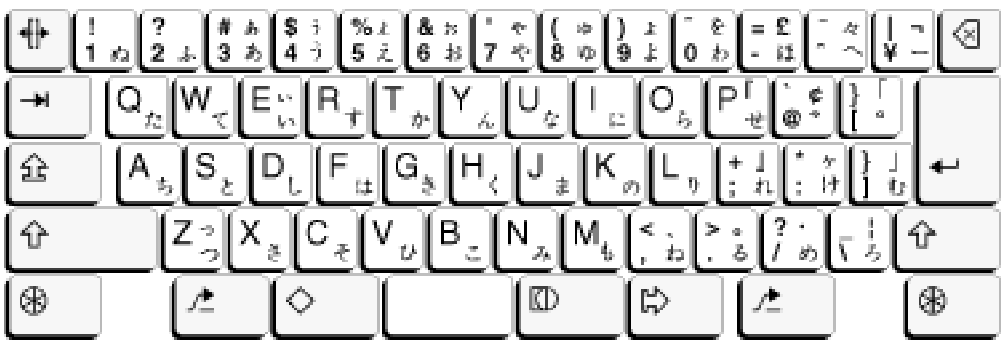
Italian IBM Enhanced Keyboard:

The figure shows the Italian IBM Enhanced Keyboard.



Japanese IBM Enhanced Keyboard:

The figure shows the Japanese IBM Enhanced Keyboard.

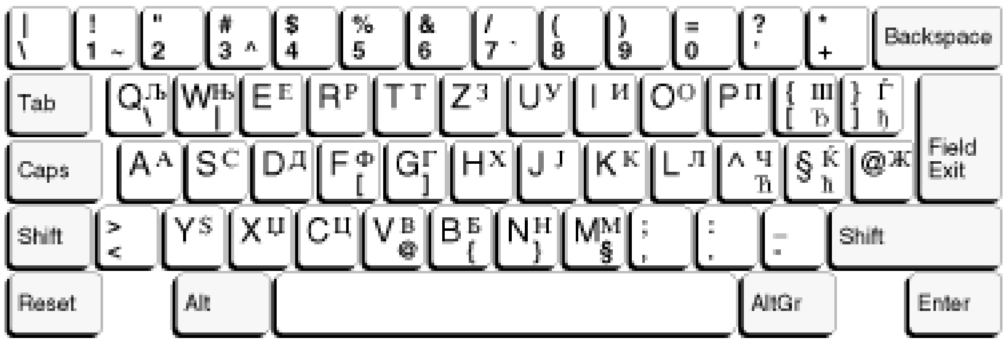


Korean IBM Enhanced Keyboard:

The figure shows the Korean IBM Enhanced Keyboard.

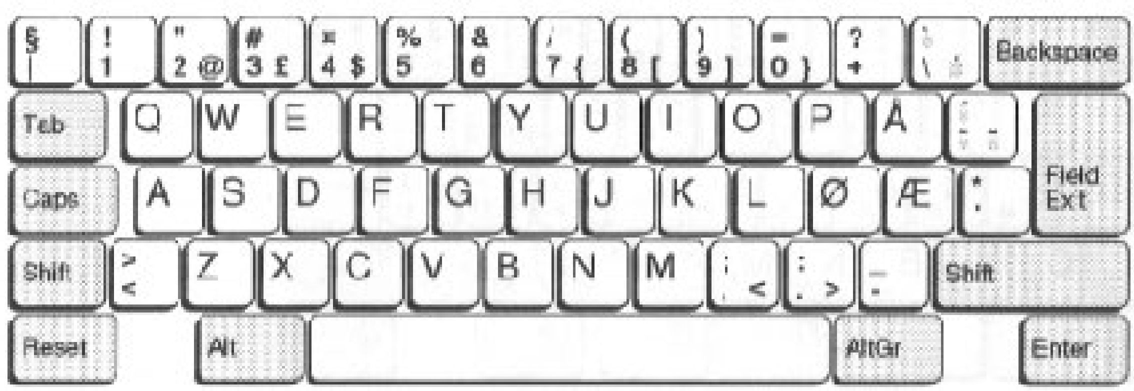
Macedonian IBM Enhanced Keyboard:

The figure shows the Macedonian IBM Enhanced Keyboard.



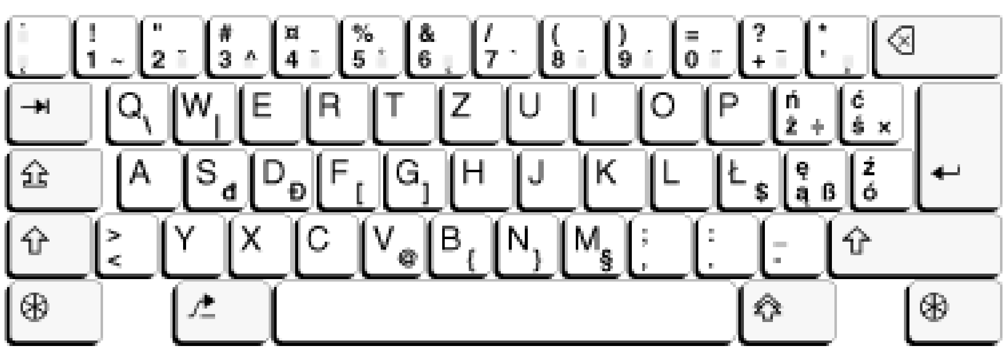
Norwegian IBM Enhanced Keyboard:

The figure shows the Norwegian IBM Enhanced Keyboard.



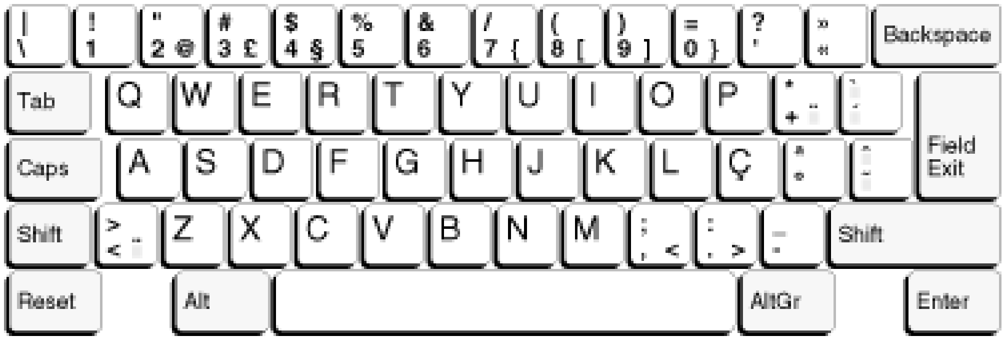
Polish IBM Enhanced Keyboard:

The figure shows the Polish IBM Enhanced Keyboard.



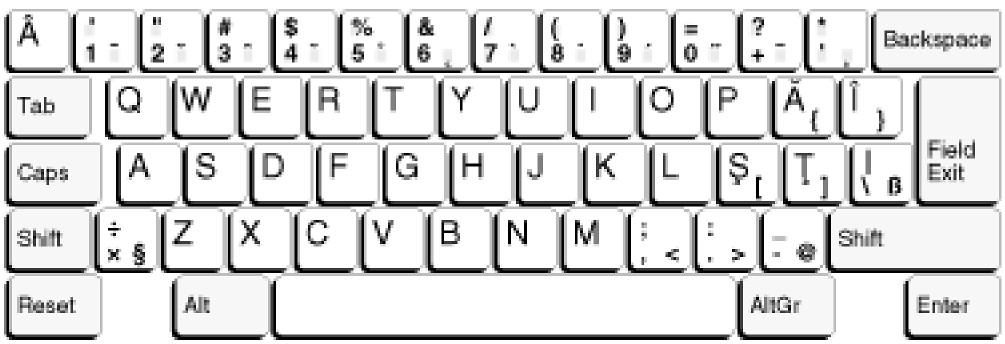
Portuguese IBM Enhanced Keyboard:

The figure shows the Portuguese IBM Enhanced Keyboard.



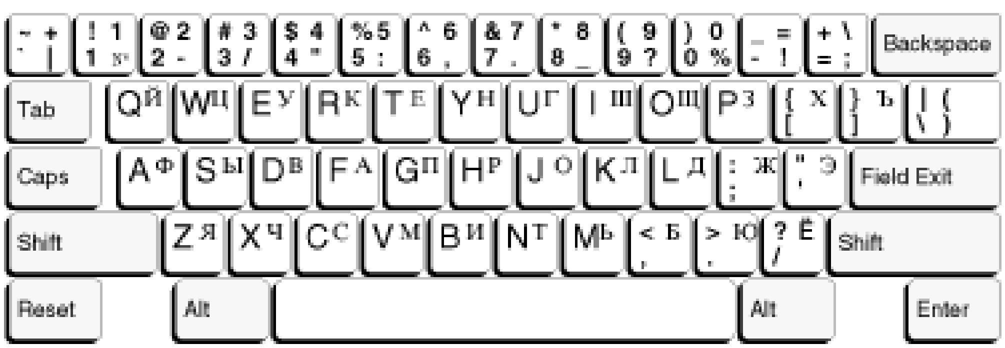
Romanian IBM Enhanced Keyboard:

The figure shows the Romanian IBM Enhanced Keyboard.



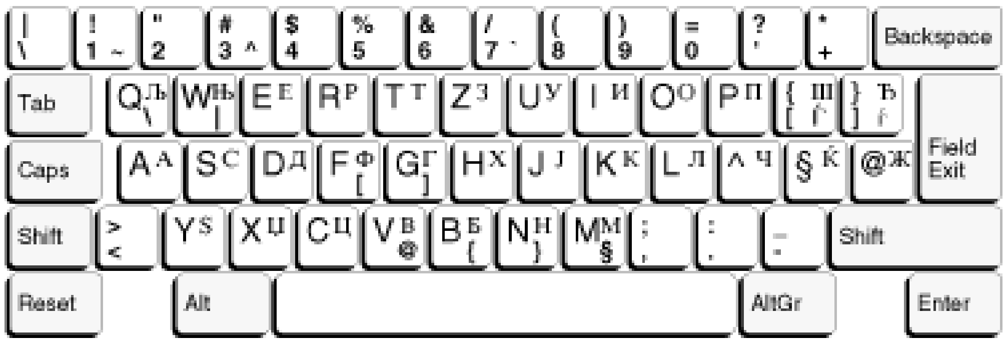
Russian IBM Enhanced Keyboard:

The figure shows the Russian IBM Enhanced Keyboard.



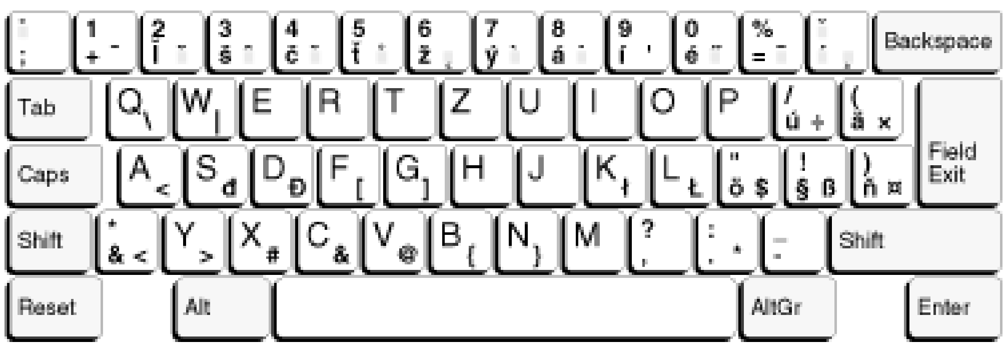
Serbian Cyrillic IBM Enhanced Keyboard:

The figure shows the Serbian Cyrillic IBM Enhanced Keyboard.



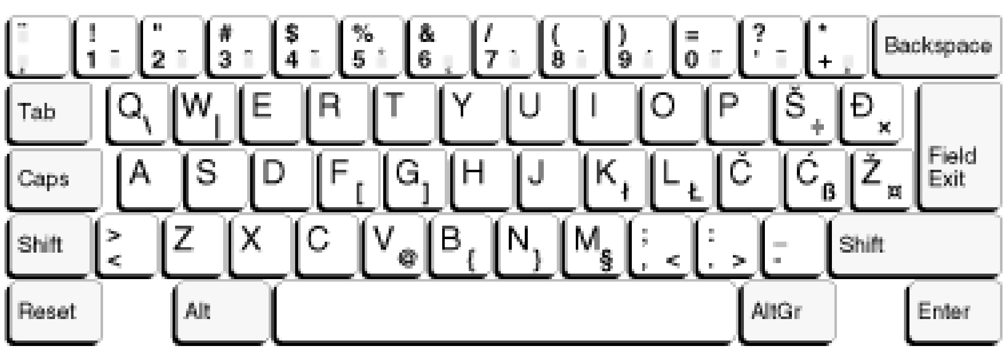
Slovakian IBM Enhanced Keyboard:

The figure shows the Slovakian IBM Enhanced Keyboard.



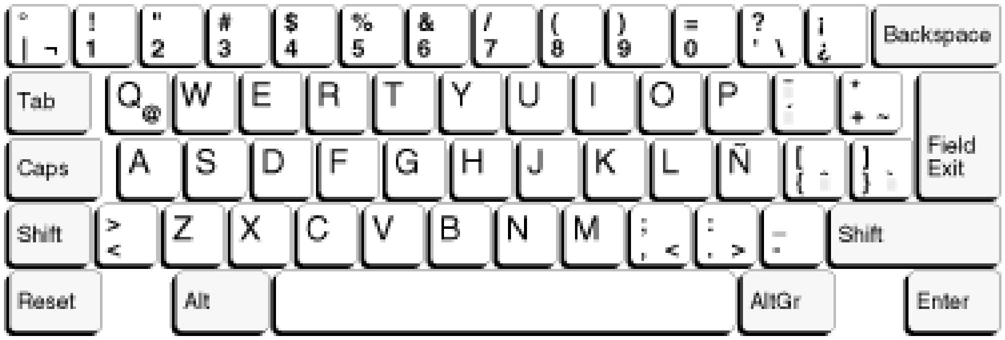
Slovenian IBM Enhanced Keyboard:

The figure shows the Slovenian IBM Enhanced Keyboard.



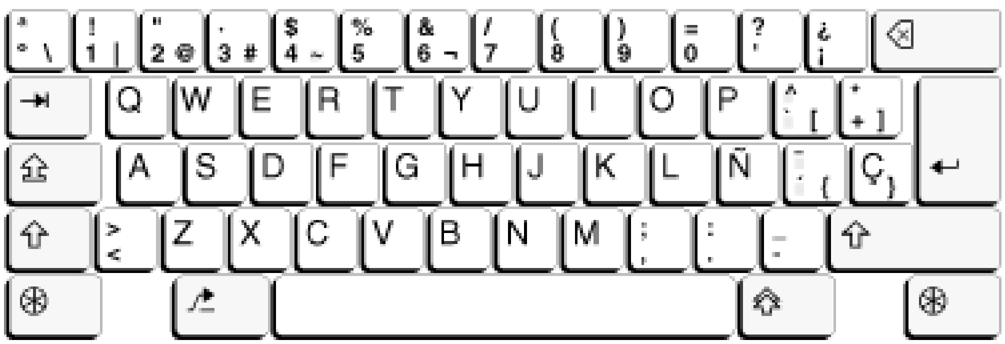
Spanish-Speaking IBM Enhanced Keyboard:

The figure shows the Spanish-Speaking IBM Enhanced Keyboard.



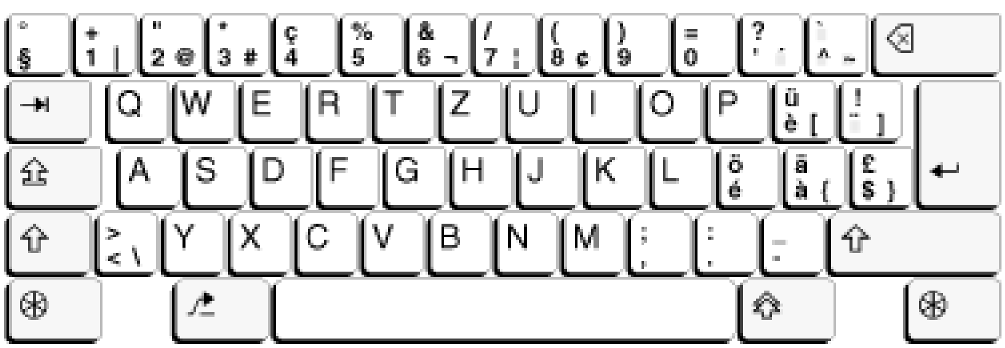
Spanish IBM Enhanced Keyboard:

The figure shows the Spanish IBM Enhanced Keyboard.



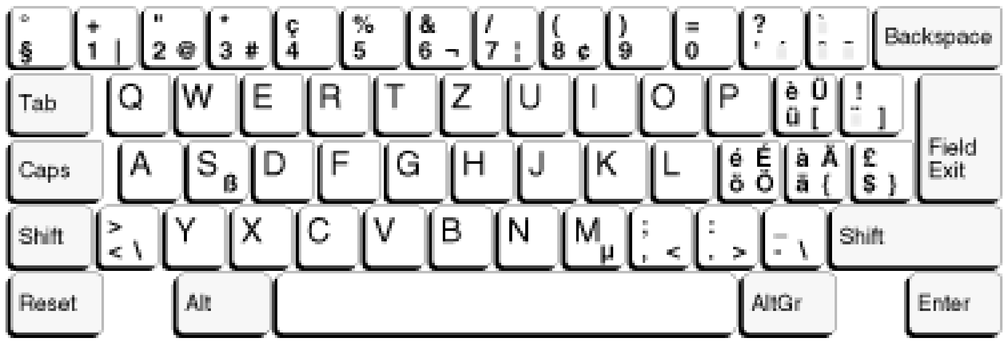
Swiss-Bilingual-French IBM Enhanced Keyboard:

The figure shows the Swiss-Bilingual-French IBM Enhanced Keyboard.



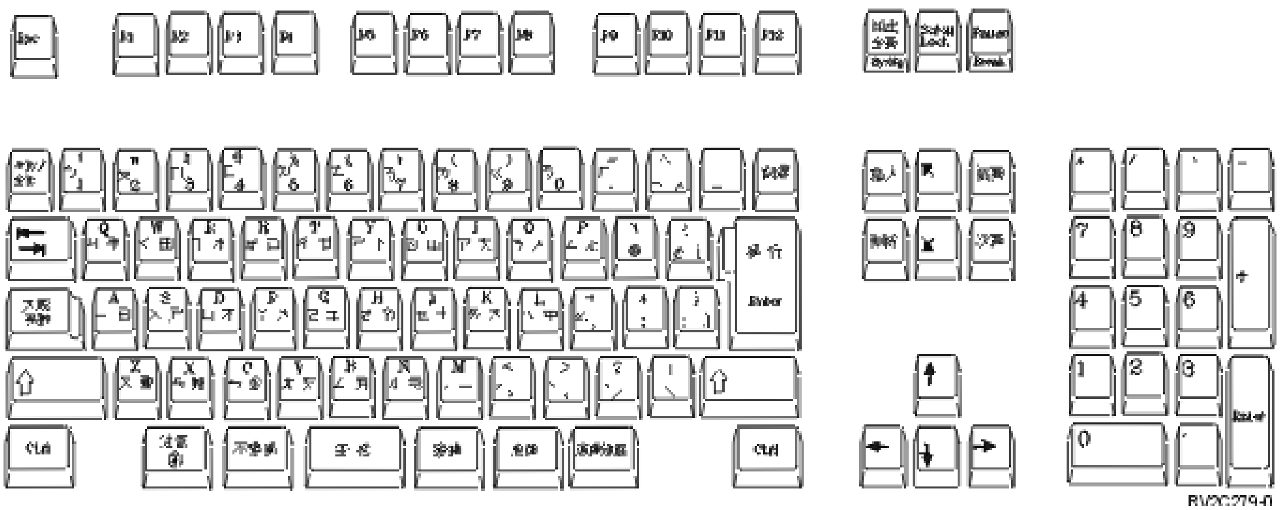
Swiss-Bilingual-German IBM Enhanced Keyboard:

The figure shows the Swiss-Bilingual-German IBM Enhanced Keyboard.



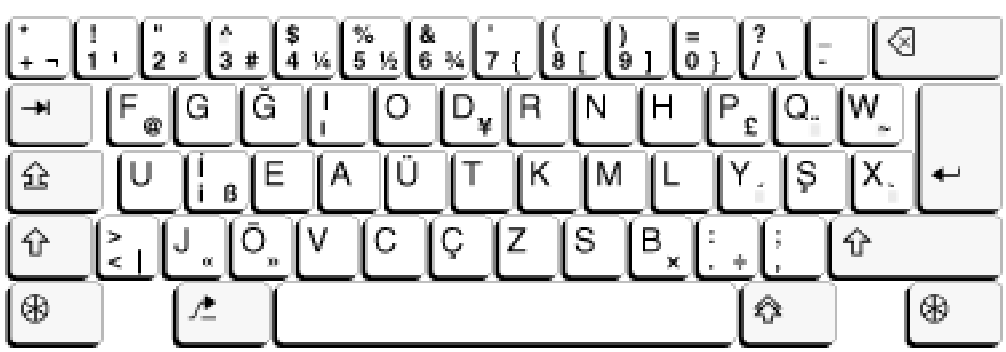
Traditional Chinese IBM Enhanced Keyboard:

The figure shows the Traditional Chinese IBM Enhanced Keyboard.



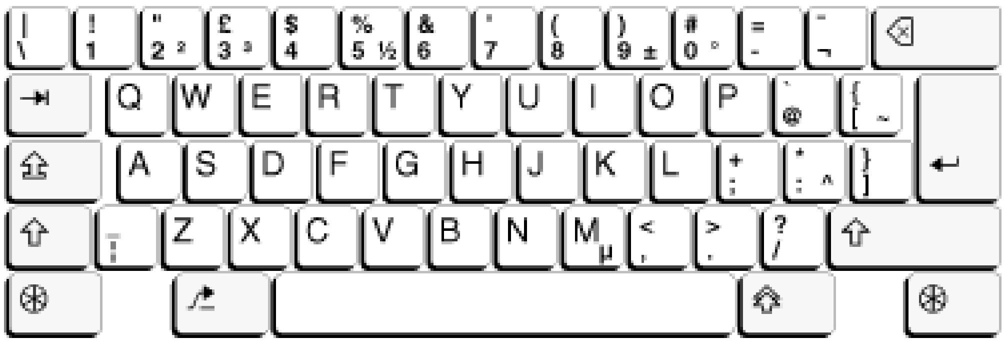
Turkish IBM Enhanced Keyboard:

The figure shows the Turkish IBM Enhanced Keyboard.



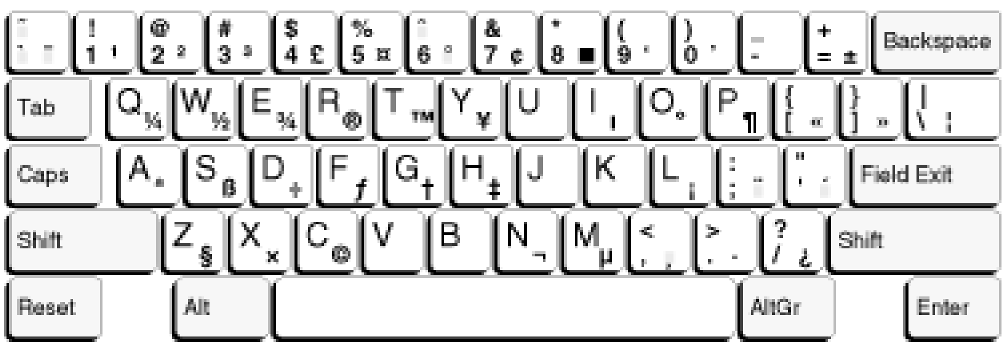
U.K. English IBM Enhanced Keyboard:

The figure shows the U.K. English IBM Enhanced Keyboard.



U.S. English IBM Enhanced Keyboard:

The figure shows the U.S. English IBM Enhanced Keyboard.



SBCS keyboard and display part numbers by language

These tables list the part numbers of the SBCS displays and keyboards that should be used for each language or country supported by IBM System i products.

Users of a particular national language version can verify if they have the correct display and keyboard by checking part numbers. The keyboard numbers are printed on the bottom of the keyboards. If the keyboard or display for that language is not correct, the characters displayed might not be correct.

- “3477 and 3476 keyboard and display part numbers” on page 292
- “3486, 3487, 3488 Model V, and 3489 keyboard and display part numbers by language” on page 294
- “Keyboard support for other devices” on page 301

Determining display part numbers

To determine the part number for the model-unique Licensed Internal Code of the display, use the following procedure:

1. Press and hold any key on the keyboard while powering the display on.
2. Move the cursor to

   `Test Workstation`

   and press the Enter key.

Note: The text on the display contains a 7-digit number. This is the part number.

3477 and 3476 keyboard and display part numbers

The following table lists the part numbers for the 3477 and 3476 displays.

General list of displays: Most languages use the following set of displays. Differences are noted in the table.

- 38F5835
- 38F5843
- 38F5845
- 56F8934
- 65F2987
- 65F2995
- 79F2020
- 79F2029
- 79F2032
- 95F4167
- 95F4171
- 95F4174
- 64F9705
- 95F5908
- 79F5064
- 95F5911
- 79F7019
- 79F7020
- 79F7022
- 79F7025
- 95F4144
- 95F4146
- 07G2170
- 95F5941
- 95F5943
- 07G2172
- 07G2174
- 07G2176
- 38F7998
- 38F7999
- 23F1574
- 23F1585
- 56F9556
- 56F9557
- 56F9604

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
Arabic	CLB	1394332	1394436	38F5818, 56F8919, 65F2991, 65F2999, 79F2045, 79F2047, 95F4178, 95F5896, 79F7084, 79F7100, 95F5971, 95F5983, 23F1581, 56F9585, 38F8008
Austrian/German	AGB	1394312	1394416	See general list
Austrian/German MNCS	AGI	1394312	1394416	See general list
Belgian MNCS	BLI	1394313	1394417	See general list
Brazilian Portuguese	BRB	1394319	1394423	See general list
Canadian French	CAB	1395662	1395567	See general list
Canadian French MNCS	CAI	1395662	1395567	See general list
Cyrillic	CYB	1394329	1394433	38F5824, 56F8925, 65F2993, 65F3001,79F5045, 79F5047, 95F5893, 95F5905, 79F7097, 95F4108, 95F5980, 07G2167, 23F1584, 56F9596
Danish	DMB	1394314	1394418	See general list
Danish MNCS	DMI	1394314	1394418	See general list
Finnish/Swedish	FNB	1394315	1394419	See general list
Finnish/Swedish MNCS	FAB	1394315	1394419	See general list
French (Azerty)	FAB	1394316	1394420	See general list
French (Azerty) MNCS	FAI	1394316	1394420	See general list
Greek (Code Page 00875)	GNB	1396767	1396768	56F9587, 79F7039, 79F7048, 79F7040, 79F7049, 95F5922, 95F5925, 95F5923, 95F5926
Greek (Code Page 00423)	GKB	1394325	1394429	56F9587, 23F1582, 56F8958, 56F8960, 56F8959, 56F8961, 79F2049, 79F5043, 79F2050, 79F5044, 95F5885, 95F5899, 95F5886, 95F5900

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
Hebrew	NCB	1394331	1394435	23F1583, 56F9595, 79F7094, 95F4105, 95F5977, 07G2164, 38F5822, 56F8923, 65F2989, 65F2997, 79F2041, 79F2043, 95F5889, 95F5902
Icelandic	ICB	1394330	1394434	38F5820, 56F8921, 56F8958, 56F8960, 79F5043, 79F2049, 95F5885, 95F5899, 23F1582, 56F9587, 79F7087, 95F4102, 95F5974, 07G2161
Icelandic MNCS	ICI	1394330	1394434	Same as Icelandic
Italian	ITB	1394317	1394421	See general list
Italian MNCS	ITI	1394317	1394421	See general list
Japanese Katakana	KAB	1395664	1395669	See general list
Latin 2	ROB	1394328	1394432	38F5824, 56F8925, 65F2993, 65F3001,79F5045, 79F5047 95F5893, 95F5905, 79F7097, 95F4108, 95F5980, 07G2167, 23F1584, 56F9596
Dutch (Netherlands)	NEB		1394427	See general list
Dutch (Netherlands) MNCS	NEI		1394427	See general list
Norwegian	NWB	1394318	1394422	See general list
Norwegian MNCS	NWI	1394318	1394422	See general list
Portuguese	PRB	1394319	1394423	See general list
Portuguese MNCS	PRI	1394319	1394423	See general list
Spanish	SPB	1394320	1394424	See general list
Spanish MNCS	SPI	1394320	1394424	See general list
Spanish speaking	SSB	1395663	1395668	See general list
Spanish speaking MNCS	SSI	1395663	1395668	See general list
Swedish	SWB	1394315	1394419	See general list
Swedish MNCS	SWI	1394315	1394419	See general list
French (Switzerland) MNCS	SFI	1394321	1394425	See general list
German (Switzerland) MNCS	SGI	1394322	1394426	See general list
Thai	THB		1395670	56F9597
English (United Kingdom)	UKB	1394324	1394428	See general list
English (United Kingdom) MNCS	UKI	1394324	1394428	See general list
English (United States/Canada)	USB	1395661, 1395660	1395666, 1395665	See general list
English (United States/Canada) MNCS	USI	1394167	1394193	See general list

3486, 3487, 3488 Model V, and 3489 keyboard and display part numbers by language

The following table lists the part numbers of the 3486, 3487, 3488 Model V, and 3489 keyboards and displays that should be used for each language or country supported by IBM System i products. Blank entries in the keyboard columns mean that keyboard part numbers are not available at this time.

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
Arabic	CLB	1394332	1394436	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Austrian/German	AGB	1394312	1394416	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Austrian/German MNCS	AGI	1394312	1394416	Same as Austrian/German
Belgian MNCS	BLI	1394313	1394417	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Bulgarian	BGB		35G4741	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
French (Canada)	CAB	1395662	1395567	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
French (Canada) MNCS	CAI	1395662	1395567	Same as French (Canada)
Croatian	YGI	1394327	1394431	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Czech	CSB		35G4743	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Danish	DMB	1394314	1394418	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Danish MNCS	DMI	1394314	1394418	Same as Danish
Finnish/Swedish	FNB	1394315	1394419	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Finnish/Swedish MNCS	FNI	1394315	1394419	Same as Finnish/Swedish

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
French (Azerty)	FAB	1394316	1394420	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
French (Azerty) MNCS	FAI	1394316	1394420	Same as French (Azerty)
Greek (Code Page 00875)	GNB	1396767	1396768	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Greek (Code Page 00423)	GKB	1394325	1394429	Same as Greek (Code Page 00875)
Hebrew	NCB	1394331	1394435	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Hungarian	HNB		35G4745	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Farsi (Iran)	IRB	53G9084	53G9085	66G1721, 66G1727
Italian	ITB	1394317	1394421	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Italian MNCS	ITI	1394317	1394421	Same as Italian

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
Japanese-Katakana	KAB	1395664	1395669	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Macedonian	MKB		35G4740	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Dutch (Netherlands)	NEB		1394427	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Dutch (Netherlands) MNCS	NEI		1394427	Same as Dutch (Netherlands)
Norwegian	NWB	1394318	1394422	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Norwegian MNCS	NWI	1394318	1394422	Same as Norwegian
Polish	PLB		35G4746	66G1721, 66G1727
Portuguese	PRB	1394319	1394423	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
Portuguese MNCS	PRI	1394319	1394423	Same as Portuguese
Romanian	RMB		35G4747	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Russian	RUB		35G4742	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Serbian, Cyrillic	SQB		35G4740	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Serbian, Latin	YGI	1394327	1394431	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Slovakian	SKB		35G4744	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
Slovenian	YGI	1394327	1394431	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Spanish	SPB	1394320	1394424	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Spanish MNCS	SPI	1394320	1394424	Same as Spanish
Spanish Speaking	SSB	1395663	1395668	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Spanish Speaking MNCS	SSI	1395663	1395668	Same as Spanish Speaking
Swedish	SWB	1394315	1394419	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Swedish MNCS	SWI	1394315	1394419	Same as Swedish
French (Switzerland) MNCS	SFI	1394321	1394425	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727

Language	KBDTYPE parameter	1A (122-Key) keyboard	G keyboard	Displays
German (Switzerland) MNCS	SGI	1394322	1394426	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Thai	THB		1395670	07G8571, 06G5322, 06G5312
Turkish (F)	TRB	35G4748	35G4749	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
Turkish (QWERTY)	TKB	1394326	1394430	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
English (United Kingdom)	UKB	1394324	1394428	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
English (United Kingdom) MNCS	UKI	1394324	1394428	Same as English (United Kingdom)
English (United States/Canada)	USB	1395661, 1395660	1395666, 1395665	06G5310, 06G5311, 06G5312, 06G5313, 06G5314, 06G5315, 06G5316, 06G5320, 06G5321, 06G5322, 06G5323, 06G5324, 06G5325, 06G5326, 06G5330, 06G5331, 06G5332, 06G5333, 06G5334, 06G5335, 06G5340, 06G5341, 06G5342, 06G5343, 06G5344, 06G5345, 07G8611, 07G8567, 07G8568, 07G8569, 06G8570, 07G8571, 07G8572, 06G5333, 06G5336, 66G1721, 66G1727
English (United States/Canada) MNCS	USI	1394167	1394193	Same as United States/Canada

Keyboard support for other devices

- Enhanced G and 122-key keyboards are not available for 5251, 5291, and 5292 display stations. These display stations use only the F keyboard (similar to the 122-key keyboard).
- The 3180 display station supports the 122-key keyboard but not the enhanced keyboard. The 3179, 3196, and 3197 support the 122-keyboard and the enhanced keyboard.
- The 3486, 3487, 3488 Model V, and 3489 display stations support all languages listed in 3477 and 3476 keyboard and display part numbers (except for Thai) on all base levels of hardware. No checking of the part numbers for the display stations is necessary.

  Note: The 3488 Model V display station requires a special monitor to support some languages. The 3489 display station may also require a special monitor to support some languages.
- The 3486, 3487, 3488 Model V, and 3489 display stations do not need the correct language keyboard to show the code page for a language. The code page used by the display station matches what is specified in the configuration record. The 3488 Model V and 3489 display stations require a special monitor to support some languages.
- The 3488 Model H display station supports the following languages or countries on all levels of hardware. No checking of part numbers is necessary. The 3488 Model H display station does not need the correct language keyboard to show a code page for one of the supported languages. The code page used must match the one in the configuration record as long as it is among the supported languages.

  Austrian/German
  Belgian
  French (Canada)
  Danish
  Dutch
  Finnish/Swedish
  French (with AZERTY keyboard)
  Italian
  Norwegian
  Portuguese
  Spanish
  Swiss-French
  Swiss-German
  U.K. English
  U.S. English
- The 3476, 3486, 3487, 3488, and 3489 display stations support the U.S. data entry keyboard (part numbers 35G4750 and 35G4751).
- The 3488 Model V and 3489 display stations also support the Brazilian Portuguese G keyboard (part number 63F1403).

Related concepts

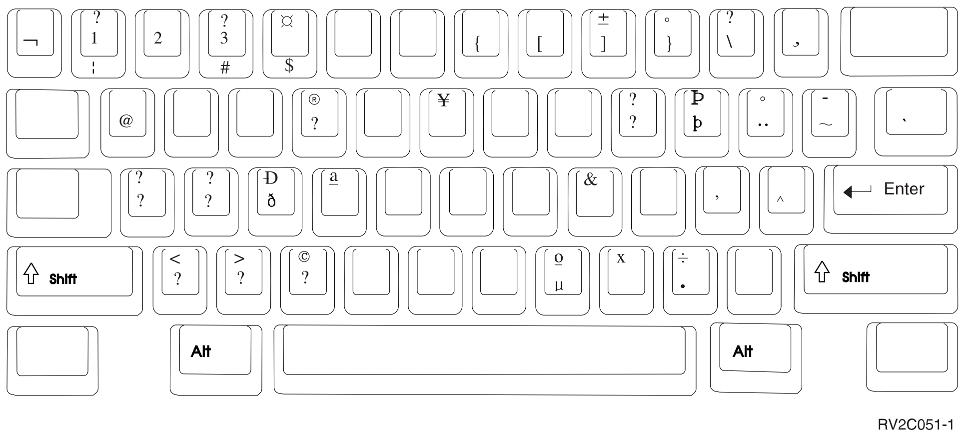
"Workstation considerations" on page 34
In a multilingual environment, different workstations support different languages on the same system.

Special-character keyboard set

The special-character keyboard set is available with the enhanced keyboard on most display stations. It allows a user to enter special characters that otherwise might not be available (labeled) on the keyboard.

The following figure shows all the characters in the special keyboard set and the keys on the enhanced keyboard to which each character is assigned. A special character can be assigned to a lowercase, uppercase, or ALT position on a key.

You can order a special template package, SCX21-9950, that contains the special-character keyboard set.



The special characters on the enhanced keyboard are used for this group of languages: Belgium, German, French, English, Icelandic, Italian, Spanish, Austrian, Danish, Portuguese, Swedish, Norwegian, Swiss/French, Swiss/German, Spanish Speaking, and Netherlands (Dutch).

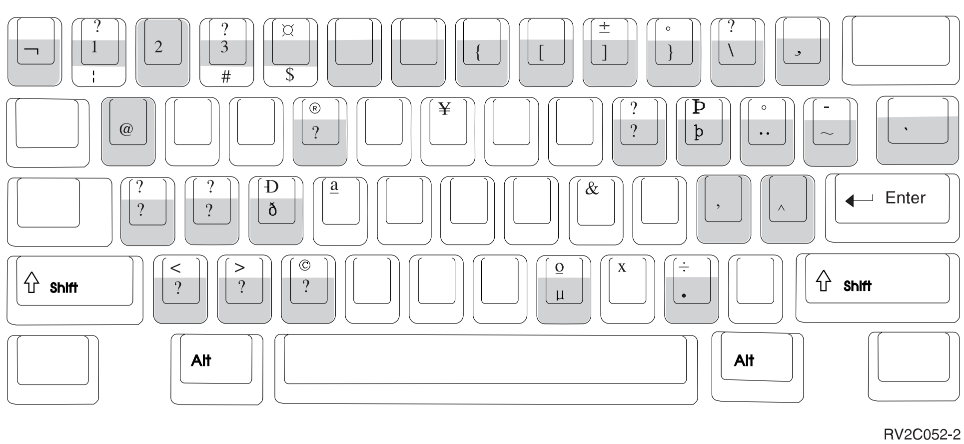
Creating a special character

The special-character keyboard function is active for only one special character at a time. To create a special character, follow these steps:

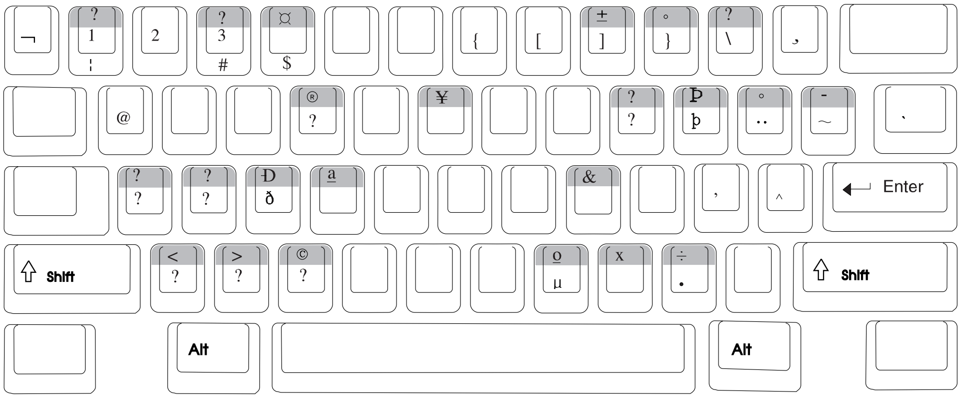
1. Press and hold the ALT key, and then press the Shift key.
2. Press the key to which a special character is assigned.

If a special character is assigned to an uppercase or ALT position, the shift or ALT key must be pressed in combination with the key assigned to the special character. The ALT and shift key sequence must be pressed before the entry of each special character.

The following figure highlights those special characters assigned to the lowercase positions.

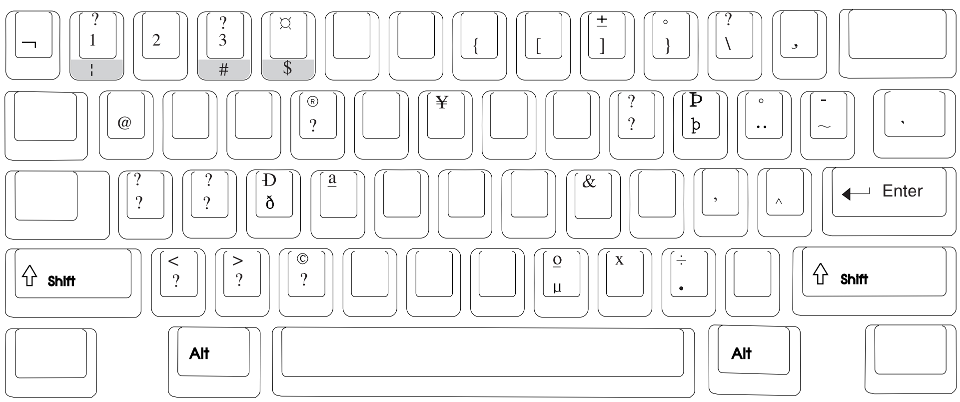


The following figure highlights the special characters assigned to the uppercase position.



RV2C053-1

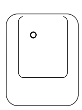
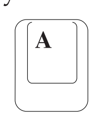
The following figure highlights the special characters assigned to the ALT key positions.



RV2C054-1

Example: Special character

For example, if you wanted to create the Å for German on an English keyboard, you should follow these steps:

1. Press and hold the ALT key, and then press the Shift key.
2. Press the Shift key with the  RV2C055-0 key. The diacritic mode symbol appears at the bottom of the screen and is waiting for the next keystroke to complete the character.
3. You now press the Shift key and the  RV2C056-0 key.

   This produces the Å. If you press this key without also pressing the shift key, you will get å. The system accepts only uppercase or lowercase A.

National language keyboard types and SBCS code pages

This table lists the keyboard types and code pages for each national language supported by the i5/OS operating system. The Create Device Display (CRTDEVDSP) command uses the KBDTYPE parameter.

Language	KBDTYPE	EBCDIC character set	EBCDIC SBCS code page	EBCDIC CCSID
Albanian	ALI	00697	00500	00500
Arabic	CLB	00235	00420	00420
Austrian/ German	AGB	00697	00273	00273
Austrian/German (MNCS)	AGI	00697	00500	00500
Belgian MNCS	BLI	00697	00500	00500
Brazilian Portuguese	BRB	00697	00037	00037
Bulgarian	BGB	01150	01025	01025
Canadian French	CAB	00341	00260	65535
Canadian French MNCS	CAI	00697	00500	00500
Croatian	YGI	00959	00870	00870
Cyrillic	CYB	00960	00880	00880
Czech	CSB	00959	00870	00870
Danish	DMB	00697	00277	00277
Danish MNCS	DMI	00697	00500	00500
Finnish/Swedish	FNB	00697	00278	00278
Finnish/Swedish MNCS	FNI	00697	00500	00500
French (Azerty)	FAB	00697	00297	00297
French (Azerty) MNCS	FAI	00697	00500	00500
French (Qwerty)	FQB	00697	00297	00297
French (Qwerty) MNCS	FQI	00697	00500	00500
Greek (See note 2.)	GNB	00925	00875	00875
Hebrew	NCB	00941	00424	00424
Hungarian	HNB	00959	00870	00870
Icelandic	ICB	00697	00871	00871
Icelandic MNCS	ICI	00697	00500	00500
International	INB	00697	00500	00500
International MNCS	INB	00697	00500	00500
Farsi (Iran)	IRB	01219	01097	01097
Italian	ITB	00697	00280	00280
Italian MNCS	ITI	00697	00500	00500
Japanese-English	JEB	00697	00281	65535
Japanese- English MNCS	JEI	00697	00500	00500
Japanese Kanji and Katakana	JKB	01172	00290	05026
Japanese Kanji and US English	JUB	00697	00037	See note 3.

Language	KBDTYPE	EBCDIC character set	EBCDIC SBCS code page	EBCDIC CCSID
Japanese Katakana	KAB	00332	00290	00290
Japanese Latin Extended	JPB	01172	01027	01027
Korean	KOB	01173	00833	00833
Latin 2	ROB	00959	00870	00870
Macedonian	MKB	01150	01025	01025
Dutch (Netherlands)	NEB	00697	00037	00037
Dutch (Netherlands) MNCS	NEI	00697	00500	00500
Norwegian	NWB	00697	00277	00277
Norwegian MNCS	NWI	00697	00500	00500
Polish	PLB	00959	00870	00870
Portuguese	PRB	00697	00037	00037
Portuguese MNCS	PRI	00697	00500	00500
Romanian	RMB	00959	00870	00870
Russian	RUB	01150	01025	01025
Serbian, Cyrillic	SQB	01150	01025	01025
Serbian, Latin	YGI	00959	00870	00870
Simplified Chinese	RCB	01174	00836	00836
Slovakian	SKB	00959	00870	00870
Slovenian	YGI	00959	00870	00870
Spanish	SPB	00697	00284	00284
Spanish MNCS	SPI	00697	00500	00500
Spanish Speaking	SSB	00697	00284	00284
Spanish Speaking MNCS	SSI	00697	00500	00500
Swedish	SWB	00697	00278	00278
Swedish MNCS	SWI	00697	00500	00500
French (Switzerland) MNCS	SFI	00697	00500	00500
German (Switzerland) MNCS	SGI	00697	00500	00500
Thai	THB	01176	00838	00838
Traditional Chinese	TAB	01175	00037	00937
Turkish (Qwerty)	TKB	01152	01026	01026
Turkish (F)	TRB	01152	01026	01026
English (United Kingdom)	UKB	00697	00285	00285
English (United Kingdom) MNCS	UKI	00697	00500	00500
English (United States and Canada)	USB	00697	00037	00037

Language	KBDTYPE	EBCDIC character set	EBCDIC SBCS code page	EBCDIC CCSID
English (United States and Canada) MNCS	USI	00697	00500	00500

Notes:

1. For KBDTYPE GKB, the EBCDIC code page is 00423.
2. Recommend SBCS CCSID 00037.

Related concepts

"Language indicator for keyboard type (QKBDTYPE) system value" on page 22
The language indicator for the keyboard type (QKBDTYPE) system value specifies the language character set for the keyboard.

"Recommendations and guidelines for using CCSIDs" on page 126
These recommendations are useful when you write globalized applications.

Related reference

Create Device Display (CRTDEVDSP) command

Code pages

Several IBM code pages match the International Standard ISO/IEC 8859. The i5/OS operating system supports parts of the ISO Standard with equivalent IBM code pages.

ISO/IEC 8859 consists of the following parts, under the general title *Information processing - 8-bit single-byte coded graphic character sets*:

- Part 1: Latin alphabet No. 1, 8859-1
- Part 2: Latin alphabet No. 2, 8859-2
- Part 3: Latin alphabet No. 3, 8859-3
- Part 4: Latin alphabet No. 4, 8835-4
- Part 5: Latin/Cyrillic alphabet, 8859-5
- Part 6: Latin/Arabic alphabet, 8858-6
- Part 7: Latin/Greek alphabet, 8859-7
- Part 8: Latin/Hebrew alphabet, 8859-8
- Part 9: Latin alphabet No. 5, 8859-9

i5/OS supports Parts 1, 2, and 9 of the ISO Standard with equivalent IBM code pages. Equivalent IBM code pages have characters that map to the ISO Standard. The equivalent IBM code pages contain all of the characters that are contained in the ISO standard. Equivalent IBM code pages are not identical to the ISO Standard. Some of the characters in the equivalent IBM code pages are at different code points than the same characters in the ISO standard.

i5/OS supports Parts 1, 2, 5, 7, 8, and 9 of the ISO Standard with identical IBM code pages. Identical IBM code pages are the same as the ISO standard.

You can find a list of code pages on the following Web sites:

- IBM Code page by CPGID: Globalizing your On Demand business
- System i Globalization

ISO standards and IBM EBCDIC code pages

Equivalent EBCDIC code page	ISO standard
00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00819, 00871	8859-1
00870	8859-2
00905	8859-9
01026	8859-9

ISO standards and IBM ASCII code pages

Equivalent ASCII code page	ISO standard
00852	8859-2
00857	8859-5

ISO standards and identical IBM code page

Identical ASCII code page	ISO standard
00813	8859-7
00819	8859-1
00916	8859-8
00920	8859-9

Notes:

1. Hexadecimal value 40 represents the space character on the EBCDIC code pages.
2. Hexadecimal value 20 represents the space character on the PC code pages.
3. Hexadecimal value FF represents the eight ones (11111111) control character.
4. The 8-digit alphanumeric label under each character in the code page chart is the graphic character global identifier (GCGID). The label is used with Sort sequence tables.

Related concepts

System i globalization

"Sort sequence tables" on page 156

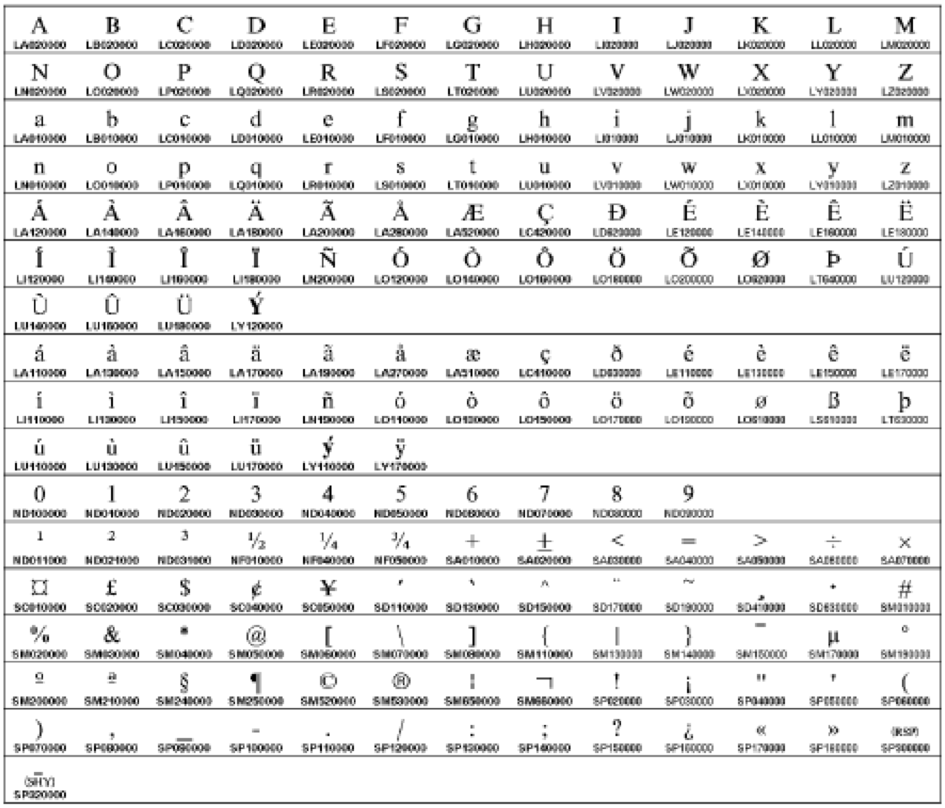
A sort sequence table is an object that contains the weight of each single-byte graphic character within a specified coded character set identifier (CCSID). The system-recognized identifier for the sort sequence table object type is *TBL.

Character sets

This reference information is about various character sets on the i5/OS operating system.

Country extended character set 00697

The figure shows country extended character set 00697.



Character Set 00697

Graphic character conversion tables

Table (*TBL) objects support non-CCSID conversions from one code page to another. The system-supplied table objects are located in the QUSRSYS library.

To see the supported code pages, run the following command:

```
WRKOBJ OBJ(QUSRSYS/*ALL) OBJTYPE(*TBL)
```

The table description shows the *from* code page and the *to* code page and character set. For example, if the from code page is 1112 and the to code page is 500, the description shows the following text:

```
CHRID(*N 1112) to CHRID(697 500)
```

This conversion method is supported on the i5/OS operating system, but is not enhanced.

To convert data, a preferred method is to use the systems CCSID support. For details, see Character Conversion APIs.

Table name	From	From text	To	To text
QA3R	(*N 1025)	Cyrillic, Multilingual	Uppercase table	
QA3S	(*N 1026)	Turkey Latin 5	Uppercase table	
QA3T	(*N 1027)	Japan extended	Uppercase table	
QA5R	(*N 1097)	Farsi	Uppercase table	
QA57	(*N 1112)	Baltic, Multilingual	Uppercase table	

Table name	From	From text	To	To text
QA6G	(*N 1122)	Estonian	Uppercase table	
QA6O	(*N 1130)	Vietnamese	Uppercase table	
QA6Q	(*N 1133)	Lao	Uppercase table	
Q037	(*N 037)	USA/Canada	Uppercase table	
Q256	(*N 256)	Multinational #1	Uppercase table	
Q260	(*N 260)	Canada French	Uppercase table	
Q273	(*N 273)	Austria/Germany	Uppercase table	
Q277	(*N 277)	Denmark/Norway	Uppercase table	
Q278	(*N 278)	Finland/Sweden	Uppercase table	
Q280	(*N 280)	Italy	Uppercase table	
Q281	(*N 281)	Japan (Latin)	Uppercase table	
Q284	(*N 284)	Spain/Latin America	Uppercase table	
Q285	(*N 285)	United Kingdom	Uppercase table	
Q290	(*N 290)	Japan Katakana	Uppercase table	
Q297	(*N 297)	France	Uppercase table	
Q420	(*N 420)	Arabic Bilingual	Uppercase table	
Q423	(*N 423)	Greece	Uppercase table	
Q424	(*N 424)	Israel (Hebrew)	Uppercase table	
Q437	(*N 437)	PC-USA	Uppercase table	
Q500	(*N 500)	Multinational #5	Uppercase table	
Q833	(*N 833)	Korea	Uppercase table	
Q836	(*N 836)	People's Republic of China	Uppercase table	
Q838	(*N 838)	Thai Extended	Uppercase table	
Q850	(*N 850)	PC-Multilingual	Uppercase table	
Q851	(*N 851)	PC-Greece (old)	Uppercase table	
Q857	(*N 857)	PC-Turkey	Uppercase table	
Q860	(*N 860)	PC-Portugal	Uppercase table	
Q861	(*N 861)	PC-Iceland	Uppercase table	
Q862	(*N 862)	PC-Israel	Uppercase table	
Q863	(*N 863)	PC-Canadian French	Uppercase table	
Q864	(*N 864)	PC-Arabic	Uppercase table	
Q865	(*N 865)	PC-Nordic	Uppercase table	
Q870	(*N 870)	Latin 2, Multilingual	Uppercase table	
Q871	(*N 871)	Iceland	Uppercase table	
Q875	(*N 875)	Greece	Uppercase table	
Q880	(*N 880)	Cyrillic, Multilingual (old)	Uppercase table	
Q891	(*N 891)	PC-Korea	Uppercase table	
Q897	(*N 897)	PC-Japan	Uppercase table	
Q903	(*N 903)	PC-People's Republic of China	Uppercase table	

Table name	From	From text	To	To text
Q904	(*N 904)	PC-People's Republic of China	Uppercase table	
Q905	(*N 905)	PC-Turkey	Uppercase table	
Q037A0MA5K	(*N 037)	USA/Canada	(1272 1090)	VT100 Line drawing set
Q037A05A5U	(*N 037)	USA/Canada	(1290 1100)	VT220 Multinational
Q037A6G897	(*N 037)	USA/Canada	(1122 897)	PC-Japan
Q037A69A3R	(*N 037)	USA/Canada	(1150 1025)	Cyrillic, Multilingual
Q037A7AA3S	(*N 037)	USA/Canada	(1152 1026)	Turkey Latin 5
Q037A7RA3B	(*N 037)	USA/Canada	(1169 1009)	International Alphabet 5
Q037A7UA3T	(*N 037)	USA/Canada	(1172 1027)	Japan extended
Q037A7U290	(*N 037)	USA/Canada	(1172 290)	Japan Katakana
Q037A7V833	(*N 037)	USA/Canada	(1173 833)	Korea
Q037A7W836	(*N 037)	USA/Canada	(1174 836)	People's Republic of China
Q037A7Y838	(*N 037)	USA/Canada	(1176 838)	Thai Extended
Q037A7Y874	(*N 037)	USA/Canada	(1176 874)	PC-Thai Extended
Q037A86A5R	(*N 037)	USA/Canada	(1219 1097)	Farsi
Q037BAJA57	(*N 037)	USA/Canada	(1305 1112)	Baltic, Multilingual
Q037BALA6G	(*N 037)	USA/Canada	(1307 1122)	Estonian
Q037BBEA6O	(*N 037)	USA/Canada	(1336 1130)	Vietnamese
Q037BBJA6Q	(*N 037)	USA/Canada	(1341 1133)	Lao
Q037101037	(*N 037)	USA/Canada	(101 037)	USA/Canada
Q037103A59	(*N 037)	USA/Canada	(103 1114)	Taiwan Industry (Big 5) code
Q037235420	(*N 037)	USA/Canada	(235 420)	Arabic Bilingual
Q037337256	(*N 037)	USA/Canada	(337 256)	Multinational #1
Q037337437	(*N 037)	USA/Canada	(337 437)	PC-USA
Q037337850	(*N 037)	USA/Canada	(337 850)	PC-Multilingual
Q037337860	(*N 037)	USA/Canada	(337 860)	PC-Portugal
Q037337863	(*N 037)	USA/Canada	(337 863)	PC-Canadian French
Q037337904	(*N 037)	USA/Canada	(337 904)	PC-People's Republic of China
Q037936836	(*N 037)	USA/Canada	(936 836)	People's Republic of China
Q037941424	(*N 037)	USA/Canada	(941 424)	Israel (Hebrew)
Q037959870	(*N 037)	USA/Canada	(959 870)	Latin 2, Multilingual
Q037960880	(*N 037)	USA/Canada	(960 880)	Cyrillic, Multilingual (old)
Q037965905	(*N 037)	USA/Canada	(965 905)	PC-Turkey
Q038337256	(*N 038)	USA/ASCII	(337 256)	Multinational #1
Q256A69A3R	(*N 256)	Multinational #1	(1150 1025)	Cyrillic, Multilingual

Table name	From	From text	To	To text
Q256A7AA3S	(*N 256)	Multinational #1	(1152 1026)	Turkey Latin 5
Q256A7V833	(*N 256)	Multinational #1	(1173 833)	Korea
Q256A7W836	(*N 256)	Multinational #1	(1174 836)	People's Republic of China
Q256A7X037	(*N 256)	Multinational #1	(1175 037)	USA/Canada
Q256001256	(*N 256)	Multinational #1	(001 256)	Multinational #1
Q256101037	(*N 256)	Multinational #1	(101 037)	USA/Canada
Q256101367	(*N 256)	Multinational #1	(101 367)	ASCII
Q256103038	(*N 256)	Multinational #1	(103 038)	USA/ASCII
Q256218423	(*N 256)	Multinational #1	(218 423)	Greece
Q256265273	(*N 256)	Multinational #1	(265 273)	Austria/Germany
Q256269274	(*N 256)	Multinational #1	(269 274)	Belgium
Q256273275	(*N 256)	Multinational #1	(273 275)	Brazil
Q256277276	(*N 256)	Multinational #1	(277 276)	Canada (French)
Q256281277	(*N 256)	Multinational #1	(281 277)	Denmark/Norway
Q256285278	(*N 256)	Multinational #1	(285 278)	Finland/Sweden
Q256288297	(*N 256)	Multinational #1	(288 297)	France
Q256289279	(*N 256)	Multinational #1	(289 279)	France
Q256293280	(*N 256)	Multinational #1	(293 280)	Italy
Q256297281	(*N 256)	Multinational #1	(297 281)	Japan (Latin)
Q256301282	(*N 256)	Multinational #1	(301 282)	Portugal
Q256305283	(*N 256)	Multinational #1	(305 283)	Spain
Q256309284	(*N 256)	Multinational #1	(309 284)	Spain/Latin America
Q256313285	(*N 256)	Multinational #1	(313 285)	United Kingdom
Q256332290	(*N 256)	Multinational #1	(332 290)	Japan Katakana
Q256337037	(*N 256)	Multinational #1	(337 037)	USA/Canada
Q256337273	(*N 256)	Multinational #1	(337 273)	Austria/Germany
Q256337274	(*N 256)	Multinational #1	(337 274)	Belgium
Q256337275	(*N 256)	Multinational #1	(337 275)	Brazil
Q256337276	(*N 256)	Multinational #1	(337 276)	Canada (French)
Q256337277	(*N 256)	Multinational #1	(337 277)	Denmark/Norway
Q256337278	(*N 256)	Multinational #1	(337 278)	Finland/Sweden
Q256337280	(*N 256)	Multinational #1	(337 280)	Italy
Q256337281	(*N 256)	Multinational #1	(337 281)	Japan (Latin)
Q256337282	(*N 256)	Multinational #1	(337 282)	Portugal
Q256337283	(*N 256)	Multinational #1	(337 283)	Spain
Q256337284	(*N 256)	Multinational #1	(337 284)	Spain/Latin America
Q256337285	(*N 256)	Multinational #1	(337 285)	United Kingdom
Q256337297	(*N 256)	Multinational #1	(337 297)	France
Q256337420	(*N 256)	Multinational #1	(337 420)	Arabic Bilingual
Q256337833	(*N 256)	Multinational #1	(337 833)	Korea

Table name	From	From text	To	To text
Q256338257	(*N 256)	Multinational #1	(338 257)	Multinational #2
Q256339258	(*N 256)	Multinational #1	(339 258)	Multinational #3
Q256340259	(*N 256)	Multinational #1	(340 259)	Symbols, Set #7
Q256341260	(*N 256)	Multinational #1	(341 260)	Canada French
Q256697871	(*N 256)	Multinational #1	(697 871)	Iceland
Q256925875	(*N 256)	Multinational #1	(925 875)	Greece
Q256933833	(*N 256)	Multinational #1	(933 833)	Korea
Q256936836	(*N 256)	Multinational #1	(936 836)	People's Republic of China
Q256938838	(*N 256)	Multinational #1	(938 838)	Thai Extended
Q256941424	(*N 256)	Multinational #1	(941 424)	Israel (Hebrew)
Q256959870	(*N 256)	Multinational #1	(959 870)	Latin 2, Multilingual
Q256960880	(*N 256)	Multinational #1	(960 880)	Cyrillic, Multilingual (old)
Q257337256	(*N 257)	Multinational #2	(337 256)	Multinational #1
Q258337256	(*N 258)	Multinational #3	(337 256)	Multinational #1
Q259337256	(*N 259)	Symbols, Set #7	(337 256)	Multinational #1
Q260337256	(*N 260)	Canada French	(337 256)	Multinational #1
Q273A7RA3B	(*N 273)	Austria/Germany	(1169 1009)	International Alphabet 5
Q273337256	(*N 273)	Austria/Germany	(337 256)	Multinational #1
Q273337437	(*N 273)	Austria/Germany	(337 437)	PC-USA
Q273337850	(*N 273)	Austria/Germany	(337 850)	PC-Multilingual
Q274337256	(*N 274)	Belgium	(337 256)	Multinational #1
Q275337256	(*N 275)	Brazil	(337 256)	Multinational #1
Q276337256	(*N 276)	Canada (French)	(337 256)	Multinational #1
Q277A7RA3B	(*N 277)	Denmark/Norway	(1169 1009)	International Alphabet 5
Q277337256	(*N 277)	Denmark/Norway	(337 256)	Multinational #1
Q277337850	(*N 277)	Denmark/Norway	(337 850)	PC-Multilingual
Q277337865	(*N 277)	Denmark/Norway	(337 865)	PC-Nordic
Q277697284	(*N 277)	Denmark/Norway	(697 284)	Spain/Latin America
Q278A7RA3B	(*N 278)	Finland/Sweden	(1169 1009)	International Alphabet 5
Q278337256	(*N 278)	Finland/Sweden	(337 256)	Multinational #1
Q278337437	(*N 278)	Finland/Sweden	(337 437)	PC-USA
Q278337850	(*N 278)	Finland/Sweden	(337 850)	PC-Multilingual
Q279337256	(*N 279)	France	(337 256)	Multinational #1
Q280A7RA3B	(*N 280)	Italy	(1169 1009)	International Alphabet 5
Q280337256	(*N 280)	Italy	(337 256)	Multinational #1
Q280337437	(*N 280)	Italy	(337 437)	PC-USA

Table name	From	From text	To	To text
Q280337850	(*N 280)	Italy	(337 850)	PC-Multilingual
Q281337256	(*N 281)	Japan (Latin)	(337 256)	Multinational #1
Q282337256	(*N 282)	Portugal	(337 256)	Multinational #1
Q282337850	(*N 282)	Portugal	(337 850)	PC-Multilingual
Q282337860	(*N 282)	Portugal	(337 860)	PC-Portugal
Q283337256	(*N 283)	Spain	(337 256)	Multinational #1
Q284A7RA3B	(*N 284)	Spain/Latin America	(1169 1009)	International Alphabet 5
Q284A7W836	(*N 284)	Spain/Latin America	(1174 836)	People's Republic of China
Q284337256	(*N 284)	Spain/Latin America	(337 256)	Multinational #1
Q284337437	(*N 284)	Spain/Latin America	(337 437)	PC-USA
Q284337850	(*N 284)	Spain/Latin America	(337 850)	PC-Multilingual
Q284697277	(*N 284)	Spain/Latin America	(697 277)	Denmark/Norway
Q285337256	(*N 285)	United Kingdom	(337 256)	Multinational #1
Q285337437	(*N 285)	United Kingdom	(337 437)	PC-USA
Q285337850	(*N 285)	United Kingdom	(337 850)	PC-Multilingual
Q290A7RA3B	(*N 290)	Japan Katakana	(1169 1009)	International Alphabet 5
Q290A7UA3T	(*N 290)	Japan Katakana	(1172 1027)	Japan extended
Q290A7UA38	(*N 290)	Japan Katakana	(1172 1041)	PC-Japan extended
Q290337256	(*N 290)	Japan Katakana	(337 256)	Multinational #1
Q290337897	(*N 290)	Japan Katakana	(337 897)	PC-Japan
Q290697037	(*N 290)	Japan Katakana	(697 037)	USA/Canada
Q290697500	(*N 290)	Japan Katakana	(697 500)	Multinational #5
Q297A7RA3B	(*N 297)	France	(1169 1009)	International Alphabet 5
Q297337256	(*N 297)	France	(337 256)	Multinational #1
Q297337437	(*N 297)	France	(337 437)	PC-USA
Q297337850	(*N 297)	France	(337 850)	PC-Multilingual
Q367A7RA3B	(*N 367)	ASCII	(1169 1009)	International Alphabet 5
Q367337256	(*N 367)	ASCII	(337 256)	Multinational #1
Q367697500	(*N 367)	ASCII	(697 500)	Multinational #5
Q420235864	(*N 420)	Arabic Bilingual	(235 864)	PC-Arabic
Q420337256	(*N 420)	Arabic Bilingual	(337 256)	Multinational #1
Q420697037	(*N 420)	Arabic Bilingual	(697 037)	USA/Canada
Q420697500	(*N 420)	Arabic Bilingual	(697 500)	Multinational #5
Q423A7RA3B	(*N 423)	Greece	(1169 1009)	International Alphabet 5
Q423218851	(*N 423)	Greece	(218 851)	PC-Greece (old)
Q423697256	(*N 423)	Greece	(697 256)	Multinational #1

Table name	From	From text	To	To text
Q423925875	(*N 423)	Greece	(925 875)	Greece
Q423998869	(*N 423)	Greece	(998 869)	PC-Greece
Q424697037	(*N 424)	Israel (Hebrew)	(697 037)	USA/Canada
Q424697256	(*N 424)	Israel (Hebrew)	(697 256)	Multinational #1
Q424697500	(*N 424)	Israel (Hebrew)	(697 500)	Multinational #5
Q424941862	(*N 424)	Israel (Hebrew)	(941 862)	PC-Israel
Q437A0ZA5S	(*N 437)	PC-USA	(1285 1098)	PC-Farsi
Q437A69A3R	(*N 437)	PC-USA	(1150 1025)	Cyrillic, Multilingual
Q437A69915	(*N 437)	PC-USA	(1150 915)	8-bit ASCII/ISO Cyrillic
Q437A7X037	(*N 437)	PC-USA	(1175 037)	USA/Canada
Q437A8C866	(*N 437)	PC-USA	(1190 866)	PC-Cyrillic #2
Q437A9I852	(*N 437)	PC-USA	(1232 852)	PC-Latin 2
Q437A9L855	(*N 437)	PC-USA	(1235 855)	PC-Cyrillic
Q437A9N857	(*N 437)	PC-USA	(1237 857)	PC-Turkey
Q437101037	(*N 437)	PC-USA	(101 037)	USA/Canada
Q437337A5R	(*N 437)	PC-USA	(337 1097)	Farsi
Q437337037	(*N 437)	PC-USA	(337 037)	USA/Canada
Q437337273	(*N 437)	PC-USA	(337 273)	Austria/Germany
Q437337278	(*N 437)	PC-USA	(337 278)	Finland/Sweden
Q437337280	(*N 437)	PC-USA	(337 280)	Italy
Q437337284	(*N 437)	PC-USA	(337 284)	Spain/Latin America
Q437337285	(*N 437)	PC-USA	(337 285)	United Kingdom
Q437337297	(*N 437)	PC-USA	(337 297)	France
Q437337500	(*N 437)	PC-USA	(337 500)	Multinational #5
Q437959870	(*N 437)	PC-USA	(959 870)	Latin 2, Multilingual
Q437960880	(*N 437)	PC-USA	(960 880)	Cyrillic, Multilingual (old)
Q500A0MA5K	(*N 500)	Multinational #5	(1272 1090)	VT100 Line drawing set
Q500A00A5Z	(*N 500)	Multinational #5	(1295 1105)	VT220 Norwegian/Danish
Q500A05A5U	(*N 500)	Multinational #5	(1290 1100)	VT220 Multinational
Q500A06A5V	(*N 500)	Multinational #5	(1291 1101)	VT220 British
Q500A07A5W	(*N 500)	Multinational #5	(1292 1102)	VT220 Dutch
Q500A08A5X	(*N 500)	Multinational #5	(1293 1103)	VT220 Finnish
Q500A69A3R	(*N 500)	Multinational #5	(1150 1025)	Cyrillic, Multilingual
Q500A7AA3S	(*N 500)	Multinational #5	(1152 1026)	Turkey Latin 5
Q500A7RA3B	(*N 500)	Multinational #5	(1169 1009)	International Alphabet 5
Q500A7UA3T	(*N 500)	Multinational #5	(1172 1027)	Japan extended
Q500A7U290	(*N 500)	Multinational #5	(1172 290)	Japan Katakana

Table name	From	From text	To	To text
Q500A7W836	(*N 500)	Multinational #5	(1174 836)	People's Republic of China
Q500A7X037	(*N 500)	Multinational #5	(1175 037)	USA/Canada
Q500A7Y838	(*N 500)	Multinational #5	(1176 838)	Thai Extended
Q500A7Y874	(*N 500)	Multinational #5	(1176 874)	PC-Thai Extended
Q500A8EA3M	(*N 500)	Multinational #5	(1192 1020)	VT220 Canadian/French
Q500A8FA3N	(*N 500)	Multinational #5	(1193 1021)	VT220 Switzerland
Q500A8HA3P	(*N 500)	Multinational #5	(1195 1023)	VT220 Spain
Q500A86A5R	(*N 500)	Multinational #5	(1219 1097)	Farsi
Q500BAAA51	(*N 500)	Multinational #5	(1296 1106)	VT220 Swedish
Q500BABA52	(*N 500)	Multinational #5	(1297 1107)	VT220 Norwegian/Danish Alt
Q500BAJA57	(*N 500)	Multinational #5	(1305 1112)	Baltic, Multilingual
Q500BALA6G	(*N 500)	Multinational #5	(1307 1122)	Estonian
Q500BBEA6O	(*N 500)	Multinational #5	(1336 1130)	Vietnamese
Q500BBJA6Q	(*N 500)	Multinational #5	(1341 1133)	Loa
Q500103367	(*N 500)	Multinational #5	(103 367)	ASCII
Q500235420	(*N 500)	Multinational #5	(235 420)	Arabic Bilingual
Q500265A3D	(*N 500)	Multinational #5	(265 1011)	VT220 Germany
Q500289A5Y	(*N 500)	Multinational #5	(289 1104)	VT220 French
Q500293A3E	(*N 500)	Multinational #5	(293 1012)	VT220 Italy
Q500337437	(*N 500)	Multinational #5	(337 437)	PC-USA
Q500337836	(*N 500)	Multinational #5	(337 836)	People's Republic of China
Q500337850	(*N 500)	Multinational #5	(337 850)	PC-Multilingual
Q500337860	(*N 500)	Multinational #5	(337 860)	PC-Portugal
Q500337861	(*N 500)	Multinational #5	(337 861)	PC-Iceland
Q500337863	(*N 500)	Multinational #5	(337 863)	PC-Canadian French
Q500337865	(*N 500)	Multinational #5	(337 865)	PC-Nordic
Q500697037	(*N 500)	Multinational #5	(697 037)	USA/Canada
Q500697280	(*N 500)	Multinational #5	(697 280)	Italy
Q500925875	(*N 500)	Multinational #5	(925 875)	Greece
Q500936836	(*N 500)	Multinational #5	(936 836)	People's Republic of China
Q500941424	(*N 500)	Multinational #5	(941 424)	Israel (Hebrew)
Q500959870	(*N 500)	Multinational #5	(959 870)	Latin 2, Multilingual
Q500960880	(*N 500)	Multinational #5	(960 880)	Cyrillic, Multilingual (old)
Q500965905	(*N 500)	Multinational #5	(965 905)	PC-Turkey
Q500981851	(*N 500)	Multinational #5	(981 851)	PC-Greece (old)
Q500998869	(*N 500)	Multinational #5	(998 869)	PC-Greece

Table name	From	From text	To	To text
Q813998869	(*N 813)	8-bit ASCII/ISO Greece	(998 869)	PC-Greece
Q819BBEA6O	(*N 819)	8-bit ASCII/ISO Latin 1	(1336 1130)	Vietnamese
Q819BBJA6Q	(*N 819)	8-bit ASCII/ISO Latin 1	(1341 1133)	Lao
Q833A0SA5I	(*N 833)	Korea	(1278 1088)	PC-Korean
Q833A7RA3B	(*N 833)	Korea	(1169 1009)	International Alphabet 5
Q833337256	(*N 833)	Korea	(337 256)	Multinational #1
Q833337891	(*N 833)	Korea	(337 891)	PC-Korea
Q833933256	(*N 833)	Korea	(933 256)	Multinational #1
Q836A7RA3B	(*N 836)	People's Republic of China	(1169 1009)	International Alphabet 5
Q836A7V833	(*N 836)	People's Republic of China	(1173 833)	Korea
Q836A7X037	(*N 836)	People's Republic of China	(1175 037)	USA/Canada
Q836101037	(*N 836)	People's Republic of China	(101 037)	USA/Canada
Q836103A50	(*N 836)	People's Republic of China	(103 1115)	People's Republic of China GB
Q836337256	(*N 836)	People's Republic of China	(337 256)	Multinational #1
Q836337500	(*N 836)	People's Republic of China	(337 500)	Multinational #5
Q836337903	(*N 836)	People's Republic of China	(337 903)	PC-People's Republic of China
Q836697037	(*N 836)	People's Republic of China	(697 037)	USA/Canada
Q836697284	(*N 836)	People's Republic of China	(697 284)	Spain/Latin America
Q836936500	(*N 836)	People's Republic of China	(936 500)	Multinational #5
Q838A7Y874	(*N 838)	Thai Extended	(1176 874)	PC-Thai Extended
Q838337037	(*N 838)	Thai Extended	(337 037)	USA/Canada
Q838697500	(*N 838)	Thai Extended	(697 500)	Multinational #5
Q850A0ZA5S	(*N 850)	PC-Multilingual	(1285 1098)	PC-Farsi
Q850A6G897	(*N 850)	PC-Multilingual	(1122 897)	PC-Japan
Q850A69A3R	(*N 850)	PC-Multilingual	(1150 1025)	Cyrillic, Multilingual
Q850A69915	(*N 850)	PC-Multilingual	(1150 915)	8-bit ASCII/ISO Cyrillic
Q850A7UA38	(*N 850)	PC-Multilingual	(1172 1041)	PC-Japan extended
Q850A7Y874	(*N 850)	PC-Multilingual	(1176 874)	PC-Thai Extended
Q850A8C866	(*N 850)	PC-Multilingual	(1190 866)	PC-Cyrillic #2
Q850A84862	(*N 850)	PC-Multilingual	(1217 862)	PC-Israel

Table name	From	From text	To	To text
Q850A9I852	(*N 850)	PC-Multilingual	(1232 852)	PC-Latin 2
Q850A9L855	(*N 850)	PC-Multilingual	(1235 855)	PC-Cyrillic
Q850A9N857	(*N 850)	PC-Multilingual	(1237 857)	PC-Turkey
Q850A9U864	(*N 850)	PC-Multilingual	(1244 864)	PC-Arabic
Q850BBEA6O	(*N 850)	PC-Multilingual	(1336 1130)	Vietnamese
Q850BBJA6Q	(*N 850)	PC-Multilingual	(1341 1133)	Lao
Q850337A5R	(*N 850)	PC-Multilingual	(337 1097)	Farsi
Q850337037	(*N 850)	PC-Multilingual	(337 037)	USA/Canada
Q850337273	(*N 850)	PC-Multilingual	(337 273)	Austria/Germany
Q850337277	(*N 850)	PC-Multilingual	(337 277)	Denmark/Norway
Q850337278	(*N 850)	PC-Multilingual	(337 278)	Finland/Sweden
Q850337280	(*N 850)	PC-Multilingual	(337 280)	Italy
Q850337282	(*N 850)	PC-Multilingual	(337 282)	Portugal
Q850337284	(*N 850)	PC-Multilingual	(337 284)	Spain/Latin America
Q850337285	(*N 850)	PC-Multilingual	(337 285)	United Kingdom
Q850337297	(*N 850)	PC-Multilingual	(337 297)	France
Q850337500	(*N 850)	PC-Multilingual	(337 500)	Multinational #5
Q850337871	(*N 850)	PC-Multilingual	(337 871)	Iceland
Q850959870	(*N 850)	PC-Multilingual	(959 870)	Latin 2, Multilingual
Q850960880	(*N 850)	PC-Multilingual	(960 880)	Cyrillic, Multilingual (old)
Q851218423	(*N 851)	PC-Greece (old)	(218 423)	Greece
Q851925875	(*N 851)	PC-Greece (old)	(925 875)	Greece
Q852A51850	(*N 852)	PC-Latin 2	(1106 850)	PC-Multilingual
Q852A69A3R	(*N 852)	PC-Latin 2	(1150 1025)	Cyrillic, Multilingual
Q852A8Y437	(*N 852)	PC-Latin 2	(1212 437)	PC-USA
Q852959870	(*N 852)	PC-Latin 2	(959 870)	Latin 2, Multilingual
Q852960880	(*N 852)	PC-Latin 2	(960 880)	Cyrillic, Multilingual (old)
Q853965905	(*N 853)	PC-Latin 3	(965 905)	PC-Turkey
Q855A51850	(*N 855)	PC-Cyrillic	(1106 850)	PC-Multilingual
Q855A69A3R	(*N 855)	PC-Cyrillic	(1150 1025)	Cyrillic, Multilingual
Q855A69915	(*N 855)	PC-Cyrillic	(1150 915)	8-bit ASCII/ISO Cyrillic
Q855A8Y437	(*N 855)	PC-Cyrillic	(1212 437)	PC-USA
Q855959870	(*N 855)	PC-Cyrillic	(959 870)	Latin 2, Multilingual
Q855960880	(*N 855)	PC-Cyrillic	(960 880)	Cyrillic, Multilingual (old)
Q857A51850	(*N 857)	PC-Turkey	(1106 850)	PC-Multilingual
Q857A7AA3S	(*N 857)	PC-Turkey	(1152 1026)	Turkey Latin 5
Q857A8Y437	(*N 857)	PC-Turkey	(1212 437)	PC-USA
Q857965905	(*N 857)	PC-Turkey	(965 905)	PC-Turkey

Table name	From	From text	To	To text
Q860337037	(*N 860)	PC-Portugal	(337 037)	USA/Canada
Q860337282	(*N 860)	PC-Portugal	(337 282)	Portugal
Q860337500	(*N 860)	PC-Portugal	(337 500)	Multinational #5
Q861337500	(*N 861)	PC-Iceland	(337 500)	Multinational #5
Q861337871	(*N 861)	PC-Iceland	(337 871)	Iceland
Q862A51850	(*N 862)	PC-Israel	(1106 850)	PC-Multilingual
Q862941424	(*N 862)	PC-Israel	(941 424)	Israel (Hebrew)
Q863337037	(*N 863)	PC-Canadian French	(337 037)	USA/Canada
Q863337500	(*N 863)	PC-Canadian French	(337 500)	Multinational #5
Q864A51850	(*N 864)	PC-Arabic	(1106 850)	PC-Multilingual
Q864235420	(*N 864)	PC-Arabic	(235 420)	Arabic Bilingual
Q865337277	(*N 865)	PC-Nordic	(337 277)	Denmark/Norway
Q865337500	(*N 865)	PC-Nordic	(337 500)	Multinational #5
Q866A51850	(*N 866)	PC-Cyrillic #2	(1106 850)	PC-Multilingual
Q866A69A3R	(*N 866)	PC-Cyrillic #2	(1150 1025)	Cyrillic, Multilingual
Q866A8Y437	(*N 866)	PC-Cyrillic #2	(1212 437)	PC-USA
Q869218423	(*N 869)	PC-Greece	(218 423)	Greece
Q869337256	(*N 869)	PC-Greece	(337 256)	Multinational #1
Q869337500	(*N 869)	PC-Greece	(337 500)	Multinational #5
Q869925813	(*N 869)	PC-Greece	(925 813)	8-bit ASCII/ISO Greece
Q869925875	(*N 869)	PC-Greece	(925 875)	Greece
Q869981851	(*N 869)	PC-Greece	(981 851)	PC-Greece (old)
Q870A69A3R	(*N 870)	Latin 2, Multilingual	(1150 1025)	Cyrillic, Multilingual
Q870A69855	(*N 870)	Latin 2, Multilingual	(1150 855)	PC-Cyrillic
Q870A69915	(*N 870)	Latin 2, Multilingual	(1150 915)	8-bit ASCII/ISO Cyrillic
Q870A7RA3B	(*N 870)	Latin 2, Multilingual	(1169 1009)	International Alphabet 5
Q870A9I852	(*N 870)	Latin 2, Multilingual	(1232 852)	PC-Latin 2
Q870337256	(*N 870)	Latin 2, Multilingual	(337 256)	Multinational #1
Q870697037	(*N 870)	Latin 2, Multilingual	(697 037)	USA/Canada
Q870697500	(*N 870)	Latin 2, Multilingual	(697 500)	Multinational #5
Q870697850	(*N 870)	Latin 2, Multilingual	(697 850)	PC-Multilingual
Q870919437	(*N 870)	Latin 2, Multilingual	(919 437)	PC-USA
Q870959852	(*N 870)	Latin 2, Multilingual	(959 852)	PC-Latin 2
Q870959912	(*N 870)	Latin 2, Multilingual	(959 912)	8-bit ASCII/ISO Latin 2
Q870960880	(*N 870)	Latin 2, Multilingual	(960 880)	Cyrillic, Multilingual (old)
Q871A7RA3B	(*N 871)	Iceland	(1169 1009)	International Alphabet 5

Table name	From	From text	To	To text
Q871337850	(*N 871)	Iceland	(337 850)	PC-Multilingual
Q871337861	(*N 871)	Iceland	(337 861)	PC-Iceland
Q871697256	(*N 871)	Iceland	(697 256)	Multinational #1
Q874A51850	(*N 874)	PC-Thai Extended	(1106 850)	PC-Multilingual
Q874A7Y500	(*N 874)	PC-Thai Extended	(1176 500)	Multinational #5
Q874A7Y838	(*N 874)	PC-Thai Extended	(1176 838)	Thai Extended
Q874337037	(*N 874)	PC-Thai Extended	(337 037)	USA/Canada
Q875A7RA3B	(*N 875)	Greece	(1169 1009)	International Alphabet 5
Q875218423	(*N 875)	Greece	(218 423)	Greece
Q875337256	(*N 875)	Greece	(337 256)	Multinational #1
Q875337500	(*N 875)	Greece	(337 500)	Multinational #5
Q875925813	(*N 875)	Greece	(925 813)	8-bit ASCII/ISO Greece
Q875981851	(*N 875)	Greece	(981 851)	PC-Greece (old)
Q875998869	(*N 875)	Greece	(998 869)	PC-Greece
Q880A69A3R	(*N 880)	Cyrillic, Multilingual (old)	(1150 1025)	Cyrillic, Multilingual
Q880A69855	(*N 880)	Cyrillic, Multilingual (old)	(1150 855)	PC-Cyrillic
Q880A69915	(*N 880)	Cyrillic, Multilingual (old)	(1150 915)	8-bit ASCII/ISO Cyrillic
Q880A7RA3B	(*N 880)	Cyrillic, Multilingual (old)	(1169 1009)	International Alphabet 5
Q880337256	(*N 880)	Cyrillic, Multilingual (old)	(337 256)	Multinational #1
Q880697037	(*N 880)	Cyrillic, Multilingual (old)	(697 037)	USA/Canada
Q880697500	(*N 880)	Cyrillic, Multilingual (old)	(697 500)	Multinational #5
Q880697850	(*N 880)	Cyrillic, Multilingual (old)	(697 850)	PC-Multilingual
Q880919437	(*N 880)	Cyrillic, Multilingual (old)	(919 437)	PC-USA
Q880959852	(*N 880)	Cyrillic, Multilingual (old)	(959 852)	PC-Latin 2
Q880959870	(*N 880)	Cyrillic, Multilingual (old)	(959 870)	Latin 2, Multilingual
Q880959912	(*N 880)	Cyrillic, Multilingual (old)	(959 912)	8-bit ASCII/ISO Latin 2
Q891337833	(*N 891)	PC-Korea	(337 833)	Korea
Q897A51850	(*N 897)	PC-Japan	(1106 850)	PC-Multilingual
Q897A7UA3T	(*N 897)	PC-Japan	(1172 1027)	Japan extended
Q897337290	(*N 897)	PC-Japan	(337 290)	Japan Katakana
Q897358037	(*N 897)	PC-Japan	(358 037)	USA/Canada

Table name	From	From text	To	To text
Q897640037	(*N 897)	PC-Japan	(640 037)	USA/Canada
Q897697037	(*N 897)	PC-Japan	(697 037)	USA/Canada
Q903A7W836	(*N 903)	PC-People's Republic of China	(1174 836)	People's Republic of China
Q903337836	(*N 903)	PC-People's Republic of China	(337 836)	People's Republic of China
Q904337037	(*N 904)	PC-People's Republic of China	(337 037)	USA/Canada
Q905A7AA3S	(*N 905)	PC-Turkey	(1152 1026)	Turkey Latin 5
Q905337037	(*N 905)	PC-Turkey	(337 037)	USA/Canada
Q905697500	(*N 905)	PC-Turkey	(697 500)	Multinational #5
Q905965853	(*N 905)	PC-Turkey	(965 853)	PC-Latin 3
Q905965857	(*N 905)	PC-Turkey	(965 857)	PC-Turkey
Q912A69A3R	(*N 912)	8-bit ASCII/ISO Latin 2	(1150 1025)	Cyrillic, Multilingual
Q912959870	(*N 912)	8-bit ASCII/ISO Latin 2	(959 870)	Latin 2, Multilingual
Q912960880	(*N 912)	8-bit ASCII/ISO Latin 2	(960 880)	Cyrillic, Multilingual (old)
Q915A51850	(*N 915)	8-bit ASCII/ISO Cyrillic	(1106 850)	PC-Multilingual
Q915A69A3R	(*N 915)	8-bit ASCII/ISO Cyrillic	(1150 1025)	Cyrillic, Multilingual
Q915A8Y437	(*N 915)	8-bit ASCII/ISO Cyrillic	(1212 437)	PC-USA
Q915A9L855	(*N 915)	8-bit ASCII/ISO Cyrillic	(1235 855)	PC-Cyrillic
Q915959870	(*N 915)	8-bit ASCII/ISO Cyrillic	(959 870)	Latin 2, Multilingual
Q915960880	(*N 915)	8-bit ASCII/ISO Cyrillic	(960 880)	Cyrillic, Multilingual (old)
Q920A7AA3S	(*N 920)	ASCII-Turkey	(1152 1026)	Turkey Latin 5
QA3BA69A3R	(*N 1009)	International Alphabet 5	(1150 1025)	Cyrillic, Multilingual
QA3BA7AA3S	(*N 1009)	International Alphabet 5	(1152 1026)	Turkey Latin 5
QA3BA7W836	(*N 1009)	International Alphabet 5	(1174 836)	People's Republic of China
QA3B103367	(*N 1009)	International Alphabet 5	(103 367)	ASCII
QA3B218423	(*N 1009)	International Alphabet 5	(218 423)	Greece
QA3B332290	(*N 1009)	International Alphabet 5	(332 290)	Japan Katakana
QA3B697037	(*N 1009)	International Alphabet 5	(697 037)	USA/Canada

Table name	From	From text	To	To text
QA3B697273	(*N 1009)	International Alphabet 5	(697 273)	Austria/Germany
QA3B697277	(*N 1009)	International Alphabet 5	(697 277)	Denmark/Norway
QA3B697278	(*N 1009)	International Alphabet 5	(697 278)	Finland/Sweden
QA3B697280	(*N 1009)	International Alphabet 5	(697 280)	Italy
QA3B697284	(*N 1009)	International Alphabet 5	(697 284)	Spain/Latin America
QA3B697297	(*N 1009)	International Alphabet 5	(697 297)	France
QA3B697500	(*N 1009)	International Alphabet 5	(697 500)	Multinational #5
QA3B697871	(*N 1009)	International Alphabet 5	(697 871)	Iceland
QA3B925875	(*N 1009)	International Alphabet 5	(925 875)	Greece
QA3B933833	(*N 1009)	International Alphabet 5	(933 833)	Korea
QA3B936836	(*N 1009)	International Alphabet 5	(936 836)	People's Republic of China
QA3B959870	(*N 1009)	International Alphabet 5	(959 870)	Latin 2, Multilingual
QA3B960880	(*N 1009)	International Alphabet 5	(960 880)	Cyrillic, Multilingual (old)
QA3D697500	(*N 1011)	VT220 Germany	(697 500)	Multinational #5
QA3E697500	(*N 1012)	VT220 Italy	(697 500)	Multinational #5
QA3M697500	(*N 1020)	VT220 Canadian/French	(697 500)	Multinational #5
QA3N697500	(*N 1021)	VT220 Switzerland	(697 500)	Multinational #5
QA3P697500	(*N 1023)	VT220 Spain	(697 500)	Multinational #5
QA3QA93A33	(*N 1024)	CCITT T.61 (EBCDIC)	(1252 1036)	CCITT T.61 IBM PC
QA3RA69855	(*N 1025)	Cyrillic, Multilingual	(1150 855)	PC-Cyrillic
QA3RA69915	(*N 1025)	Cyrillic, Multilingual	(1150 915)	8-bit ASCII/ISO Cyrillic
QA3RA7RA3B	(*N 1025)	Cyrillic, Multilingual	(1169 1009)	International Alphabet 5
QA3RA8C866	(*N 1025)	Cyrillic, Multilingual	(1190 866)	PC-Cyrillic #2
QA3RA9L855	(*N 1025)	Cyrillic, Multilingual	(1235 855)	PC-Cyrillic
QA3R337256	(*N 1025)	Cyrillic, Multilingual	(337 256)	Multinational #1
QA3R697037	(*N 1025)	Cyrillic, Multilingual	(697 037)	USA/Canada
QA3R697500	(*N 1025)	Cyrillic, Multilingual	(697 500)	Multinational #5
QA3R697850	(*N 1025)	Cyrillic, Multilingual	(697 850)	PC-Multilingual
QA3R919437	(*N 1025)	Cyrillic, Multilingual	(919 437)	PC-USA
QA3R959852	(*N 1025)	Cyrillic, Multilingual	(959 852)	PC-Latin 2

Table name	From	From text	To	To text
QA3R959870	(*N 1025)	Cyrillic, Multilingual	(959 870)	Latin 2, Multilingual
QA3R959912	(*N 1025)	Cyrillic, Multilingual	(959 912)	8-bit ASCII/ISO Latin 2
QA3R960880	(*N 1025)	Cyrillic, Multilingual	(960 880)	Cyrillic, Multilingual (old)
QA3SA7A857	(*N 1026)	Turkey Latin 5	(1152 857)	PC-Turkey
QA3SA7A920	(*N 1026)	Turkey Latin 5	(1152 920)	ASCII-Turkey
QA3SA7RA3B	(*N 1026)	Turkey Latin 5	(1169 1009)	International Alphabet 5
QA3S337037	(*N 1026)	Turkey Latin 5	(337 037)	USA/Canada
QA3S337256	(*N 1026)	Turkey Latin 5	(337 256)	Multinational #1
QA3S337500	(*N 1026)	Turkey Latin 5	(337 500)	Multinational #5
QA3S965905	(*N 1026)	Turkey Latin 5	(965 905)	PC-Turkey
QA3TA6G897	(*N 1027)	Japan extended	(1122 897)	PC-Japan
QA3TA7UA38	(*N 1027)	Japan extended	(1172 1041)	PC-Japan extended
QA3TA7U290	(*N 1027)	Japan extended	(1172 290)	Japan Katakana
QA3T697037	(*N 1027)	Japan extended	(697 037)	USA/Canada
QA3T697500	(*N 1027)	Japan extended	(697 500)	Multinational #5
QA33A93A3Q	(*N 1036)	CCITT T.61 IBM PC	(1252 1024)	CCITT T.61 (EBCDIC)
QA38A51850	(*N 1041)	PC-Japan extended	(1106 850)	PC-Multilingual
QA38A7UA3T	(*N 1041)	PC-Japan extended	(1172 1027)	Japan extended
QA38A7U290	(*N 1041)	PC-Japan extended	(1172 290)	Japan Katakana
QA5IA7V833	(*N 1088)	PC-Korean	(1173 833)	Korea
QA5KA7X037	(*N 1090)	VT100 Line drawing set	(1175 037)	USA/Canada
QA5K697037	(*N 1090)	VT100 Line drawing set	(697 037)	USA/Canada
QA5K697500	(*N 1090)	VT100 Line drawing set	(697 500)	Multinational #5
QA5RA0ZA5S	(*N 1097)	Farsi	(1285 1098)	PC-Farsi
QA5R337437	(*N 1097)	Farsi	(337 437)	PC-USA
QA5R337850	(*N 1097)	Farsi	(337 850)	PC-Multilingual
QA5R697037	(*N 1097)	Farsi	(697 037)	USA/Canada
QA5R697500	(*N 1097)	Farsi	(697 500)	Multinational #5
QA5SA86A5R	(*N 1098)	PC-Farsi	(1219 1097)	Farsi
QA5S919437	(*N 1098)	PC-Farsi	(919 437)	PC-USA
QA5S980850	(*N 1098)	PC-Farsi	(980 850)	PC-Multilingual
QA5UA7X037	(*N 1100)	VT220 Multinational	(1175 037)	USA/Canada
QA5U697037	(*N 1100)	VT220 Multinational	(697 037)	USA/Canada
QA5U697500	(*N 1100)	VT220 Multinational	(697 500)	Multinational #5
QA5V697500	(*N 1101)	VT220 British	(697 500)	Multinational #5
QA5W697500	(*N 1102)	VT220 Dutch	(697 500)	Multinational #5

Table name	From	From text	To	To text
QA5X697500	(*N 1103)	VT220 Finnish	(697 500)	Multinational #5
QA5Y697500	(*N 1104)	VT220 French	(697 500)	Multinational #5
QA5Z697500	(*N 1105)	VT220 Norwegian/Danish	(697 500)	Multinational #5
QA50A7W836	(*N 1115)	People's Republic of China GB	(1174 836)	People's Republic of China
QA51697500	(*N 1106)	VT220 Swedish	(697 500)	Multinational #5
QA52697500	(*N 1107)	VT220 Norwegian/Danish Alt	(697 500)	Multinational #5
QA57697037	(*N 1112)	Baltic, Multilingual	(697 037)	USA/Canada
QA57697500	(*N 1112)	Baltic, Multilingual	(697 500)	Multinational #5
QA59697037	(*N 1114)	Taiwan Industry (Big 5) code	(697 037)	USA/Canada
QA6G697037	(*N 1122)	Estonian	(697 037)	USA/Canada
QA6G697500	(*N 1122)	Estonian	(697 500)	Multinational #5
QA6OA51850	(*N 1130)	Vietnamese	(1106 850)	PC-Multilingual
QA6OBC8A99	(*N 1130)	Vietnamese	(BC8 1258)	MS Window, Vietnamese
QA6O697037	(*N 1130)	Vietnamese	(697 037)	USA/Canada
QA6O697500	(*N 1130)	Vietnamese	(697 500)	Multinational #5
QA6O697819	(*N 1130)	Vietnamese	(697 819)	8-bit ASCII/ISO Latin 1
QA6QA51850	(*N 1133)	Lao	(1106 850)	PC-Multilingual
QA6QBBJA6R	(*N 1133)	Lao	(1341 1133)	8-bit ASCII/ISO Lao
QA6Q697037	(*N 1133)	Lao	(697 037)	USA/Canada
QA6Q697500	(*N 1133)	Lao	(697 500)	Multinational #5
QA6Q697819	(*N 1133)	Lao	(697 819)	8-bit ASCII/ISO Latin 1
QA6RBBJA6Q	(*N 1133)	8-bit ASCII/ISO Lao	(1341 1133)	Lao
QA99BBEA6O	(*N 1258)	MS Window, Vietnamese	(1336 1130)	Vietnamese

Related tasks

"Job default coded character set identifier" on page 13
A job attribute, job default CCSID (DFTCCSID), is created for jobs with a CCSID of 65535. The DFTCCSID value is used by a system code when a CCSID other than 65535 is needed.

International DP 94 00103 (ASCII)

The figure shows International DP 94 00103 (ASCII).

A LA020000	B LB020000	C LC020000	D LD020000	E LE020000	F LF020000	G LG020000	H LH020000	I LI020000	J LJ020000	K LK020000	L LL020000	M LM020000
N LN020000	O LO020000	P LP020000	Q LQ020000	R LR020000	S LS020000	T LT020000	U LU020000	V LV020000	W LW020000	X LX020000	Y LY020000	Z LZ020000
a LA010000	b LB010000	c LC010000	d LD010000	e LE010000	f LF010000	g LG010000	h LH010000	i LI010000	j LJ010000	k LK010000	l LL010000	m LM010000
n LN010000	o LO010000	p LP010000	q LQ010000	r LR010000	s LS010000	t LT010000	u LU010000	v LV010000	w LW010000	x LX010000	y LY010000	z LZ010000
0 ND100000	1 ND010000	2 ND020000	3 ND030000	4 ND040000	5 ND050000	6 ND060000	7 ND070000	8 ND080000	9 ND090000			
+ SA010000	< SA030000	= SA040000	> SA050000	$ SC030000	` SD130000	^ SD150000	~ SD190000					
# SM010000	% SM020000	& SM030000	* SM040000	@ SM050000	[SM060000	\ SM070000	] SM080000	{ SM110000	| SM130000	} SM140000	! SP020000	" SP040000
' SP050000	(SP060000	) SP070000	, SP080000	_ SP090000	- SP100000	. SP110000	/ SP120000	: SP130000	; SP140000	? SP150000		

Character Set 00103

Character set 01169 (International Alphabet 5)

The figure shows character set 01169 (International Alphabet 5).

A LA020000	B LB020000	C LC020000	D LD020000	E LE020000	F LF020000	G LG020000	H LH020000	I LI020000	J LJ020000	K LK020000	L LL020000	M LM020000
N LN020000	O LO020000	P LP020000	Q LQ020000	R LR020000	S LS020000	T LT020000	U LU020000	V LV020000	W LW020000	X LX020000	Y LY020000	Z LZ020000
a LA010000	b LB010000	c LC010000	d LD010000	e LE010000	f LF010000	g LG010000	h LH010000	i LI010000	j LJ010000	k LK010000	l LL010000	m LM010000
n LN010000	o LO010000	p LP010000	q LQ010000	r LR010000	s LS010000	t LT010000	u LU010000	v LV010000	w LW010000	x LX010000	y LY010000	z LZ010000
0 ND100000	1 ND010000	2 ND020000	3 ND030000	4 ND040000	5 ND050000	6 ND060000	7 ND070000	8 ND080000	9 ND090000			
+ SA010000	< SA030000	= SA040000	> SA050000	¤ SC010000	` SD130000	^ SD150000	~ SD190000	# SM010000	% SM020000	& SM030000	* SM040000	@ SM050000
[SM060000	\ SM070000	] SM080000	{ SM110000	| SM130000	} SM140000	! SP020000	" SP040000	' SP050000	(SP060000	) SP070000	, SP080000	_ SP090000
- SP100000	. SP110000	/ SP120000	: SP130000	; SP140000	? SP150000							

Character Set 01169

Invariant character set

An *invariant character set* is a character set, such as the syntactic character set, whose code point assignments do not change from code page to code page. The table illustrates the invariant character set (character set 00640) on the i5/OS operating system.

With only a few exceptions, each EBCDIC code page contains a common set of graphic characters. Within an encoding scheme, the common characters can be found at the same code points. The following EBCDIC code pages are exceptions:

- EBCDIC code page 290 has Katakana characters at the code points where lowercase a through z are in the invariant character set
- EBCDIC code page 905, where the quotation mark (") is not at the same code point as it is in the invariant character set
- EBCDIC code page 1026, where the quotation mark (") is not at the same code point as it is in the invariant character set
- EBCDIC code page 420 does not contain the invariant character SM040000 (asterisk (*)). However, code page 420 has a similar character named SM040007 (*) at the same code point where SM040000 normally exists.

- EBCDIC code page 420 does not contain the invariant character SM020000 (percent sign (%)). However, code page 420 has a similar character named SM020007 (٪) at the same code point where SM020000 normally exists.

Invariant character set symbols

GCGID	Description	Graphic character
LA010000	Latin small letter a	a
LA020000	Latin capital letter A	A
LB010000	Latin small letter b	b
LB020000	Latin capital letter B	B
LC010000	Latin small letter c	c
LC020000	Latin capital letter C	C
LD010000	Latin small letter d	d
LD020000	Latin capital letter D	D
LE010000	Latin small letter e	e
LE020000	Latin capital letter E	E
LF010000	Latin small letter f	f
LF020000	Latin capital letter F	F
LG010000	Latin small letter g	g
LG020000	Latin capital letter G	G
LH010000	Latin small letter h	h
LH020000	Latin capital letter H	H
LI010000	Latin small letter i	i
LI020000	Latin capital letter I	I
LJ010000	Latin small letter j	j
LJ020000	Latin capital letter J	J
LK010000	Latin small letter k	k
LK020000	Latin capital letter K	K
LL010000	Latin small letter l	l
LL020000	Latin capital letter L	L
LM010000	Latin small letter m	m
LM020000	Latin capital letter M	M
LN010000	Latin small letter n	n
LN020000	Latin capital letter N	N
LO010000	Latin small letter o	o
LO020000	Latin capital letter O	O
LP010000	Latin small letter p	p
LP020000	Latin capital letter P	P
LQ010000	Latin small letter q	q
LQ020000	Latin capital letter Q	Q
LR010000	Latin small letter r	r
LR020000	Latin capital letter R	R

GCGID	Description	Graphic character
LS010000	Latin small letter s	s
LS020000	Latin capital letter S	S
LT010000	Latin small letter t	t
LT020000	Latin capital letter T	T
LU010000	Latin small letter u	u
LU020000	Latin capital letter U	U
LV010000	Latin small letter v	v
LV020000	Latin capital letter V	V
LW010000	Latin small letter w	w
LW020000	Latin capital letter W	W
LX010000	Latin small letter x	x
LX020000	Latin capital letter X	X
LY010000	Latin small letter y	y
LY020000	Latin capital letter Y	Y
LZ010000	Latin small letter z	z
LZ020000	Latin capital letter Z	Z
ND100000	Arabic number zero	0
ND010000	Arabic number one	1
ND020000	Arabic number two	2
ND030000	Arabic number three	3
ND040000	Arabic number four	4
ND050000	Arabic number five	5
ND060000	Arabic number six	6
ND070000	Arabic number seven	7
ND080000	Arabic number eight	8
ND090000	Arabic number nine	9
SA010000	Plus sign	+
SA030000	Less than sign	<
SA040000	Equal sign	=
SA050000	Greater than sign	>
SM020000	Percent sign	%
SM030000	Ampersand	&
SM040000	Asterisk	*
SP040000	Straight quotation mark	″
SP050000	Straight single quotation mark	′
SP060000	Left parenthesis	(
SP070000	Right parenthesis	)
SP080000	Comma	,
SP090000	Underscore	_
SP100000	Hyphen	-
SP110000	Period	.

GCGID	Description	Graphic character
SP120000	Slash right	/
SP130000	Colon	:
SP140000	Semicolon	;
SP150000	Question mark	?

Related concepts

"Application part names" on page 61
When you want to enable your application for different languages and countries, consider the environments of the target systems in your naming conventions.

"Coding globalized applications with high-level languages" on page 97
Your major goal must be to have only one general set of running code that is common for all language versions and to make your programs table-driven as much as possible.

Monocase tables

Here is a list of monocase tables on the i5/OS operating system.

Code page	Table object for monocase	Description
00037	Q037	USA/Canada (EBCDIC)
00256	Q256	International 1 (EBCDIC)
00260	Q260	Canadian French (EBCDIC)
00273	Q273	Germany/Austria (EBCDIC)
00277	Q277	Denmark, Norway (EBCDIC)
00278	Q278	Finland, Sweden (EBCDIC)
00280	Q280	Italy (EBCDIC)
00281	Q281	Japan Latin (EBCDIC)
00284	Q284	Spain/Latin America (EBCDIC)
00285	Q285	United Kingdom (EBCDIC)
00290	Q290	Japanese Katakana extended
00297	Q297	France (EBCDIC)
00420	Q420	Arabic Bilingual (EBCDIC)
00423	Q423	Greece (EBCDIC)
00424	Q424	Israel (Hebrew)
00437	Q437	USA (IBM personal computer)
00500	Q500	Multilingual #5
00833	Q833	Korean extended (EBCDIC)
00836	Q836	Simplified Chinese extended (EBCDIC)
00838	Q838	Thai extended (EBCDIC)
00850	Q850	Multilingual (IBM personal computer)
00851	Q851	Greece (IBM personal computer)
00857	Q857	Turkey (ISO 8859-5)
00860	Q860	Portugal (IBM personal computer)
00861	Q861	Iceland (IBM personal computer)

Code page	Table object for monocase	Description
00862	Q862	Israel (IBM personal computer)
00863	Q863	Canadian French (IBM personal computer)
00864	Q864	Arabic (IBM personal computer)
00865	Q865	Nordic (IBM personal computer)
00870	Q870	Multilingual (ISO 8859-2)
00871	Q871	Iceland (EBCDIC)
00875	Q875	Greece (EBCDIC)
00880	Q880	Cyrillic, Multilingual
00891	Q891	Korea (IBM personal computer)
00897	Q897	Japan PC #1 (IBM Personal Computer)
00903	Q903	People's Republic of China (IBM personal computer)
00904	Q904	Taiwan (IBM personal computer)
00905	Q905	PC-Turkey (ISO 8859-9)
01025	QA3R	Cyrillic, multilingual (EBCDIC)
01026	QA3S	Turkey (ISO 8859-9)
01027	QA3T	Japanese (Latin) extended (EBCDIC)
01097	QA5R	Farsi bilingual (EBCDIC)

Portable character set

The X/Open portable character set is a superset of the IBM invariant character set (00640), which includes symbols not represented in the IBM invariant character set 00640.

The portable character set includes the following 13 symbols that are not represented in the IBM invariant character set 00640.

GCGID	Description	Graphic character
SC030000	Dollar	$
SD110000	Accent acute	´
SD150000	Caret	^
SD190000	Tilde	~
SM010000	Number sign	#
SM050000	At sign	@
SM060000	Left bracket	[
SM070000	Back slash	\
SM080000	Right bracket	]
SM110000	Left brace	{
SM130000	Logical or	|
SM140000	Right brace	}
SP020000	Exclamation point	!

The portable character set also includes the space character and control characters representing the horizontal tab, the vertical tab, and form feed.

Related concepts

"Locale symbolic names" on page 220
The i5/OS operating system supports locale symbolic names based on predefined names from the X/Open Standard portable character set.

Syntactic/invariant character set 00640

With only a few exceptions, each EBCDIC code page contains a common set of graphic characters. Within an encoding scheme, the common characters can be found at the same code points. However, there are exceptions.

The exceptions are as follows:

- EBCDIC code page 290 has Katakana characters at the code points where lowercase a through z are in the invariant character set.
- EBCDIC code page 905 where the quotation mark (") is not at the same code point as it is in the invariant character set.
- EBCDIC code page 1026 where the quotation mark (") is not at the same code point as it is in the invariant character set.
- EBCDIC code page 420 does not contain the invariant character SM040000 (asterisk (*)). However, code page 420 has a similar character named SM040007 (*) at the same code point where SM040000 normally exists.
- EBCDIC code page 420 does not contain the invariant character SM020000 (percent sign (%)). However, code page 420 has a similar character named SM020007 (%) at the same code point where SM020000 normally exists.

A	B	C	D	E	F	G	H	I	J	K	L	M
LA020000	LB020000	LC020000	LD020000	LE020000	LF020000	LG020000	LH020000	LI020000	LJ020000	LK020000	LL020000	LM020000
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
LN020000	LO020000	LP020000	LQ020000	LR020000	LS020000	LT020000	LU020000	LV020000	LW020000	LX020000	LY020000	LZ020000
a	b	c	d	e	f	g	h	i	j	k	l	m
LA010000	LB010000	LC010000	LD010000	LE010000	LF010000	LG010000	LH010000	LI010000	LJ010000	LK010000	LL010000	LM010000
n	o	p	q	r	s	t	u	v	w	x	y	z
LN010000	LO010000	LP010000	LQ010000	LR010000	LS010000	LT010000	LU010000	LV010000	LW010000	LX010000	LY010000	LZ010000
0	1	2	3	4	5	6	7	8	9			
ND100000	ND010000	ND020000	ND030000	ND040000	ND050000	ND060000	ND070000	ND080000	ND090000			
+	<	=	>	%	&	*	"	'	(	)	,	_
SA010000	SA030000	SA040000	SA050000	SM020000	SM030000	SM040000	SP040000	SP050000	SP060000	SP070000	SP080000	SP090000
-	.	/	:	;	?							
SP100000	SP110000	SP120000	SP130000	SP140000	SP150000							

Character Set 00640

T.61 Character Set 01252

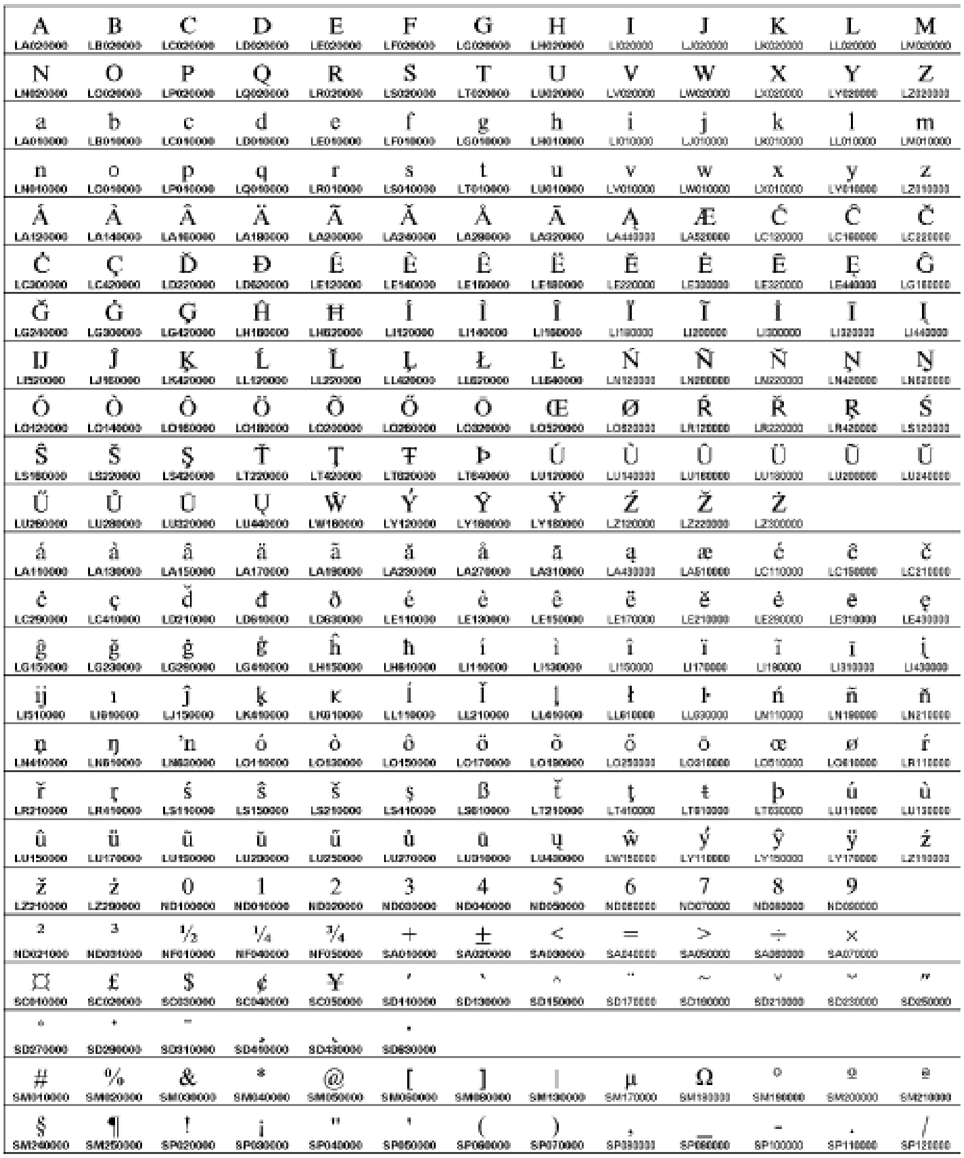
This figure shows T.61 Character Set 01252.

A LA020000	B LB020000	C LC020000	D LD020000	E LE020000	F LF020000	G LG020000	H LH020000	I LI020000	J LJ020000	K LK020000	L LL020000	M LM020000
N LN020000	O LO020000	P LP020000	Q LQ020000	R LR020000	S LS020000	T LT020000	U LU020000	V LV020000	W LW020000	X LX020000	Y LY020000	Z LZ020000
a LA010000	b LB010000	c LC010000	d LD010000	e LE010000	f LF010000	g LG010000	h LH010000	i LI010000	j LJ010000	k LK010000	l LL010000	m LM010000
n LN010000	o LO010000	p LP010000	q LQ010000	r LR010000	s LS010000	t LT010000	u LU010000	v LV010000	w LW010000	x LX010000	y LY010000	z LZ010000
Æ LA520000	Đ LD620000	Ħ LH620000	Ĳ LI520000	Ł LL620000	Ŀ LL640000	Ŋ LN620000	Œ LO520000	Ø LO620000	Ŧ LT620000	Þ LT640000		
æ LA510000	đ LD610000	ð LD630000	ħ LH610000	ĳ LI510000	ı LI610000	ĸ LK610000	ł LL610000	ŀ LL630000	ŋ LN610000	ŉ LN630000	œ LO510000	ø LO610000
ß LS610000	ŧ LT610000	þ LT630000										
0 ND100000	1 ND010000	2 ND020000	3 ND030000	4 ND040000	5 ND050000	6 ND060000	7 ND070000	8 ND080000	9 ND090000			
^{2} ND021000	^{3} ND031000	½ NF010000	¼ NF040000	¾ NF050000	+ SA010000	± SA020000	< SA030000	= SA040000	> SA050000	÷ SA060000	× SA070000	¤ SC010000
£ SC020000	$ SC030000	¢ SC040000	¥ SC050000	´ SD110000	` SD130000	^ SD150000	¨ SD170000	~ SD190000	ˇ SD210000	˘ SD230000	˝ SD250000	˚ SD270000
˙ SD290000	¯ SD310000	¸ SD410000	˛ SD430000	· SD630000	# SM010000	% SM020000	& SM030000	* SM040000	@ SM050000	[SM060000	] SM080000	| SM130000
µ SM170000	Ω SM180000	° SM190000	º SM200000	ª SM210000	§ SM240000	¶ SM250000	! SP020000	¡ SP030000	" SP040000	' SP050000	(SP060000	) SP070000
, SP080000	_ SP090000	‗ SP090000	- SP100000	. SP110000	/ SP120000	: SP130000	; SP140000	? SP150000	¿ SP160000	« SP170000	» SP180000	

Character Set 01252

T.61 Character Repertoire 01253

The figure shows T.61 Character Repertoire 01253.



Character Set 01253 (Sheet 1 of 2)

Character Set 01253 (Sheet 2 of 2)

T.61 graphic character conversions

The table contains a list of T.61 conversions supported on the i5/OS operating system.

The *N character in the From Value field means any character set is used.

These conversion tables are used to convert data to and from character set 01253 on code page 01024 to another supported character set and code page.

Table name	From description	From value	To description	To value
International alphabet 5	(*N 01009)	CCITT T.61 (EBCDIC)	(01253 01024)	
CCITT T.61 (EBCDIC)	(*N 01024)	Cyrillic, multilingual	(01150 01025)	
CCITT T.61 (EBCDIC)	(*N 01024)	Turkey EBCDIC	(01152 01026)	
CCITT T.61 (EBCDIC)	(*N 01024)	International alphabet 5	(01169 01009)	
CCITT T.61 (EBCDIC)	(*N 01024)	People's Republic of China	(01174 00836)	
CCITT T.61 (EBCDIC)	(*N 01024)	ASCII	(00103 00367)	
CCITT T.61 (EBCDIC)	(*N 01024)	Greece	(00218 00423)	
CCITT T.61 (EBCDIC)	(*N 01024)	Japan Katakana	(00332 00290)	
CCITT T.61 (EBCDIC)	(*N 01024)	USA/Canada	(00697 00037)	
CCITT T.61 (EBCDIC)	(*N 01024)	Austria/Germany	(00697 00273)	
CCITT T.61 (EBCDIC)	(*N 01024)	Denmark/Norway	(00697 00277)	
CCITT T.61 (EBCDIC)	(*N 01024)	Finland/Sweden	(00697 00278)	
CCITT T.61 (EBCDIC)	(*N 01024)	Italy	(00697 00280)	
CCITT T.61 (EBCDIC)	(*N 01024)	Spain/Latin America	(00697 00284)	
CCITT T.61 (EBCDIC)	(*N 01024)	France	(00697 00297)	
CCITT T.61 (EBCDIC)	(*N 01024)	Multinational #5	(00697 00500)	
CCITT T.61 (EBCDIC)	(*N 01024)	Iceland	(00697 00871)	
CCITT T.61 (EBCDIC)	(*N 01024)	Greece	(00925 00875)	
CCITT T.61 (EBCDIC)	(*N 01024)	Korea	(00933 00833)	
CCITT T.61 (EBCDIC)	(*N 01024)	People's Republic of China	(00936 00836)	
CCITT T.61 (EBCDIC)	(*N 01024)	Latin 2, multilingual	(00959 00870)	
CCITT T.61 (EBCDIC)	(*N 01024)	Cyrillic, multilingual (old)	(00960 00880)	
Cyrillic, multilingual	(*N 01025)	CCITT T.61 (EBCDIC)	(01253 001024)	
Turkey EBCDIC	(*N 01026)	CCITT T.61 (EBCDIC)	(01253 01024)	
USA/Canada	(*N 00037)	CCITT T.61 (EBCDIC)	(01253 01024)	
Austria/Germany	(*N 00273)	CCITT T.61 (EBCDIC)	(01253 01024)	
Denmark/Norway	(*N 00277)	CCITT T.61 (EBCDIC)	(01253 01024)	
Finland/Sweden	(*N 00278)	CCITT T.61 (EBCDIC)	(01253 01024)	
Italy	(*N 00280)	CCITT T.61 (EBCDIC)	(01253 01024)	
Spain/Latin America	(*N 00284)	CCITT T.61 (EBCDIC)	(01253 01024)	
Japan Katakana	(*N 00290)	CCITT T.61 (EBCDIC)	(01253 01024)	
France	(*N 00297)	CCITT T.61 (EBCDIC)	(01253 01024)	
ASCII	(*N 00367)	CCITT T.61 (EBCDIC)	(01253 01024)	
Greece	(*N 00423)	CCITT T.61 (EBCDIC)	(01253 01024)	
Multinational #5	(*N 00500)	CCITT T.61 (EBCDIC)	(01253 01024)	
Korea	(*N 00833)	CCITT T.61 (EBCDIC)	(01253 01024)	

Table name	From description	From value	To description	To value
People’s Republic of China	(*N 00836)	CCITT T.61 (EBCDIC)	(01253 01024)	
Latin 2, multilingual	(*N 00870)	CCITT T.61 (EBCDIC)	(01253 01024)	
Iceland	(*N 00871)	CCITT T.61 (EBCDIC)	(01253 01024)	
Greece	(*N 00875)	CCITT T.61 (EBCDIC)	(01253 01024)	
Cyrillic, multilingual (old)	(*N 00880)	CCITT T.61 (EBCDIC)	(01253 01024)	

CCSID reference information

Coded character set identifier (CCSID) is a 16-bit number that includes a specific set of encoding scheme identifiers, character set identifiers, code page identifiers, and other information that uniquely identifies the coded graphic-character representation.

Related concepts

“Character data integrity” on page 6
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.

“Automatic character set and code page conversion” on page 37
The i5/OS operating system provides automatic conversion between character set and code pages for all applications that are enabled for national language support.

“Coding globalized applications that use bidirectional data” on page 106
When you are developing NLV-enabled applications, you should consider some specific restrictions on bidirectional languages.

CCSID values defined on i5/OS

This table lists the coded character set identifiers (CCSIDs) that are defined on the i5/OS operating system.

CCSID	Encoding	Description
00037	1100	US, Canada, Netherlands, Portugal, Brazil, New Zealand, Australia
00256	1100	Netherlands
00273	1100	Austria, Germany
00277	1100	Denmark, Norway
00278	1100	Finland, Sweden
00280	1100	Italy
00284	1100	Spanish, Latin America
00285	1100	United Kingdom
00290	1100	Japan Katakana
00297	1100	France
00300	1200	Japan English
00301	2200	Japanese PC Data
00367	5100	ANSI X3.4 ASCII standard; USA
00420	1100	Arabic-speaking countries
00423	1100	Greece

CCSID	Encoding	Description
00424	1100	Hebrew
00425	1100	Arabic-speaking countries
00437	2100	PC Data; PC Base; USA
00500	1100	Belgium, Canada, Switzerland, International Latin-1
00720	2100	MS-DOS Arabic
00737	2100	MS-DOS Greek PC-Data
00775	2100	MS-DOS Baltic PC-Data
00813	4100	ISO 8859-7; Greek/Latin
00819	4100	ISO 8859-1; Latin Alphabet No. 1
00833	1100	Korea (extended range)
00834	1200	Korea host double byte (including 1880 UDC)
00835	1200	Traditional Chinese host double byte (including 6204 UDC)
00836	1100	Simplified Chinese (extended range)
00837	1200	Simplified Chinese
00838	1100	Thailand (extended range)
00850	2100	PC Data; MLP 222 Latin Alphabet 1
00851	2100	PC Data; Greek
00852	2100	PC Data; Latin-2 Multilingual
00855	2100	PC Data; ROECE Cyrillic
00857	2100	PC Data; Turkey Latin #5
00858	2100	PC Data: MLP 222; Latin Alphabet Number 1 w/euro; Latin-1 Countries
00860	2100	PC Data; Portugal
00861	2100	PC Data; Iceland
00862	2100	PC Data; Hebrew
00863	2100	PC Data; Canada
00864	2100	PC Data; Arabic
00865	2100	PC Data; Denmark, Norway
00866	2100	PC Data; Cyrillic #2 - Personal Computer
00868	2100	PC Data: Urdu
00869	2100	PC Data; Greek
00870	1100	Latin-2 Multilingual
00871	1100	Iceland
00874	2100	Thai PC Data
00875	1100	Greece
00878	4105	Russian Internet KOI8-R Cyrillic
00880	1100	Cyrillic Multilingual
00891	2100	Korean PC Data (non-extended)
00897	2100	Japanese PC Data (non-extended)
00903	2100	Simplified Chinese PC Data (non-extended)
00904	2100	Traditional Chinese PC Data
00905	1100	Turkey Latin-3

CCSID	Encoding	Description
00912	4100	ISO 8859-2; ROECE Latin-2 Multilingual
00914	4100	Latin 4 - ISO 8859-4
00915	4100	ISO 8859-5; Cyrillic; 8-bit ISO
00916	4100	ISO 8859-8; Hebrew
00918	1100	Urdu EBCDIC
00920	4100	ISO 8859-9; Latin 5
00921	4100	Baltic, 8-bit (ISO 8859-13)
00922	4100	Estonia, 8-bit (ISO)
00923	4100	ISO 8859-15: Latin Alphabet with euro
00924	1100	Latin 9 EBCDIC
00926	2200	Korean PC Data DBCS, UDC 1880
00927	2200	Traditional Chinese PC Data DBCS, UDC 6204
00928	2200	Simplified Chinese PC Data DBCS, UDC 1880
00930	1301	Japan Katakana (extended range) 4370 UDC (User Defined Characters)
00932	2300	Japan PC Data Mixed
00933	1301	Korea (extended range), 1880 UDC
00934	2300	Korean PC Data
00935	1301	Simplified Chinese (extended range)
00936	2300	Simplified Chinese (non-extended)
00937	1301	Traditional Chinese (extended range)
00938	2300	Traditional Chinese (non-extended)
00939	1301	Japan English (extended range) 4370 UDC
00941	2200	Japanese DBCS PC for Open environment (Multi-vendor code): 6878 JIS X 0208-1990 characters, 386 IBM selected characters, 1880 IBM UDC (X’F040’ to X’F9FC’)
00942	2300	Japanese PC Data Mixed
00943	2300	Japanese PC Data Mixed for Open environment (Multi-vendor code): 6878 JIS X 0208-1990 characters, 386 IBM selected DBCS characters, 1880 UDC (X’F040’ to X’F9FC’)
00944	2300	Korean PC Data Mixed
00946	2300	Simplified Chinese PC Data Mixed
00947	2200	ASCII Double-byte
00948	2300	Traditional Chinese PC Data Mixed 6204 UDC (User Defined Characters)
00949	2300	Republic of Korea National Standard Graphic Character Set (KS) PC Data mixed-byte including 1800 UDC
00950	2300	Traditional Chinese PC Data Mixed for Big5
00951	2200	Republic of Korea National Standard Graphic Character Set (KS) PC Data double-byte including 1800 UDC
00954	4403	Japanese EUC; G0 - JIS X201 Roman set (00895); G1 - JIS X208-1990 set (00952); G2 - JIS X201 Katakana set (04992); G3 - JIS X212 set (00953)
00956	5404	JIS X201 Roman for CP 00895; JIS X208-1983 for CP 00952
00957	5404	JIS X201 Roman for CP 00895; JIS X208-1978 for CP 00955
00958	5404	ASCII for CP 00367; JIS X208-1983 for CP 00952

CCSID	Encoding	Description
00959	5404	ASCII for CP 00367; JIS X208-1978 for CP 00955
00964	4403	G0 - ASCII for CP 00367; G1- CNS 11643 plane 1 for CP 960
00965	5404	ASCII for CP 00367; CNS 11643 plane 1 for CP 960
00970	4403	G0 ASCII for CP 00367; G1 KSC X5601-1989 (including 188 UDCs) for CP 971
00971	8200	Korean EUC, G1 - KS C5601-1989 (including 188 UDC)
01008	4100	Arabic 8-bit ISO/ASCII
01009	5100	IS0-7: IRV
01010	5100	ISO-7; France
01011	5100	ISO-7; Germany
01012	5100	ISO-7; Italy
01013	5100	ISO-7; United Kingdom
01014	5100	ISO-7; Spain
01015	5100	ISO-7; Portugal
01016	5100	ISO-7; Norway
01017	5100	ISO-7; Denmark
01018	5100	ISO-7; Finland and Sweden
01019	5100	ISO-7; Belgium and Netherlands
01025	1100	Cyrillic Multilingual
01026	1100	Turkey Latin 5 CECP
01027	1100	Japan English (extended range)
01040	2100	Korean Latin PC Data extended
01041	2100	Japanese PC Data extended
01042	2100	Simplified Chinese PC Data extended
01043	2100	Traditional Chinese PC Data extended
01046	2100	PC Data - Arabic Extended
01051	4100	HP Emulation(for use with Latin 1). GCGID SF150000 is mapped to a control X'7F'
01088	2100	Korean PC Data single-byte
01089	4100	ISO 8859-6: Arabic (string type 5)
01097	1100	Farsi
01098	2100	Farsi (IBM-PC)
01112	1100	Baltic, Multilingual
01114	2100	Traditional Chinese, Taiwan Industry Graphic Character Set (Big5)
01115	2100	Simplified Chinese National Standard (GB), personal computer SBCS
01122	1100	Estonia
01123	1100	Cyrillic Ukraine EBCDIC
01124	4100	Cyrillic Ukraine 8-Bit
01125	2100	Cyrillic Ukraine PC-Data
01126	2100	Windows Korean PC Data Single-Byte
01129	4100	ISO-8 Vietnamese

CCSID	Encoding	Description
01130	1100	EBCDIC Vietnamese
01131	2100	Cyrillic Belarus PC-Data
01132	1100	EBCDIC Lao
01133	4100	ISO-8 Lao
01137	1100	Devanagari EBCDIC
01140	1100	ECECP: USA, Canada, Netherlands, Portugal, Brazil, Australia, New Zealand
01141	1100	ECECP: Austria, Germany
01142	1100	ECECP: Denmark, Norway
01143	1100	ECECP: Finland, Sweden
01144	1100	ECECP: Italy
01145	1100	ECECP: Spain, Latin America (Spanish)
01146	1100	ECECP: United Kingdom
01147	1100	ECECP: France
01148	1100	ECECP: International 1
01149	1100	ECECP: Iceland
01153	1100	Latin-2 - EBCDIC Multilingual with euro
01154	1100	Cyrillic Multilingual with euro
01155	1100	Turkey Latin 5 with euro
01156	1100	Baltic, Multilingual with euro
01157	1100	Estonia EBCDIC with euro
01158	1100	Cyrillic Ukraine EBCDIC with euro
01160	1100	Thai host with euro
01164	1100	EBCDIC Vietnamese with euro
01200	7200	UTF-16 Unicode, big endian
01208	7807	UTF-8
01250	4105	Windows, Latin 2
01251	4105	Windows, Cyrillic
01252	4105	Windows,Latin 1
01253	4105	Windows, Greek
01254	4105	Windows, Turkish
01255	4105	Windows, Hebrew
01256	4105	Windows, Arabic
01257	4105	Windows, Baltic Rim
01258	4105	MS Windows, Vietnamese
01275	4105	Apple Latin-1
01280	4105	Apple Greek
01281	4105	Apple Turkey
01282	4105	Apple Central European (Latin-2)
01283	4105	Apple Cyrillic
01362	2200	Windows Korean PC DBCS-PC, including 11 172 full hangul

CCSID	Encoding	Description
01363	2300	Windows Korean PC Mixed, including 11 172 full hangul
01364	1301	Korean host mixed extended including 11 172 full hangul
01380	2200	Simplified Chinese, People's Republic of China National Standard (GB), personal computer DBCS
01381	2300	Simplified Chinese, People's Republic of China National Standard (GB) personal computer mixed SBCS and DBCS
01382	8200	Simplified Chinese DBCS PC GB 2312-80 set, including 31 IBM selected and 1360 UDC.
01383	4403	Simplified Chinese, EUC • G0 set; ASCII • G1 set; GB 2312-80 set (1382)
01385	2200	Simplified Chinese DBCS-PC GBK, all GBK character set and others
01386	2300	Simplified Chinese PC Data GBK mixed, all GBK character set and others
01388	1301	Simplified Chinese DBCS- GB 18030 Host with UDCs and Uygur extension.
01399	1301	Japanese Latin-Kanji Host Mixed including 4370 UDC, Extended SBCS (includes SBCS and DBCS euro)
04396	1200	Japanese Host DB including 1880
04930	1200	Korean DBCS-Host extended including 11 172 full hangul
04933	1200	Simplified Chinese DBCS Host (GBK), all GBK character set and others
04948	2100	Latin 2 PC Data Multilingual
04951	2100	Cyrillic PC Data Multilingual
04952	2100	Hebrew PC Data
04953	2100	Turkey PC Data Latin 5
04960	2100	Arabic PC Data
04965	2100	Greek PC Data
04970	2100	Thai PC Data Single-Byte
04971	1100	Greek (including euro)
05026	1301	Japan Katakana (extended range) 1880 UDC
05035	1301	Japan English (extended range) 1880 UDC
05050	4403	G0 - JIS X201 Roman for CP 895; G1 JIS X208-1990 for CP 952
05052	5404	JIS X201 Roman for CP 895; JIS X208-1983 for CP 952
05053	5404	JIS X201 Roman for CP 895; JIS X208-1978 for CP 955
05054	5404	ASCII for CP 367; JIS X208-1983 for CP 952
05055	5404	ASCII for CP 367; JIS X208-1978 for CP 955
05123	1100	Japanese Latin Host Extended SBCS (includes euro)
05210	2100	Simplified Chinese PC Data Single-Byte (GBK), growing CS
05348	4105	Windows, Latin 1 with euro
08612	01100	Arabic (base shapes only)
09030	1100	Thai Host Extended SBCS
09056	2100	PC Data: Arabic PC Storage/Interchange
09066	2100	Thai PC Data Extended SBCS
12708	1100	Arabic (base shapes, Lamaleph ligatures and Hindi digits) (string type 7)

CCSID	Encoding	Description
13121	1100	Korean Host Extended SBCS
13124	1100	Simplified Chinese Host Data Single-Byte (GBK) equivalent to Simplified Chinese Host Data Single-Byte (GB) except growing CS
13488	7200	ISO/IEC 10646 Universal Coded Character Set Level 2 (UCS-2)
16684	1200	Japanese Latin Host Double-Byte including 4370 UDC (includes euro)
17354	5404	G0 - ASCII for CP 00367; G1 - KSC X5601-1989 (including 188 UDCs) for CP 00971
25546	5409	Korean 2022-KR TCP, ASCII, KS C5601-1989 (includes 188 UDC, RFC1557 using SO/SI)
28709	1100	Traditional Chinese (extended range)
33722	4403	Japanese EUC • G0; JIS X201 Roman set (00895) • G1; JIS X208-1990 set (00952) • G2; JIS X201 Katakana set (04992) • G3; JIS X212 set (09145)
57345	5404	All Japanese 2022 characters
61952	7200	i5/OS specific (old CCSID for UCS). Use of 13488 is recommended instead.
62210	4100	i5/OS specific ISO 8859-8; Hebrew, string type 4.
62211	1100	i5/OS specific EBCDIC; Hebrew, string type 5
62215	4105	i5/OS specific MS Windows; Hebrew, string type 4
62218	2100	i5/OS specific PC data; Arabic, string type 4
62222	4100	i5/OS specific ISO 8859-9; Hebrew, string type 6
62223	4105	i5/OS specific MS Windows; Hebrew, string type 6
62224	1100	i5/OS specific EBCDIC; Arabic, string type 6
62228	4105	i5/OS specific MS Windows; Arabic, string type 6
62235	1100	i5/OS specific EBCDIC; Hebrew, string type 6
62238	4100	i5/OS specific ISO 8859-9; Hebrew, string type 10
62239	4105	i5/OS specific MS Windows; Hebrew, string type 10
62245	1100	i5/OS specific EBCDIC; Hebrew, string type 10
65534		Look at lower level CCSID
65535		Special value indicating data is hex and should not be converted. This is the default for the QCCSID system value.

Related information

Globalization: iSeries CCSID Information

IBM Coded Character Sets and Related Resources Web site

Supported CCSID mappings

The i5/OS operating system can dynamically create CCSID mappings on demand by using Unicode. These CCSID mappings are predefined for better performance or mapping customization.

| **Note:** Some of the mappings that involve mapping from a mixed encoding (such as CCSID 937) to a
| single byte encoding (such as CCSID 37) might return errors if any ideographic characters exist in
| the data.

From CCSID	To CCSID
00037	00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00367, 00420, 00423, 00424, 00425, 00437, 00500, 00720, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00901, 00902, 00903, 00904, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00924, 00930, 00933, 00935, 00937, 00939, 00948, 01025, 01026, 01027, 01041, 01043, 01051, 01088, 01089, 01097, 01098, 01112, 01114, 01115, 01122, 01123, 01124, 01126, 01130, 01131, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01250, 01251, 01252, 01253, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01364, 01388, 01399, 04909, 04970, 04971, 05026, 05035, 05123, 05348, 08612, 09030, 12708, 13121, 13124, 13488, 28709, 61952, 62211, 62224, 62235, 62245, 62251
00256	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00500, 00737, 00775, 00819, 00833, 00836, 00838, 00850, 00870, 00871, 00875, 00880, 00905, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01112, 01122, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 08612, 09030, 13121, 13124, 13488, 28709, 61952
00273	00037, 00256, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01250, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00277	00037, 00256, 00273, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00278	00037, 00256, 00273, 00277, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00280	00037, 00256, 00273, 00277, 00278, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00284	00037, 00256, 00273, 00277, 00278, 00280, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00285	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00290	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00737, 00775, 00819, 00833, 00836, 00850, 00871, 00895, 00897, 00930, 00933, 00935, 00937, 00939, 00942, 00943, 00954, 01025, 01027, 01041, 01112, 01122, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 09030, 13121, 13124, 13488, 28709, 61952

From CCSID	To CCSID
00297	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 09030, 13121, 13124, 13488, 28709, 61952
00300	00301, 00930, 00939, 00941, 01200, 01208, 04396, 05026, 05035, 13488, 61952
00301	00300, 01200, 01208, 04396, 13488, 16684, 61952
00367	00037, 00500, 00833, 01200, 01208, 13121, 13124, 13488, 61952
00420	00037, 00256, 00425, 00500, 00720, 00737, 00775, 00819, 00850, 00864, 00937, 01008, 01046, 01089, 01112, 01122, 01200, 01208, 01252, 01256, 04960, 08612, 09030, 09056, 12708, 13488, 28709, 61952, 62218, 62224, 62228, 62248, 62251
00423	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00737, 00775, 00813, 00819, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00937, 01025, 01026, 01027, 01041, 01042, 01043, 01112, 01122, 01200, 01208, 01252, 01253, 04965, 04971, 05123, 09030, 13488, 28709, 61952
00424	00037, 00256, 00500, 00737, 00775, 00819, 00850, 00862, 00916, 00937, 01112, 01122, 01200, 01208, 01252, 01255, 04952, 09030, 13488, 28709, 61952, 62210, 62211, 62215, 62222, 62223, 62235, 62238, 62239, 62245
00425	00037, 00420, 00500, 00819, 00864, 01046, 01089, 01140, 01148, 01200, 01208, 01252, 01256, 08612, 12708, 13488, 61952, 62224, 62228
00437	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00852, 00855, 00857, 00860, 00861, 00863, 00865, 00866, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00937, 01025, 01026, 01027, 01042, 01051, 01097, 01098, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01257, 01275, 01280, 01281, 01282, 01283, 04971, 05123, 05348, 13121, 13124, 13488, 28709, 61952
00500	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00367, 00420, 00423, 00424, 00425, 00437, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00869, 00870, 00871, 00875, 00880, 00891, 00897, 00901, 00902, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00924, 00930, 00933, 00935, 00937, 00939, 01010, 01011, 01012, 01013, 01014, 01015, 01016, 01017, 01018, 01019, 01025, 01026, 01027, 01051, 01088, 01089, 01097, 01098, 01112, 01114, 01115, 01122, 01123, 01124, 01126, 01130, 01131, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01250, 01251, 01252, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01364, 01388, 01399, 04909, 04970, 04971, 05026, 05035, 05123, 05348, 08612, 09030, 12708, 13121, 13124, 13488, 28709, 61952, 62211, 62224, 62235, 62245, 62251
00720	00037, 00420, 00864, 01200, 01208, 01256, 13488, 61952
00737	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00437, 00500, 00833, 00836, 00838, 00850, 00869, 00870, 00871, 00875, 00880, 00905, 01025, 01026, 01027, 01097, 01200, 01208, 01252, 01253, 04971, 05123, 13121, 13124, 13488, 28709, 61952
00775	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00437, 00500, 00833, 00836, 00838, 00850, 00870, 00871, 00875, 00880, 00905, 01025, 01026, 01027, 01097, 01112, 01122, 01200, 01208, 01252, 01257, 05123, 13121, 13124, 13488, 28709, 61952
00813	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00819, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 01025, 01026, 01027, 01041, 01042, 01043, 01200, 01208, 01252, 04909, 04971, 05123, 13488, 28709, 61952

From CCSID	To CCSID
00819	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00425, 00437, 00500, 00813, 00833, 00836, 00838, 00850, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00924, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01041, 01042, 01043, 01051, 01088, 01089, 01097, 01098, 01112, 01114, 01122, 01126, 01130, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01250, 01251, 01252, 01253, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01364, 01388, 01399, 04971, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952, 62211, 62235, 62245, 62251
00833	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00367, 00437, 00500, 00737, 00775, 00819, 00836, 00850, 00871, 00891, 00930, 00933, 00935, 00937, 00939, 00944, 00949, 01027, 01040, 01088, 01112, 01122, 01126, 01200, 01208, 01252, 01363, 01364, 01388, 01399, 05026, 05035, 05123, 09030, 13121, 13124, 13488, 28709, 61952
00834	00926, 00933, 00951, 00971, 01200, 01208, 01362, 01364, 04930, 13488, 61952
00835	00927, 00937, 00947, 01200, 01208, 13488, 61952
00836	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00437, 00500, 00737, 00775, 00819, 00833, 00850, 00871, 00903, 00930, 00933, 00935, 00937, 00939, 00946, 01027, 01042, 01112, 01114, 01115, 01122, 01200, 01208, 01252, 01364, 01381, 01386, 01388, 01399, 05026, 05035, 05123, 09030, 13121, 13124, 13488, 28709, 61952
00837	00928, 00935, 01200, 01208, 01380, 01382, 01385, 01388, 13488, 61952
00838	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00937, 01025, 01026, 01027, 01041, 01042, 01043, 01112, 01122, 01160, 01200, 01208, 01252, 04970, 05123, 09030, 13488, 28709, 61952
00850	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00924, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01041, 01051, 01089, 01097, 01098, 01112, 01122, 01130, 01132, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01200, 01208, 01250, 01251, 01252, 01254, 01255, 01256, 01257, 01275, 01280, 01281, 01282, 01283, 01364, 01388, 01399, 04971, 05026, 05035, 05123, 05348, 08612, 09030, 09056, 13121, 13124, 13488, 28709, 61952, 62211, 62235, 62245
00851	01200, 01208, 13488, 61952
00852	00037, 00273, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00869, 00870, 00874, 00875, 00897, 00903, 00912, 01153, 01200, 01208, 01252, 04948, 13488, 61952
00855	00037, 00437, 00500, 00819, 00850, 00915, 01025, 01200, 01208, 01252, 13488, 61952
00857	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00863, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00905, 00912, 00916, 01025, 01026, 01042, 01200, 01208, 01252, 04953, 13488, 28709, 61952
00858	00901, 00902, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01399, 04909, 04971, 05123, 05348
00860	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00861, 00863, 00865, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00937, 01025, 01027, 01041, 01042, 01043, 01200, 01208, 01252, 05123, 13488, 28709, 61952
00861	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00860, 00863, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 01025, 01026, 01027, 01041, 01042, 01043, 01149, 01200, 01208, 01252, 05123, 13488, 28709, 61952

From CCSID	To CCSID
00862	00037, 00424, 00500, 00819, 00850, 00916, 01200, 01208, 01252, 01255, 13488, 61952, 62211, 62235, 62245
00863	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00857, 00860, 00861, 00865, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00937, 01025, 01026, 01027, 01041, 01042, 01043, 01200, 01208, 01252, 05123, 13488, 28709, 61952
00864	00037, 00420, 00425, 00500, 00720, 00819, 00850, 01200, 01208, 01252, 08612, 09056, 13488, 61952, 62251
00865	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00860, 00863, 00871, 00937, 01200, 01208, 01252, 13488, 28709, 61952
00866	00037, 00437, 00500, 00819, 00850, 01025, 01200, 01208, 01252, 13488, 61952
00868	00918, 01200, 01208, 13488, 61952
00869	00037, 00423, 00500, 00737, 00813, 00819, 00838, 00850, 00852, 00870, 00874, 00875, 00880, 00897, 00903, 00912, 01025, 01027, 01041, 01042, 01043, 01200, 01208, 01252, 04971, 05123, 13488, 28709, 61952
00870	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00838, 00850, 00852, 00857, 00861, 00863, 00869, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00915, 00916, 00920, 00937, 01025, 01026, 01112, 01122, 01147, 01153, 01200, 01208, 01250, 01252, 01282, 04948, 04951, 09030, 13488, 28709, 61952
00871	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00870, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 05348, 09030, 13121, 13124, 13488, 28709, 61952
00874	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00813, 00819, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00875, 00880, 00897, 00903, 00912, 00916, 00920, 01025, 01026, 01027, 01041, 01042, 01043, 01200, 01208, 05123, 13488, 28709, 61952
00875	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00880, 00897, 00903, 00912, 00916, 00920, 00937, 01025, 01026, 01027, 01041, 01043, 01112, 01122, 01200, 01208, 01252, 01253, 01280, 04965, 04971, 05123, 09030, 13488, 28709, 61952
00878	01200, 01208, 13488, 61952
00880	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00838, 00850, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00897, 00903, 00912, 00915, 00916, 00920, 00937, 01025, 01026, 01027, 01041, 01042, 01043, 01112, 01122, 01200, 01208, 01251, 01252, 01283, 04948, 04951, 05123, 09030, 13488, 28709, 61952
00891	00500, 00833, 01200, 01208, 13121, 13488, 61952
00897	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00880, 00903, 00912, 00916, 00920, 00930, 00939, 01025, 01026, 01027, 01042, 01043, 01200, 01208, 01252, 01399, 05026, 05035, 05123, 13488, 28709, 61952
00901	00037, 00500, 00858, 01140, 01148, 01156, 01200, 01208, 05348, 13488, 61952
00902	00037, 00500, 00858, 01140, 01148, 01157, 01200, 01208, 05348, 13488, 61952
00903	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00813, 00819, 00836, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00912, 00916, 00920, 01025, 01026, 01027, 01041, 01042, 01043, 05123, 13124, 28709
00904	00037, 28709

From CCSID	To CCSID
00905	00037, 00256, 00500, 00737, 00775, 00819, 00850, 00857, 00920, 00937, 01026, 01112, 01122, 01200, 01208, 01252, 01254, 04953, 09030, 13488, 28709, 61952
00912	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00852, 00857, 00860, 00861, 00863, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00916, 00920, 01025, 01026, 01027, 01041, 01042, 01043, 01153, 01200, 01208, 01252, 05123, 13488, 28709, 61952
00914	00037, 00437, 00500, 00819, 00850, 01200, 01208, 01252, 01257, 13488, 61952
00915	00037, 00437, 00500, 00819, 00850, 00855, 00870, 00880, 01025, 01200, 01208, 01252, 13488, 61952
00916	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00424, 00437, 00500, 00813, 00819, 00838, 00850, 00857, 00860, 00861, 00862, 00863, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00920, 01025, 01026, 01027, 01041, 01042, 01043, 01200, 01208, 01252, 01255, 05123, 13488, 28709, 61952, 62211, 62235, 62245
00918	00037, 00437, 00500, 00819, 00850, 00868, 01200, 01208, 01252, 13488, 61952
00920	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00813, 00819, 00838, 00850, 00860, 00861, 00863, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00905, 00912, 00916, 01025, 01026, 01155, 01200, 01208, 01252, 13488, 28709, 61952
00921	00037, 00437, 00500, 00819, 00850, 01112, 01200, 01208, 01252, 01257, 13488, 61952
00922	00037, 00437, 00500, 00819, 00850, 01122, 01200, 01208, 01252, 01257, 13488, 61952
00923	00924, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 04971, 13488, 61952
00924	00037, 00500, 00819, 00850, 00923, 01200, 01208, 13488, 61952
00926	00834, 01200, 01208, 13488, 61952
00927	00835, 01200, 01208, 13488, 61952
00928	00837, 01200, 01208, 13488, 61952
00930	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00500, 00819, 00833, 00836, 00850, 00871, 00895, 00897, 00932, 00933, 00935, 00937, 00939, 00942, 00943, 00954, 00956, 00957, 00958, 00959, 01027, 01041, 01200, 01208, 01252, 01364, 01388, 01399, 04396, 05026, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 13121, 13124, 13488, 16684, 28709, 33722, 61952
00932	00290, 00930, 00939, 01027, 01200, 01208, 01399, 05026, 05035, 13488, 61952
00933	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00819, 00833, 00834, 00836, 00850, 00871, 00930, 00934, 00935, 00937, 00939, 00944, 00949, 00970, 01027, 01040, 01200, 01208, 01252, 01363, 01364, 01388, 01399, 05026, 05035, 05123, 13121, 13124, 13488, 17354, 25546, 28709, 61952
00934	00833, 00933, 01200, 01208, 01364, 13488, 61952
00935	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00819, 00833, 00836, 00837, 00850, 00871, 00930, 00933, 00936, 00937, 00939, 00946, 01027, 01042, 01200, 01208, 01252, 01364, 01381, 01383, 01386, 01388, 01399, 05026, 05035, 05123, 13121, 13124, 13488, 28709, 61952
00936	00836, 00935, 01388, 13124
00937	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00437, 00500, 00819, 00833, 00835, 00836, 00838, 00850, 00860, 00863, 00865, 00870, 00871, 00875, 00880, 00905, 00930, 00933, 00935, 00938, 00939, 00948, 00950, 00964, 00965, 01025, 01026, 01027, 01043, 01114, 01140, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 13121, 13124, 13488, 28709, 61952
00938	00937, 01200, 01208, 13488, 28709, 61952

From CCSID	To CCSID
00939	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00500, 00819, 00833, 00836, 00850, 00871, 00895, 00897, 00930, 00932, 00933, 00935, 00937, 00942, 00943, 00954, 00956, 00957, 00958, 00959, 01027, 01041, 01200, 01208, 01252, 01364, 01388, 01399, 04396, 05026, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 13121, 13124, 13488, 16684, 28709, 33722, 61952
00941	00300, 01200, 01208, 04396, 13488, 16684, 61952
00942	00290, 00930, 00939, 01027, 01041, 01200, 01208, 01399, 05026, 05035, 05123, 13488, 61952
00943	00290, 00930, 00939, 01027, 01200, 01208, 01399, 05026, 05035, 05123, 13488, 61952
00944	00833, 00933, 01040, 01200, 01208, 01364, 13121, 13488, 61952
00946	00836, 00935, 01042, 01200, 01208, 01388, 13124, 13488, 61952
00947	00835, 01200, 01208, 13488, 61952
00948	00037, 00937, 01043, 01200, 01208, 13488, 28709, 61952
00949	00833, 00933, 01200, 01208, 01364, 13121, 13488, 61952
00950	00937, 01200, 01208, 13488, 28709, 61952
00951	00834, 01200, 01208, 04930, 13488, 61952
00954	00290, 00930, 00939, 01027, 01200, 01208, 01399, 05026, 05035, 13488, 61952
00956	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
00957	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
00958	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
00959	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
00964	00937, 01200, 01208, 13488, 61952
00965	00937, 01200, 01208, 13488, 61952
00970	00933, 01200, 01208, 01364, 13488, 61952
00971	00834, 04930
01008	00420, 08612
01009	01200, 01208, 13488, 61952
01010	00500, 01200, 01208, 13488, 61952
01011	00500, 01200, 01208, 13488, 61952
01012	00500, 01200, 01208, 13488, 61952
01013	00500, 01200, 01208, 13488, 61952
01014	00500, 01200, 01208, 13488, 61952
01015	00500, 01200, 01208, 13488, 61952
01016	00500, 01200, 01208, 13488, 61952
01017	00500, 01200, 01208, 13488, 61952
01018	00500, 01200, 01208, 13488, 61952
01019	00500, 01200, 01208, 13488, 61952
01025	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00838, 00850, 00855, 00857, 00860, 00861, 00863, 00866, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00915, 00916, 00920, 00937, 01026, 01027, 01042, 01112, 01122, 01131, 01154, 01200, 01208, 01251, 01252, 01283, 04948, 04951, 05123, 09030, 13488, 28709, 61952

From CCSID	To CCSID
01026	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00838, 00850, 00857, 00861, 00863, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00905, 00912, 00916, 00920, 00937, 01025, 01042, 01112, 01122, 01155, 01200, 01208, 01252, 01254, 01281, 04953, 09030, 13488, 28709, 61952
01027	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00860, 00861, 00863, 00869, 00871, 00874, 00875, 00880, 00895, 00897, 00903, 00912, 00916, 00930, 00933, 00935, 00937, 00939, 00942, 00943, 00954, 01025, 01041, 01042, 01112, 01122, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 09030, 13121, 13124, 13488, 28709, 61952
01040	00833, 00933, 00944, 01364, 13121
01041	00037, 00290, 00423, 00813, 00819, 00838, 00850, 00860, 00861, 00863, 00869, 00874, 00875, 00880, 00903, 00912, 00916, 00930, 00939, 00942, 01027, 01042, 01399, 05026, 05035, 05123
01042	00423, 00437, 00813, 00819, 00836, 00838, 00857, 00860, 00861, 00863, 00869, 00874, 00880, 00897, 00903, 00912, 00916, 00935, 00946, 01025, 01026, 01027, 01041, 01043, 01388, 05123, 13124
01043	00037, 00423, 00813, 00819, 00838, 00860, 00861, 00863, 00869, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00937, 00948, 01042, 28709
01046	00420, 00425, 01200, 01208, 08612, 13488, 61952, 62251
01051	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00871, 01200, 01208, 01252, 05348, 13488, 61952
01088	00037, 00500, 00819, 00833, 13121, 28709
01089	00037, 00420, 00425, 00500, 00819, 00850, 01200, 01208, 13488, 61952, 62251
01097	00037, 00437, 00500, 00737, 00775, 00819, 00850, 01098, 01112, 01122, 01200, 01208, 01252, 01256, 09030, 13488, 61952
01098	00037, 00437, 00500, 00819, 00850, 01097, 01200, 01208, 01252, 13488, 61952
01112	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00500, 00775, 00819, 00833, 00836, 00838, 00850, 00870, 00871, 00875, 00880, 00905, 00921, 01025, 01026, 01027, 01097, 01122, 01156, 01200, 01208, 01252, 01257, 05123, 09030, 13121, 13124, 13488, 28709, 61952
01114	00037, 00500, 00819, 00836, 00937, 13124, 28709
01115	00037, 00500, 00836, 13124
01122	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00500, 00775, 00819, 00833, 00836, 00838, 00850, 00870, 00871, 00875, 00880, 00905, 00922, 01025, 01026, 01027, 01097, 01112, 01157, 01200, 01208, 01252, 01257, 05123, 09030, 13121, 13124, 13488, 28709, 61952
01123	00037, 00500, 01124, 01125, 01158, 01200, 01208, 01251, 13488, 61952
01124	00037, 00500, 01123, 01200, 01208, 13488, 61952
01125	01123, 01200, 01208, 13488, 61952
01126	00037, 00500, 00819, 00833, 01252, 13121
01129	01200, 01208, 13488, 61952
01130	00037, 00500, 00819, 00850, 01164, 01200, 01208, 01252, 01258, 13488, 61952
01131	00037, 00500, 01025, 01200, 01208, 13488, 61952
01132	00037, 00500, 00819, 00850, 01133, 01200, 01208, 01252, 13488, 61952
01133	01132
01137	00037, 00500, 00819, 01200, 01208, 01252, 13488, 61952

From CCSID	To CCSID
01140	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00425, 00437, 00500, 00819, 00850, 00858, 00871, 00901, 00902, 00923, 00937, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952, 62251
01141	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00871, 00923, 01140, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01142	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00871, 00923, 01140, 01141, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01143	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00871, 00923, 01140, 01141, 01142, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01144	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00871, 00923, 01140, 01141, 01142, 01143, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01145	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00871, 00923, 01140, 01141, 01142, 01143, 01144, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01146	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00871, 00923, 01140, 01141, 01142, 01143, 01144, 01145, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01147	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00870, 00871, 00923, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01148	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00425, 00437, 00500, 00819, 00850, 00858, 00871, 00901, 00902, 00923, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952, 62251
01149	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00437, 00500, 00819, 00850, 00858, 00861, 00871, 00923, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01252, 01399, 04909, 04971, 05123, 05348, 13488, 61952
01153	00037, 00500, 00819, 00850, 00852, 00870, 00912, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01250, 04971, 13488, 61952
01154	00037, 00500, 00819, 01025, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01251, 04971, 05123, 13488, 61952
01155	00037, 00500, 00819, 00920, 01026, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01254, 04971, 13488, 61952
01156	00037, 00500, 00819, 00901, 01112, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01257, 04971, 05123, 13488, 61952
01157	00037, 00500, 00819, 00902, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01257, 04971, 05123, 13488, 61952
01158	00037, 00500, 00819, 01123, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01251, 13488, 61952

From CCSID	To CCSID
01160	00037, 00500, 00819, 00838, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 05123, 13488, 61952
01164	00037, 00500, 00819, 01130, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01258, 13488, 61952
01200	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00301, 00367, 00420, 00423, 00424, 00425, 00437, 00500, 00720, 00737, 00775, 00813, 00819, 00833, 00834, 00835, 00836, 00837, 00838, 00850, 00851, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00868, 00869, 00870, 00871, 00874, 00875, 00878, 00880, 00891, 00897, 00901, 00902, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00923, 00924, 00926, 00927, 00928, 00930, 00932, 00933, 00934, 00935, 00937, 00938, 00939, 00941, 00942, 00943, 00944, 00946, 00947, 00948, 00949, 00950, 00951, 00954, 00956, 00957, 00958, 00959, 00964, 00965, 00970, 01009, 01010, 01011, 01012, 01013, 01014, 01015, 01016, 01017, 01018, 01019, 01025, 01026, 01027, 01046, 01051, 01089, 01097, 01098, 01112, 01122, 01123, 01124, 01125, 01129, 01130, 01131, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01208, 01250, 01251, 01252, 01253, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01362, 01363, 01364, 01380, 01381, 01383, 01385, 01386, 01388, 01392, 01399, 04396, 04909, 04930, 04933, 04948, 04951, 04952, 04960, 04971, 05026, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 05348, 08612, 09030, 09056, 09066, 12708, 13121, 13124, 13488, 16684, 17354, 28709, 33722, 61952, 62211, 62224, 62235, 62245, 62248, 62251
01208	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00301, 00367, 00420, 00423, 00424, 00425, 00437, 00500, 00720, 00737, 00775, 00813, 00819, 00833, 00834, 00835, 00836, 00837, 00838, 00850, 00851, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00868, 00869, 00870, 00871, 00874, 00875, 00878, 00880, 00891, 00897, 00901, 00902, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00923, 00924, 00926, 00927, 00928, 00930, 00932, 00933, 00934, 00935, 00937, 00938, 00939, 00941, 00942, 00943, 00944, 00946, 00947, 00948, 00949, 00950, 00951, 00954, 00956, 00957, 00958, 00959, 00964, 00965, 00970, 01009, 01010, 01011, 01012, 01013, 01014, 01015, 01016, 01017, 01018, 01019, 01025, 01026, 01027, 01046, 01051, 01089, 01097, 01098, 01112, 01122, 01123, 01124, 01125, 01129, 01130, 01131, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01250, 01251, 01252, 01253, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01362, 01363, 01364, 01380, 01381, 01383, 01385, 01386, 01388, 01392, 01399, 04396, 04909, 04930, 04933, 04948, 04951, 04952, 04960, 04971, 05026, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 05348, 08612, 09030, 09056, 09066, 12708, 13121, 13124, 13488, 16684, 17354, 28709, 33722, 62211, 62224, 62235, 62245, 62248, 62251
01250	00037, 00273, 00500, 00819, 00850, 00870, 01153, 01200, 01208, 01252, 01282, 13488, 61952
01251	00037, 00500, 00819, 00850, 00880, 01025, 01123, 01154, 01158, 01200, 01208, 01252, 01283, 13488, 61952
01252	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00425, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00869, 00870, 00871, 00875, 00880, 00897, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00930, 00933, 00935, 00937, 00939, 01025, 01026, 01027, 01051, 01097, 01098, 01112, 01122, 01126, 01130, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01250, 01251, 01254, 01255, 01256, 01257, 01275, 01280, 01281, 01282, 01283, 01364, 01388, 01399, 04971, 05026, 05035, 05123, 05348, 13121, 13124, 13488, 28709, 61952, 62211, 62235, 62245, 62251
01253	00037, 00423, 00737, 00819, 00875, 01200, 01208, 01280, 04971, 13488, 61952
01254	00037, 00500, 00819, 00850, 00905, 01026, 01155, 01200, 01208, 01252, 13488, 61952
01255	00037, 00424, 00500, 00819, 00850, 00862, 00916, 01200, 01208, 01252, 01281, 13488, 61952, 62211, 62235, 62245
01256	00037, 00420, 00425, 00500, 00720, 00819, 00850, 01097, 01200, 01208, 01252, 08612, 12708, 13488, 61952, 62224, 62251

From CCSID	To CCSID
01257	00037, 00437, 00500, 00775, 00819, 00850, 00914, 00921, 00922, 01112, 01122, 01156, 01157, 01200, 01208, 01252, 13488, 61952
01258	00037, 00500, 00819, 01130, 01164, 01200, 01208, 13488, 61952
01275	00037, 00437, 00500, 00819, 00850, 01200, 01208, 01252, 05348, 13488, 61952
01280	00037, 00437, 00500, 00819, 00850, 00875, 01200, 01208, 01252, 01253, 13488, 61952
01281	00037, 00437, 00500, 00819, 00850, 01026, 01200, 01208, 01252, 01255, 13488, 61952
01282	00037, 00437, 00500, 00819, 00850, 00870, 01200, 01208, 01250, 01252, 13488, 61952
01283	00037, 00437, 00500, 00819, 00850, 00880, 01025, 01200, 01208, 01251, 01252, 13488, 61952
01362	00834, 01200, 01208, 04930, 13488, 61952
01363	00833, 00933, 01200, 01208, 01364, 13488, 61952
01364	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00819, 00833, 00834, 00836, 00850, 00871, 00930, 00933, 00934, 00935, 00937, 00939, 00944, 00949, 00970, 01027, 01040, 01200, 01208, 01252, 01363, 01388, 01399, 05026, 05035, 05123, 13121, 13124, 13488, 28709, 61952
01380	00837, 01200, 01208, 04933, 13488, 61952
01381	00836, 00935, 01200, 01208, 01388, 13488, 61952
01382	00837, 04933
01383	00935, 01200, 01208, 01388, 13488, 61952
01385	00837, 01200, 01208, 04933, 13488, 61952
01386	00836, 00935, 01200, 01208, 01388, 13124, 13488, 61952
01388	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00819, 00833, 00836, 00837, 00850, 00871, 00930, 00933, 00935, 00936, 00937, 00939, 00946, 01027, 01042, 01200, 01208, 01252, 01364, 01381, 01383, 01386, 01399, 05026, 05035, 05123, 13121, 13124, 13488, 28709, 61952
01392	01200, 01208, 13488
01399	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00819, 00833, 00836, 00850, 00858, 00871, 00895, 00897, 00930, 00932, 00933, 00935, 00937, 00939, 00942, 00943, 00954, 01027, 01041, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01364, 01388, 05026, 05035, 05050, 05123, 13121, 13124, 13488, 16684, 28709, 33722, 61952
04396	00300, 00301, 00930, 00939, 00941, 01200, 01208, 05026, 05035, 13488, 16684, 61952
04909	00037, 00500, 00813, 00858, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 04971, 13488, 61952
04930	00834, 00951, 00971, 01200, 01208, 01362, 13488, 61952
04933	01200, 01208, 01380, 01382, 01385, 13488, 61952
04948	00852, 00870, 00880, 01025, 01200, 01208, 13488, 61952
04951	00870, 00880, 01025, 01200, 01208, 13488, 61952
04952	00424, 01200, 01208, 13488, 61952
04953	00857, 00905, 01026
04960	00420, 01200, 01208, 08612, 13488, 61952
04965	00423, 00875
04970	00037, 00500, 00838
04971	00037, 00423, 00437, 00500, 00737, 00813, 00819, 00850, 00858, 00869, 00875, 00923, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01200, 01208, 01252, 01253, 04909, 05348, 13488, 61952

From CCSID	To CCSID
05026	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00500, 00819, 00833, 00836, 00850, 00871, 00895, 00897, 00930, 00932, 00933, 00935, 00937, 00939, 00942, 00943, 00954, 00956, 00957, 00958, 00959, 01027, 01041, 01200, 01208, 01252, 01364, 01388, 01399, 04396, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 13121, 13124, 13488, 16684, 28709, 33722, 61952
05035	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00500, 00819, 00833, 00836, 00850, 00871, 00895, 00897, 00930, 00932, 00933, 00935, 00937, 00939, 00942, 00943, 00954, 00956, 00957, 00958, 00959, 01027, 01041, 01200, 01208, 01252, 01364, 01388, 01399, 04396, 05026, 05050, 05052, 05053, 05054, 05055, 05123, 13121, 13124, 13488, 16684, 28709, 33722, 61952
05050	00930, 00939, 01200, 01208, 01399, 05026, 05035, 13488, 61952
05052	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
05053	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
05054	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
05055	00930, 00939, 01200, 01208, 05026, 05035, 13488, 61952
05123	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00423, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00858, 00860, 00861, 00863, 00869, 00871, 00874, 00875, 00880, 00897, 00903, 00912, 00916, 00930, 00933, 00935, 00937, 00939, 00942, 00943, 01025, 01027, 01041, 01042, 01112, 01122, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01154, 01156, 01157, 01160, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05348, 09030, 13121, 13124, 13488, 28709, 61952
05210	13124
05348	00037, 00273, 00277, 00278, 00280, 00284, 00285, 00437, 00500, 00819, 00850, 00858, 00871, 00901, 00902, 01051, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01200, 01208, 01252, 01275, 04971, 05123, 13488, 61952
08612	00037, 00256, 00420, 00425, 00500, 00850, 00864, 01008, 01046, 01200, 01208, 01256, 04960, 12708, 13488, 28709, 61952, 62224, 62251
09030	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00500, 00819, 00833, 00836, 00838, 00850, 00870, 00871, 00875, 00880, 00905, 01025, 01026, 01027, 01097, 01112, 01122, 01200, 01208, 05123, 09066, 13121, 13488, 28709, 61952
09056	00420, 00850, 00864, 01200, 01208, 13488, 61952
09066	01200, 01208, 09030, 13488, 61952
12708	00037, 00420, 00425, 00500, 01200, 01208, 01256, 08612, 13488, 61952, 62224, 62251
13121	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00367, 00437, 00500, 00737, 00775, 00819, 00833, 00836, 00850, 00871, 00891, 00930, 00933, 00935, 00937, 00939, 00944, 00949, 01027, 01040, 01088, 01112, 01122, 01126, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 09030, 13124, 13488, 28709, 61952
13124	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00367, 00437, 00500, 00737, 00775, 00819, 00833, 00836, 00850, 00871, 00903, 00930, 00933, 00935, 00937, 00939, 00946, 01027, 01042, 01112, 01114, 01115, 01122, 01200, 01208, 01252, 01364, 01386, 01388, 01399, 05026, 05035, 05123, 05210, 13121, 13488, 28709, 61952

From CCSID	To CCSID
13488	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00301, 00367, 00420, 00423, 00424, 00425, 00437, 00500, 00720, 00737, 00775, 00813, 00819, 00833, 00834, 00835, 00836, 00837, 00838, 00850, 00851, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00868, 00869, 00870, 00871, 00874, 00875, 00878, 00880, 00891, 00897, 00901, 00902, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00923, 00924, 00926, 00927, 00928, 00930, 00932, 00933, 00934, 00935, 00937, 00938, 00939, 00941, 00942, 00943, 00944, 00946, 00947, 00948, 00949, 00950, 00951, 00954, 00956, 00957, 00958, 00959, 00964, 00965, 00970, 01009, 01010, 01011, 01012, 01013, 01014, 01015, 01016, 01017, 01018, 01019, 01025, 01026, 01027, 01046, 01051, 01089, 01097, 01098, 01112, 01122, 01123, 01124, 01125, 01129, 01130, 01131, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01208, 01250, 01251, 01252, 01253, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01362, 01363, 01364, 01380, 01381, 01383, 01385, 01386, 01388, 01392, 01399, 04396, 04909, 04930, 04933, 04948, 04951, 04952, 04960, 04971, 05026, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 05348, 08612, 09030, 09056, 09066, 12708, 13121, 13124, 16684, 17354, 28709, 33722, 61952, 62211, 62224, 62235, 62245, 62248, 62251
16684	00301, 00930, 00939, 00941, 01200, 01208, 01399, 04396, 05026, 05035, 13488, 61952
17354	00933, 01200, 01208, 13488, 61952
25546	00933
28709	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00420, 00423, 00424, 00437, 00500, 00737, 00775, 00813, 00819, 00833, 00836, 00838, 00850, 00857, 00860, 00861, 00863, 00865, 00869, 00870, 00871, 00874, 00875, 00880, 00897, 00903, 00904, 00905, 00912, 00916, 00920, 00930, 00933, 00935, 00937, 00939, 00948, 00950, 01025, 01026, 01027, 01043, 01088, 01112, 01114, 01122, 01200, 01208, 01252, 01364, 01388, 01399, 05026, 05035, 05123, 08612, 09030, 13121, 13124, 13488, 61952
33722	00930, 00939, 01200, 01208, 01399, 05026, 05035, 13488, 61952
57345	00930, 00939, 05026, 05035
57777	01208, 13488
61952	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00300, 00301, 00367, 00420, 00423, 00424, 00425, 00437, 00500, 00720, 00737, 00775, 00813, 00819, 00833, 00834, 00835, 00836, 00837, 00838, 00850, 00851, 00852, 00855, 00857, 00860, 00861, 00862, 00863, 00864, 00865, 00866, 00868, 00869, 00870, 00871, 00874, 00875, 00878, 00880, 00891, 00897, 00901, 00902, 00905, 00912, 00914, 00915, 00916, 00918, 00920, 00921, 00922, 00923, 00924, 00926, 00927, 00928, 00930, 00932, 00933, 00934, 00935, 00937, 00938, 00939, 00941, 00942, 00943, 00944, 00946, 00947, 00948, 00949, 00950, 00951, 00954, 00956, 00957, 00958, 00959, 00964, 00965, 00970, 01009, 01010, 01011, 01012, 01013, 01014, 01015, 01016, 01017, 01018, 01019, 01025, 01026, 01027, 01046, 01051, 01089, 01097, 01098, 01112, 01122, 01123, 01124, 01125, 01129, 01130, 01131, 01132, 01137, 01140, 01141, 01142, 01143, 01144, 01145, 01146, 01147, 01148, 01149, 01153, 01154, 01155, 01156, 01157, 01158, 01160, 01164, 01200, 01250, 01251, 01252, 01253, 01254, 01255, 01256, 01257, 01258, 01275, 01280, 01281, 01282, 01283, 01362, 01363, 01364, 01380, 01381, 01383, 01385, 01386, 01388, 01399, 04396, 04909, 04930, 04933, 04948, 04951, 04952, 04960, 04971, 05026, 05035, 05050, 05052, 05053, 05054, 05055, 05123, 05348, 08612, 09030, 09056, 09066, 12708, 13121, 13124, 13488, 16684, 17354, 28709, 33722, 62211, 62224, 62235, 62245, 62248, 62251
62209	62211, 62235, 62245
62210	00424, 62211, 62235, 62245
62211	00037, 00424, 00500, 00819, 00850, 00862, 00916, 01200, 01208, 01252, 01255, 13488, 61952, 62209, 62210, 62213, 62215, 62222, 62223, 62235, 62245
62213	62211
62215	00424, 62211, 62235, 62245
62218	00420
62221	62235

From CCSID	To CCSID
62222	00424, 62211, 62235, 62245
62223	00424, 62211, 62235, 62245
62224	00037, 00420, 00425, 00500, 01200, 01208, 01256, 08612, 12708, 13488, 61952, 62251
62228	00420, 00425, 62251
62235	00037, 00424, 00500, 00819, 00850, 00862, 00916, 01200, 01208, 01252, 01255, 13488, 61952, 62209, 62210, 62211, 62215, 62221, 62222, 62223, 62245
62238	00424, 62245
62239	00424, 62245
62245	00037, 00424, 00500, 00819, 00850, 00862, 00916, 01200, 01208, 01252, 01255, 13488, 61952, 62209, 62210, 62211, 62215, 62222, 62223, 62235, 62238, 62239
62248	00420, 01200, 01208, 13488, 61952
62251	00037, 00420, 00500, 00819, 00864, 01046, 01089, 01140, 01148, 01200, 01208, 01252, 01256, 08612, 12708, 13488, 61952, 62224, 62228

Associated CCSIDs

This table shows the associated CCSIDs for a given CCSID value and encoding scheme.

Although not represented in the table, if you request an encoding scheme that is the same as the input CCSID, you will get back the input CCSID. For example, if you request the associated CCSID of CCSID 00037 with an encoding scheme of 01100, you will get back 00037.

Input CCSID	1100	1200	1301	2100	2200	2300	4100	4105	4403	5100	5404
00037		00835	00937	00437	00947	00950	00819	01252		00367	
00256				00437			00819	01252		00367	
00273				00850			00819	01252			
00277				00865			00819	01252			
00278				00865			00819	01252			
00280				00850			00819	01252			
00284				00850			00819	01252			
00285				00850			00819	01252			
00290		04396	05026	01041	00301	00942					
00297				00850			00819	01252			
00300	00290		00930	01041	00301	00942					
00301	00290	04396	05026	01041		00942					
00367	00500			00850			00819	01252			
00420				00864			01089	01256			
00423				00869			00813	01253			
00424				00862			00916	01255			
00437	00037						00819	01252			
00500				00850			00819	01252		00367	
00720	00420						01089	01256			
00737	00875						00813	01253			
00775	01112						00921	01257			

Input CCSID	1100	1200	1301	2100	2200	2300	4100	4105	4403	5100	5404
00813	00875							01253			
00819	00500			00850				01252		00367	
00833		00834	00933	01088	00951	00949					
00834	00833		00933	01088	00951	00949					
00835	28709		00937	01043	00927	00950					
00836		00837	00935	01115	01380	01381					
00837	00836		00935	01115	01380	01381					
00838				00874							
00850	00500						00819	01252			
00851	00875						00813	01253			
00852	00870						00912	01250			
00855	01025							01251			
00857	01026						00920	01254			
00860	00037						00819	01252			
00861	00871						00819	01252			
00862	00424						00916	01255			
00863	00500						00819	01252			
00864	00420						01089	01256			
00865	00277						00819	01252			
00866	01025			00866				01251			
00869	00875						00813	01253			
00870				00852			00912	01250			
00871				00861			00819	01252			
00874	00838										
00875				00869			00813	01253			
00880				00855							
00891	00833	00834	00933		00926	00934					
00897	00290	04396	05026		00301	00932					
00903	00836	00837	00935		00928	00936					
00904	28709	00835	00937		00927	00938					
00905				00857			00920				
00912	00870			00852				01250			
00915	01025			00855				01251			
00916	00424			00862				01255			
00920	01026			00857				01254			
00921	01112			01125				01257			
00922	01122			01125				01251			
00923	00924			00850				01252		00367	
00924				00850			00923	01252		00367	
00926	00833	00834	00933	01040		00944					

Input CCSID	1100	1200	1301	2100	2200	2300	4100	4105	4403	5100	5404
00927	28709	00835	00937	01043		00950					
00928	00836	00837	00935	01042		00946					
00930	00290	00300		01041	00301	00942					05052
00932	00290	04396	05026	00897	00301						
00933	00833	00834		01088	00951	00949			00970		17354
00934	00833	00834	00933	00891	00926						
00935	00836	00837		01115	01380	01381					
00936	00836	00837	00935	00903	00928						
00937	28709	00835		01043	00927	00950			00964		00965
00938	28709	00835	00937	00904	00927						
00939	01027	00300		01041	00301	00942					05052
00942	00290	04396	05026	01041	00301						
00943	00290	04396	05026	00897	00301						
00944	00833	00834	00933	01040	00926						
00946	00836	00837	00935	01042	00928						
00947		00835	00937								
00948	28709	00835	00937	01043	00927						
00949	00833	00834	00933	01088	00951				00970		17354
00950	28709	00835	00937	01114	00947						
00951	00833	00834	00933	01088		00949			00970		17354
00956	00290	00300	05026								
00957	00290	00300	05026								
00958	00290	00300	05026								
00959	00290	00300	05026								
00964	00037	00835	00937								
00965	00037	00835	00937								
00970	13121	04930	01364								
01008	00420										
01009	00500										
01010	00500										
01011	00500										
01012	00500										
01013	00500										
01014	00500										
01015	00500										
01016	00500										
01017	00500										
01018	00500										
01019	00500										
01025				00855				01251			

Input CCSID	1100	1200	1301	2100	2200	2300	4100	4105	4403	5100	5404
01026				00857			00920	01254			
01027		04396	05035	01041	00301	00942					
01040	00833	00834	00933		00926	00944					
01041	00290	04396	05026		00301	00942					
01042	00836	00837	00935		00928	00946					
01043	28709	00835	00937		00927	00950					
01046	00420										
01051	00037			00850							
01088	00833	00834	00933		00951	00949			00970		17354
01089	00420							01256			
01097				01098							
01098	01097										
01112								01257			
01114	28709										
01115	00836	00837	00935		01380	01381					
01122				01125			01124	01257			
01123				01125			01124	01251			
01126	13121	04930	01364		01362	01363			00970		17354
01129	01130							01258			
01130							01129	01258			
01132							01133				
01133	01132										
01140				00437			00923	01252		00367	
01141				00850			00923	01252		00367	
01142				00865			00923	01252		00367	
01143				00865			00923	01252		00367	
01144				00850			00923	01252		00367	
01145				00850			00923	01252		00367	
01146				00850			00923	01252		00367	
01147				00850			00923	01252		00367	
01148				00850			00923	01252		00367	
01149				00861			00923	01252		00367	
01153				00852			00912	01250			
01154				00855				01251			
01155				00857			00920	01254			
01156								01257			
01157				01125			01124	01257			
01158				01125			01124	01251			
01160				00874							
01164							01129	01258			

Input CCSID	1100	1200	1301	2100	2200	2300	4100	4105	4403	5100	5404
01250	00870			00852							
01251	01025			00855							
01252	00500			00850			00819			00367	
01253	00875			00869							
01254	01026			00857							
01255	00424			00862							
01256	00420			00864							
01257	01112						00921				
01258	01130			01258			01129				
01275	00037			00850							
01280	00875			00869							
01281	01026			00857							
01282	00870			00852							
01283	01025			00855							
01362	13121	04930	01364	01126		01363			00970		17354
01363	13121	04930	01364	01126	01362				00970		17354
01364	13121	04930		01126	01362	01363			00970		17354
01380	00836	00837	00935	01115		01381					
01381	00836	00837	00935	01115	01380						
01386	13124	04933	01388								
01388	13124	04933		01114	01385	01386					
01399	05123	16684		01041	00301	00942			05050		05052
04396	00290		05026	01041	00301	00942					
04930	13121		01364	01126	01362	01363					
04933	13124		01388	01114	01385	01386					
04948	00870										
04951	01025										
04952	00424										
04953	01026										
04960	00420										
04965	00875										
05026	00290	04396		01041	00301	00942			05050		05052
05035	01027	04396		01041	00301	00942					05052
05050	00290	00300	05026	01041	00301	00942					
05052	00290	00300	05026	01041	00301	00942					
05053	00290	00300	05026	01041	00301	00942					
00000											
05054	00290	00300	05026	01041	00301	00942					
00000											
05055	00290	00300	05026	01041	00301	00942					

Input CCSID	1100	1200	1301	2100	2200	2300	4100	4105	4403	5100	5404
00000											
05123		16684	01399	01041	00301	00942					
00000											
09030				09066							
09066	09030										
13121		04930	01364	01126	01362	01363					
13124		04933	01388	01114	01385	01386					
00000											
16684	05123		01399	01041	00301	00942					
00000											
17354	00833	00834	00933								
25546	00833	00834	00933	01088	00951	00949					
28709		00835	00937	01043	00927	00948					
33722	00290	04366	05026								
57345	00290	00300	05026								

Encoding schemes for the CCSIDs

This table shows the encoding scheme values (from Character Data Representation Architecture) that are used for the CCSIDs.

Encoding scheme identifier (ESID) hexadecimal	Interpretation
1100	EBCDIC, single-byte, No code extension is allowed. Number of States = 1.
1200	EBCDIC, double-byte, No code extension is allowed. Number of States = 1.
1300	EBCDIC, mixed-byte, No code extension is allowed Number of States = 1.
1301	EBCDIC, mixed single-byte and double-byte, using shift-in (SI) and shift-out (SO) code extension method Number of States = 2.
2100	IBM-PC Data, single-byte, No code extension is allowed Number of States = 1.
2200	IBM-PC Data, double-byte, No code extension is allowed Number of States = 1.
2300	IBM-PC Data, mixed single-byte and double-byte, with implicit code extension Number of States = 2.
3100	IBM-PC Display, single-byte, No code extension is allowed Number of States = 1.
3200	IBM-PC Display, double-byte, No code extension is allowed Number of States = 1.
3300	IBM-PC Display, mixed single-byte and double-byte, with implicit code extension Number of States = 2.
4100	ISO 8, single-byte, No code extension is allowed Number of States = 1.
4105	ISO 8 (ASCII code), single-byte, Graphics in C1 Note that graphic characters may be present in the area normally reserved for the C1 control codes. (ie X'80' to X'9F') Number of States = 1.
4403	IBM EUC Number of States = 2 to 4
5100	ISO 7 (ASCII code), single-byte, No code extension is allowed Number of States = 1.

Encoding scheme identifier (ESID) hexadecimal	Interpretation
5404	ISO 2022 TCP/IP Number of States = 2 to 4.
7200	UCS-2, No code extension is allowed Number of States = 1.
7808	UTF-8, No code extension is allowed Number of States = 3.

Language identifiers and associated default CCSIDs

This table shows the language identifiers and the job default CCSID (DFTCCSID) values associated with those identifiers.

If the QTQ_DEFAULT_CCSID system level environment variable is not defined, then the defaults act as indicated in the following table.

If the QTQ_DEFAULT_CCSID system level environment variable is defined, then it consists of pairs of valid language ID and single-byte or mixed-byte EBCDIC CCSID values. You can use this environment variable to change to the euro value. If this value is not properly defined, the system issues an informational message and uses the value in the following table.

For example, the following command specifies that you want the default CCSID to be 500 for the ENU and ENP language identifiers:

```
ADDENVVAR ENVVAR(QTQ_DEFAULT_CCSID) VALUE('ENU 00500 ENP 00500') LEVEL(*SYS)
```

You should be careful to ensure that this environment is set to a valid value. If this value is not set correctly, you might not be able to log on to the system. Before you change this value, you might consider having two signed-on sessions. Then, you can test your change on one of the sessions and make changes on the other if needed.

The Job default coded character set identifier topic includes information about the QTQ_DEFAULT_CCSID system-level environment variable and how it is used to find the default value.

Language identifier	Single-byte CCSID	Mixed-byte CCSID	Language
AFR	00037	None	Afrikaans
ARA	00420	None	Arabic
BEL	01025	None	Byelorussian
BGR	01025	None	Bulgarian
CAT	00284	None	Catalan
CHS	00836	00935	Simplified Chinese
CHT	00037	00937	Traditional Chinese
CSY	00870	None	Czech
DAN	00277	None	Danish
DES	00500	None	Swiss German
DEU	00273 (1141 with Euro)	None	German
ELL	00875 (4971 with Euro)	None	Greek
ENA	00037	None	Australian English
ENG	00285	None	United Kingdom English

Language identifier	Single-byte CCSID	Mixed-byte CCSID	Language
ENP	00037	None	Uppercase English
ENU	00037	None	United States English
ESP	00284 (1145 with Euro)	None	Spanish
FAR	01097	None	Farsi
FIN	00278 (1143 with Euro)	None	Finnish
FRA	00297 (1147 with Euro)	None	French
FRB	00500 (1148 with Euro)	None	Belgian French
FRC	00500	None	Canadian French
FRS	00500	None	Swiss French
GAE	00285 (1146 with Euro)	None	Irish Gaelic
HEB	00424	None	Hebrew
HRV	00870	None	Croatian
HUN	00870	None	Hungarian
ISL	00871 (1149 with Euro)	None	Icelandic
ITA	00280 (1144 with Euro)	None	Italian
ITS	00500	None	Swiss Italian
JPN	00290	05026	Japanese Katakana The mixed value for DFTCCSID is 05026 when the job CCSID is 65535. In order for the DFTCCSID to be 05035, the job CCSID must be 05035.
KOR	00833	00933	Korean
MKD	01025	None	Macedonian
NLB	00500 (1148 with Euro)	None	Belgian Dutch
NLD	00037 (1140 with Euro)	None	Dutch
NON	00277	None	Norwegian Nynorsk
NOR	00277	None	Norwegian Bokmal
PLK	00870	None	Polish
PTB	00037	None	Brazilian Portuguese
PTG	00037 (1140 with Euro)	None	Portuguese
ROM	00870	None	Romanian
RUS	01025	None	Russian
SKY	00870	None	Slovakian
SLO	00870	None	Slovenian
SQI	00500	None	Albanian
SRB	01025	None	Serbian Cyrillic
SRL	00870	None	Serbian Latin
SVE	00278	None	Swedish
THA	00838	None	Thai
TRK	01026	None	Turkish

Related concepts

"Recommendations and guidelines for using CCSIDs" on page 126
These recommendations are useful when you write globalized applications.

"Database management" on page 130
Database management support provides default coded character set identifier (CCSID) values for database files on the system. All database files are assigned a CCSID. At file creation time, the CCSID is either explicitly assigned through DDS, SQL, or IDDU, or implicitly assigned the job default CCSID (DFTCCSID).

"LC_COLLATE category" on page 203
The LC_COLLATE category defines character or string collation information. Within LC_COLLATE you can specify a sort sequence to use using the cpysyscol keyword. The *cpysyscol* keyword value is used in place of the LC_COLLATE category definitions.

Related tasks

"Job default coded character set identifier" on page 13
A job attribute, job default CCSID (DFTCCSID), is created for jobs with a CCSID of 65535. The DFTCCSID value is used by a system code when a CCSID other than 65535 is needed.

Locale reference information

A *locale* is an identifier for a set of conventions governing the input, processing, and display of computerized data so that they match the requirements and expectations of a particular user community. This information covers locales on the i5/OS operating system.

System-supplied locales and recommended CCSIDs

The system-supplied locale source definition file members are in the optionally installable library QSYSLOCALE in the QLOCALESRC source file. The source file members are encoded in CCSID 37 and are read only.

The following table shows the locale source file definition members that are shipped with i5/OS, the recommended CCSID for creating each locale, and how the locale is supplied (as source, object, or both).

The source definition file members themselves cannot be changed. They can be copied and then edited, if needed.

Notes:

- Source is changed in V5R4 to support euro values.
- The Japanese 5035 is shipped as an object but uses the same JA_JP source member as the Japanese.

Description	Member	CCSID	How shipped
Albania/Albanian	SQ_AL	500	Both
Algeria/Arabic	AR_DZ	420	Source
Arabic-speaking countries/Arabic	AR_AA	420	Both
Argentina/Spanish	ES_AR	284	Source
Australia/English	EN_AU	37	Source
Austria/German	DE_AT	1141	Source
Austria/German euro	DE_AT_E	1141	Source
Austria/German preeuro	DE_AT_PE	273	Source
Bahrain/Arabic	AR_BH	420	Source
Belgium/Dutch	NL_BE	1148	Both

Description	Member	CCSID	How shipped
Belgium/Dutch euro	NL_BE_E	1148	Both
Belgium/Dutch preeuro	NL_BE_PE	500	Both
Belgium/English	EN_BE	1148	Both
Belgium/English euro	EN_BE_E	1148	Source
Belgium/English preeuro	EN_BE_PE	37	Source
Belgium/French	FR_BE	1148	Both
Belgium/French euro	FR_BE_E	1148	Both
Belgium/French preeuro	FR_BE_PE	500	Both
Bolivia/Spanish	ES_BO	284	Source
Brazil/Portuguese	PT_BR	37	Both
Bulgaria/Bulgarian	BG_BG	1025	Both
Bulgaria/Bulgarian-Lotus	BG_BG_L	1025	Both
Byelorussia/Byelorussian	BE_BY	1025	Source
Canada/English	EN_CA	37	Source
Canada/French	FR_CA	500	Both
Chile/Spanish	ES_CL	284	Source
China/Simplified Chinese	ZH_CN	1388	Both
China/Simplified Chinese (old)	ZH_CN_GBK	935	Source
China (Hong Kong S.A.R.)/English	EN_HK	37	Source
China (Hong Kong S.A.R.)/Simplified Chinese	ZH_HK_S	1388	Source
China (Hong Kong S.A.R.)/Traditional Chinese	ZH_HK_T	937	Source
Columbia/Spanish	ES_CO	284	Source
Costa Rica/Spanish	ES_CR	284	Source
Croatia/Croatian	HR_HR	870	Both
Czech Republic/Czech	CS_CZ	870	Both
Denmark/Danish	DA_DK	1142	Both
Denmark/Danish euro	DA_DK_E	1142	Source
Denmark/Danish preeuro	DA_DK_PE	277	Source
Dominican Republic/Spanish	ES_DO	284	Source
Ecuador/Spanish	ES_EC	284	Source
Egypt/Arabic	AR_EG	420	Source
El Salvador/Spanish	ES_SV	284	Source
Estonia/Estonian	ET_EE	1122	Both
Finland/Finnish	FI_FI	1143	Both
Finland/Finnish euro	FI_FI_E	1143	Both
Finland/Finnish preeuro	FI_FI_PE	278	Both
France/French	FR_FR	1147	Both

Description	Member	CCSID	How shipped
France/French euro	FR_FR_E	1147	Both
France/French preeuro	FR_FR_PE	297	Both
Germany/German	DE_DE	1141	Both
Germany/German euro	DE_DE_E	1141	Both
Germany/German preeuro	DE_DE_PE	273	Both
Great Britain/English	EN_GB	285	Both
Greece/Greek	EL_GR	875	Both
Greece/Greek euro	EL_GR_E	875	Source
Greece/Greek preeuro	EL_GR_PE	875	Source
Guatemala/Spanish	ES_GT	284	Source
Honduras/Spanish	ES_HN	284	Source
Hungary/Hungarian	HU_HU	870	Both
Iceland/Icelandic	IS_IS	871	Both
India/English	EN_IN	37	Source
India/Hindi	HI_IN	1137	Source
India/Tamil	TA_IN	13488	Source
India/Telugu	TE_IN	13488	Source
Indonesia/Indonesian	ID_ID	500	Source
Ireland/English	EN_IE	1140	Source
Ireland/English euro	EN_IE_E	1140	Source
Ireland/English preeuro	EN_IE_PE	37	Source
Israel/Hebrew	HE_IL	424	Both
Israel/English	IW_IL	424	Both
Italy/Italian	IT_IT	1144	Both
Italy/Italian euro	IT_IT_E	1144	Both
Italy/Italian preeuro	IT_IT_PE	280	Both
Japan 13488	JA_13488	13488	Both
Japan 5035	JA_5035	5035	Object
Japan/Japanese	JA_JP	5026	Both
Jordan/Arabic	AR_JO	420	Source
Kazakhstan/Kazakh	KK_KZ	13488	Source
Kuwait/Arabic	AR_KW	420	Source
Latvia/Latvian	LV_LV	1112	Both
Lebanon/Arabic	AR_LB	420	Source
Lithuania/Lithuanian	LT_LT	1112	Both
Luxembourg/French	FR_LU	1147	Source
Luxembourg/French euro	FR_LU_E	1147	Source
Luxembourg/French preeuro	FR_LU_PE	297	Source
Luxembourg/German	DE_LU	1141	Source
Luxembourg/German euro	DE_LU_E	1141	Source

Description	Member	CCSID	How shipped
Luxembourg/German preeuro	DE_LU_PE	273	Source
Macedonia/Macedonian	MK_MK	1025	Both
Macedonia/Macedonian-Lotus	MK_MK_L	1025	Both
Malaysia/Malaysian	MS_MY	500	Source
Mexico/Spanish	ES_MX	284	Source
Morocco/Arabic	AR_MA	420	Source
Netherlands/Dutch	NL_NL	1140	Both
Netherlands/Dutch euro	NL_NL_E	1140	Both
Netherlands/Dutch preeuro	NL_NL_PE	37	Both
New Zealand/English	EN_NZ	37	Source
Nicaragua/Spanish	ES_NI	284	Source
Norway/Norwegian	NO_NO	277	Both
Norway/Norwegian (Bokmal)	NB_NO	277	Source
Oman/Arabic	AR_OM	420	Source
Panama/Spanish	ES_PA	284	Source
Paraguay/Spanish	ES_PY	284	Source
Peru/Spanish	ES_PE	284	Source
Philippines/English	EN_PH	37	Source
Poland/Polish	PL_PL	870	Both
Portugal/Portuguese	PT_PT	1140	Both
Portugal/Portuguese euro	PT_PT_E	1140	Both
Portugal/Portuguese preeuro	PT_PT_PE	37	Both
Puerto Rico/Spanish	ES_PR	284	Source
Qatar/Arabic	AR_QA	420	Source
Romania/Romanian	RO_RO	870	Both
Russia/Russian	RU_RU	1025	Both
Saudi Arabia/Arabic	AR_SA	420	Source
Serbia/Serbian, Cyrillic	SR_SP	1025	Both
Serbia/Serbian, Cyrillic-Lotus	SR_SP_L	1025	Both
Serbia/Serbian, Latin	SH_SP	870	Both
Singapore/English	EN_SG	37	Source
Singapore/Simplified Chinese	ZH_SG	1388	Source
Slovak/Slovakian	SK_SK	870	Both
Slovenia/Slovenian	SL_SI	870	Both
South Africa/English	EN_ZA	37	Source
South Korea/Korean	KO_KR	933	Both

Description	Member	CCSID	How shipped
Spain/Catalan	CA_ES	1145	Source
Spain/Catalan euro	CA_ES_E	1145	Source
Spain/Catalan preeuro	CA_PES_PE	284	Source
Spain/Spanish	ES_ES	1145	Both
Spain/Spanish euro	ES_ES_E	1145	Both
Spain/Spanish preeuro	ES_PES_PE	284	Both
Sweden/Swedish	SV_SE	1143	Both
Sweden/Swedish euro	SV_SE_E	1143	Source
Sweden/Swedish preeuro	SV_SE_PE	278	Source
Switzerland/French	FR_CH	500	Both
Switzerland/French-Lotus	FR_CH_L	500	Both
Switzerland/German	DE_CH	500	Both
Switzerland/German-Lotus	DE_CH_L	500	Both
Switzerland/Italian	IT_CH	500	Source
Syria/Arabic	AR_SY	420	Source
Taiwan/Mandarin, Traditional Chinese	ZH_TW	937	Both
Thailand/Thai	TH_TH	838	Both
Tunisia/Arabic	AR_TN	420	Source
Turkey/Turkish	TR_TR	1026	Both
Turkey/Turkish, English	TR_TR2	1026	Source
Ukraine/Ukrainian	UK_UA	1025	Both
United Arab Emirates/Arabic	AR_AE	420	Source
United Kingdom/English euro	EN_GB_E	1146	Source
United States/Spanish	ES_US	284	Source
Uruguay/Spanish	ES_UY	284	Source
USA/English	EN_US	37	Both
USA/English,Upper Case	EN_UPPER	37	Both
Venezuela/Spanish	ES_VE	284	Source
Vietnam/Vietnamese	VI_VN	1129	Source
Yemen/Arabic	AR_YE	420	Source
Yugoslavia/Serbian, Latin	SH_YU	870	Source
Yugoslavia/Serbian, Cyrillic	SR_YU	1025	Source

Related concepts

"Installing and enabling locales" on page 44
If you are installing a new release, you can request that library QSYSLOCALE be installed on the system at that time.

"Working with locales" on page 200
Locales are used primarily in ILE-based application programs. Additionally, the Retrieve Locale Information (OPM, QLGRTVLC; ILE, QlgRetrieveLocaleInformation) API retrieves one or all categories of a locale.

"Locale categories" on page 201
This table describes the locale categories that are supported on the i5/OS operating system.

"Example: Creating and enabling a locale" on page 225
This example contains the steps necessary for creating and enabling a locale.

Mapping of locale symbolic names

This table lists locale symbolic names.

The table shows the following items:

- Some of the common locale symbolic names used by i5/OS and the UCS-2 code points they represent
- When appropriate, a comparison of code points between UCS-2 code page and various IBM code pages
- The IBM graphic character global identifier (GCGID)
- An illustration of the GCGID

This table shows code points up through x'00FF'. The locale compiler understands most of the names defined by the Unicode organization. See the Unicode Web site (http://www.unicode.org) for an extended listing.

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<NUL>	0000	NULL (NUL)	037/00		
<SOH>	0001	START OF HEADING (SOH)	037/01		
<STX>	0002	START OF TEXT (STX)	037/02		
<ETX>	0003	END OF TEXT (ETX)	037/03		
<EOT>	0004	END OF TRANSMISSION (EOT)	037/37		
<ENQ>	0005	ENQUIRY (ENQ)	037/2D		
<ACK>	0006	ACKNOWLEDGE (ACK)	037/2E		
<alert>	0007	BELL	037/2F		
<BEL>	0007	BELL (BEL)	037/2F		
<BS>	0008	BACKSPACE (BS)	037/16		
<backspace>	0008	BACKSPACE	037/16		
<tab>	0009	CHARACTER TABULATION	037/05		
<HT>	0009	CHARACTER TABULATION (HT)	037/05		
<newline>	000A	LINE FEED	037/25		
<LF>	000A	LINE FEED (LF)	037/25		
<vertical-tab>	000B	LINE TABULATION	037/0B		
<VT>	000B	LINE TABULATION (VT)	037/0B		
<FF>	000C	FORM FEED (FF)	037/0C		
<form-feed>	000C	FORM FEED	037/0C		
<carriage-return>	000D	CARRIAGE RETURN	037/0D		
<SO>	000E	SHIFT OUT	037/0E		

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<SI>	000F	SHIFT IN	037/0F		
<DLE>	0010	DATALINK ESCAPE (DLE)	037/10		
<DC1>	0011	DEVICE CONTROL ONE (DC1)	037/11		
<DC2>	0012	DEVICE CONTROL TWO (DC2)	037/12		
<DC3>	0013	DEVICE CONTROL THREE (DC3)	037/13		
<DC4>	0014	DEVICE CONTROL FOUR (DC4)	037/3C		
<NAK>	0015	NEGATIVE ACKNOWLEDGE (NAK)	037/3D		
<SYN>	0016	SYNCRONOUS IDLE (SYN)	037/32		
<ETB>	0017	END OF TRANSMISSION BLOCK (ETB)	037/26		
<CAN>	0018	CANCEL (CAN)	037/18		
<EM>	0019	END OF MEDIA	037/19		
<SUB>	001A	SUBSTITUTE (SUB)	037/3F		
<ESC>	001B	ESCAPE (ESC)	037/27		
<IS4>	001C	FILE SEPARATOR (IS4)	037/1C		
<FS>	001C	FILE SEPARATOR (IS4)	037/1C		
<IS3>	001D	GROUP SEPARATOR (IS3)	037/1D		
<GS>	001D	GROUP SEPARATOR (IS3)	037/1D		
<IS2>	001E	RECORD SEPARATOR (IS2)	037/1E		
<RS>	001E	RECORD SEPARATOR (IS2)	037/1E		
<IS1>	001F	UNIT SEPARATOR (IS1)	037/1F		
<US>	001F	UNIT SEPARATOR (IS1)	037/1F		
<DEL>	007F	DELETE (DEL)	037/07		
<space>	0020	SPACE	037/40	SP010000	SP
<exclamation-mark>	0021	EXCLAMATION MARK	500/4F	SP020000	!
<quotation-mark>	0022	QUOTATION MARK	500/7F	SP040000	"
<number-sign>	0023	NUMBER SIGN	500/7B	SM010000	#
<dollar-sign>	0024	DOLLAR SIGN	500/5B	SC030000	$
<percent-sign>	0025	PERCENT SIGN	500/6C	SM020000	%

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<ampersand>	0026	AMPERSAND	500/50	SM030000	&
<apostrophe>	0027	APOSTROPHE	500/7D	SP050000	'
<left-parenthesis>	0028	LEFT PARENTHESIS	500/4D	SP060000	(
<right-parenthesis>	0029	RIGHT PARENTHESIS	500/5D	SP070000	)
<asterisk>	002A	ASTERISK	500/5C	SM040000	*
<plus-sign>	002B	PLUS SIGN	500/4E	SA010000	+
<comma>	002C	COMMA	500/6B	SP080000	,
<hyphen>	002D	HYPHEN	500/60	SP100000	-
<hyphen-minus>	002D	HYPHEN-MINUS	500/60	SP100000	-
<period>	002E	PERIOD	500/4B	SP110000	.
<full-stop>	002E	FULL STOP	500/4B	SP110000	.
<slash>	002F	SLASH	500/61	SP120000	/
<solidus>	002F	SOLIDUS	500/61	SP120000	/
<zero>	0030	DIGIT ZERO	500/F0	ND100000	0
<one>	0031	DIGIT ONE	500/F1	ND010000	1
<two>	0032	DIGIT TWO	500/F2	ND020000	2
<three>	0033	DIGIT THREE	500/F3	ND030000	3
<four>	0034	DIGIT FOUR	500/F4	ND040000	4
<five>	0035	DIGIT FIVE	500/F5	ND050000	5
<six>	0036	DIGIT SIX	500/F6	ND060000	6
<seven>	0037	DIGIT SEVEN	500/F7	ND070000	7
<eight>	0038	DIGIT EIGHT	500/F8	ND080000	8
<nine>	0039	DIGIT NINE	500/F9	ND090000	9
<colon>	003A	COLON	500/7A	SP130000	:
<semicolon>	003B	SEMICOLON	500/5E	SP140000	;
<less-than-sign>	003C	LESS-THAN SIGN	500/4C	SA030000	<

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<equals-sign>	003D	EQUALS SIGN	500/7E	SA040000	=
<greater-than-sign>	003E	GREATER-THAN SIGN	500/6E	SA050000	>
<question-mark>	003F	QUESTION MARK	500/6F	SP150000	?
<commercial-at>	0040	COMMERCIAL AT	500/7C	SM050000	@
<A>	0041	LATIN CAPITAL LETTER A	500/C1	LA020000	A
<B>	0042	LATIN CAPITAL LETTER B	500/C2	LB020000	B
<C>	0043	LATIN CAPITAL LETTER C	500/C3	LC020000	C
<D>	0044	LATIN CAPITAL LETTER D	500/C4	LD020000	D
<E>	0045	LATIN CAPITAL LETTER E	500/C5	LE020000	E
<F>	0046	LATIN CAPITAL LETTER F	500/C6	LF020000	F
<G>	0047	LATIN CAPITAL LETTER G	500/C7	LG020000	G
<H>	0048	LATIN CAPITAL LETTER H	500/C8	LH020000	H
<I>	0049	LATIN CAPITAL LETTER I	500/C9	LI020000	I
<J>	004A	LATIN CAPITAL LETTER J	500/D1	LJ020000	J
<K>	004B	LATIN CAPITAL LETTER K	500/D2	LK020000	K
<L>	004C	LATIN CAPITAL LETTER L	500/D3	LL020000	L
<M>	004D	LATIN CAPITAL LETTER M	500/D4	LM020000	M
<N>	004E	LATIN CAPITAL LETTER N	500/D5	LN020000	N
<O>	004F	LATIN CAPITAL LETTER O	500/D6	LO020000	O
	0050	LATIN CAPITAL LETTER P	500/D7	LP020000	P
<Q>	0051	LATIN CAPITAL LETTER Q	500/D8	LQ020000	Q
<R>	0052	LATIN CAPITAL LETTER R	500/D9	LR020000	R
<S>	0053	LATIN CAPITAL LETTER S	500/E2	LS020000	S
<T>	0054	LATIN CAPITAL LETTER T	500/E3	LT020000	T
<U>	0055	LATIN CAPITAL LETTER U	500/E4	LU020000	U
<V>	0056	LATIN CAPITAL LETTER V	500/E5	LV020000	V

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<W>	0057	LATIN CAPITAL LETTER W	500/E6	LW020000	W
<X>	0058	LATIN CAPITAL LETTER X	500/E7	LX020000	X
<Y>	0059	LATIN CAPITAL LETTER Y	500/E8	LY020000	Y
<Z>	005A	LATIN CAPITAL LETTER Z	500/E9	LZ020000	Z
<left-square-bracket>	005B	LEFT SQUARE BRACKET	500/4A	SM060000	[
<backslash>	005C	BACKSLASH	500/E0	SM070000	\
<reverse-solidus>	005C	REVERSE SOLIDUS	500/E0	SM070000	\
<right-square-bracket>	005D	RIGHT SQUARE BRACKET	500/5A	SM080000	]
<circumflex>	005E	CIRCUMFLEX	500/5F	SD150000	^
<circumflex-accent>	005E	CIRCUMFLEX ACCENT	500/5F	SD150000	^
<underscore>	005F	UNDERSCORE	500/6D	SP090000	_
<underline>	005F	UNDERLINE	500/6D	SP090000	_
<low-line>	005F	LOW LINE	500/6D	SP090000	_
<grave-accent>	0060	GRAVE ACCENT	500/79	SD130000	`
<a>	0061	LATIN SMALL LETTER A	500/81	LA010000	a
<b>	0062	LATIN SMALL LETTER B	500/82	LB010000	b
<c>	0063	LATIN SMALL LETTER C	500/83	LC010000	c
<d>	0064	LATIN SMALL LETTER D	500/84	LD010000	d
<e>	0065	LATIN SMALL LETTER E	500/85	LE010000	e
<f>	0066	LATIN SMALL LETTER F	500/86	LF010000	f
<g>	0067	LATIN SMALL LETTER G	500/87	LG010000	g
<h>	0068	LATIN SMALL LETTER H	500/88	LH010000	h
<i>	0069	LATIN SMALL LETTER I	500/89	LI010000	i
<j>	006A	LATIN SMALL LETTER J	500/91	LJ010000	j
<k>	006B	LATIN SMALL LETTER K	500/92	LK010000	k
<l>	006C	LATIN SMALL LETTER L	500/93	LL010000	l

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<m>	006D	LATIN SMALL LETTER M	500/94	LM010000	m
<n>	006E	LATIN SMALL LETTER N	500/95	LN010000	n
<o>	006F	LATIN SMALL LETTER O	500/96	LO010000	o
	0070	LATIN SMALL LETTER P	500/97	LP010000	p
<q>	0071	LATIN SMALL LETTER Q	500/98	LQ010000	q
<r>	0072	LATIN SMALL LETTER R	500/99	LR010000	r
<s>	0073	LATIN SMALL LETTER S	500/A2	LS010000	s
<t>	0074	LATIN SMALL LETTER T	500/A3	LT010000	t
<u>	0075	LATIN SMALL LETTER U	500/A4	LU010000	u
<v>	0076	LATIN SMALL LETTER V	500/A5	LV010000	v
<w>	0077	LATIN SMALL LETTER W	500/A6	LW010000	w
<x>	0078	LATIN SMALL LETTER X	500/A7	LX010000	x
<y>	0079	LATIN SMALL LETTER Y	500/A8	LY010000	y
<z>	007A	LATIN SMALL LETTER Z	500/A9	LZ010000	z
<left-brace>	007B	LEFT BRACE	500/C0	SM110000	{
<left-curly-bracket>	007B	LEFT CURLY BRACKET	500/C0	SM110000	{
<vertical-line>	007C	VERTICAL LINE	500/BB	SM130000	|
<right-brace>	007D	RIGHT BRACE	500/D0	SM140000	}
<right-curly-bracket>	007D	RIGHT CURLY BRACKET	500/D0	SM140000	}
<tilde>	007E	TILDE	500/A1	SD190000	~
<BPH>	0082	BREAK PERMITTED HERE	037/22		
<NBH>	0083	NO BREAK HERE	037/23		
<IND>	0084	INDEX	037/24		
<NEL>	0085	NEXT LINE	037/15		
<SSA>	0086	START OF SELECTED AREA	037/06		
<ESA>	0087	END OF SELECTED AREA	037/17		
<HTS>	0088	CHARACTER TABULATION SET	037/28		

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<HTJ>	0089	CHARACTER TABULATION WITH JUSTIFICATION	037/29		
<VTS>	008A	LINE TABULATION SET	037/2A		
<PLD>	008B	PARTIAL LINE DOWN	037/2B		
<PLU>	008C	PARTIAL LINE UP	037/2C		
<RI>	008D	REVERSE INDEX	037/09		
<SS2>	008E	SINGLE SHIFT TWO	037/0A		
<SS3>	008F	SINGLE SHIFT THREE	037/1B		
<DCS>	0090	DEVICE CONTROL STRING	037/30		
<PU1>	0091	PRIVATE USE ONE	037/31		
<PU2>	0092	PRIVATE USE TWO	037/1A		
<STS>	0093	SET TRANSMIT STATE	037/33		
<CCH>	0094	CANCEL CHARACTER	037/34		
<MW>	0095	MESSAGE WAITING	037/35		
<SPS>	0096	START OF GUARDED AREA	037/36		
<EPA>	0097	END OF GUARDED AREA	037/08		
<SOS>	0098	START OF STRING	037/38		
<SCI>	009A	SINGLE CHARACTER INTRODUCER	037/39		
<CSI>	009B	CONTROL SEQUENCE INTRODUCER	037/3B		
<ST>	009C	STRING TERMINATOR	037/04		
<OSC>	009D	OPERATING SYSTEM COMMAND	037/14		
<PM>	009E	PRIVACY MESSAGE	037/3E		
<APC>	009F	APPLICATION PROGRAM COMMAND	037/FF		
<nobreakspace>	00A0	NO-BREAK SPACE	500/41	SP300000	(NBSP)
<exclamation-down>	00A1	INVERTED EXCLAMATION MARK	500/AA	SP030000	¡
<cent>	00A2	CENT SIGN	500/B0	SC040000	¢

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<sterling>	00A3	POUND SIGN	500/B1	SC020000	£
<currency>	00A4	CURRENCY SIGN	500/9F	SC010000	¤
<yen>	00A5	YEN SIGN	500/B2	SC050000	¥
<yen-sign>	00A5	YEN SIGN	850/BE	SC050000	¥
<broken-bar>	00A6	BROKEN VERTICAL BAR	500/6A	SM650000	¦
<section>	00A7	SECTION SIGN	500/B5	SM240000	§
<diaresis>	00A8	SPACING DIAERESIS	500/8D	SD170000	¨
<copyright>	00A9	COPYRIGHT SIGN	500/B4	SM520000	©
<feminine>	00AA	FEMININE ORDINAL INDICATOR	500/9A	SM210000	ª
<guillemot-left>	00AB	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK	500/8A	SP170000	«
<not>	00AC	NOT SIGN	500/BA	SM660000	¬
<dash>	00AD	SOFT HYPHEN	500/CA	SP320000	SHY
<registered>	00AE	REGISTERED TRADE MARK SIGN	500/AF	SM530000	®
<macron>	00AF	SPACING MACRON	500/BC	SM150000	¯
<degree>	00B0	DEGREE SIGN	500/90	SM190000	°
<plus-minus>	00B1	PLUS-OR-MINUS SIGN	500/8F	SA020000	±
<two-superior>	00B2	SUPERSCRIPT DIGIT TWO	500/EA	ND021000	²
<three-superior>	00B3	SUPERSCRIPT DIGIT THREE	500/FA	ND031000	³
<acute>	00B4	SPACING ACUTE	500/BE	SD110000	´
<mu>	00B5	MICRO SIGN	500/A0	SM170000	µ
<paragraph>	00B6	PARAGRAPH (PILCROW) SIGN	500/B6	SM250000	¶
<dot>	00B7	MIDDLE DOT	500/B3	SD630000	·
<cedilla>	00B8	SPACING CEDILLA	500/9D	SD410000	¸
<one-superior>	00B9	SUPERSCRIPT DIGIT ONE	500/DA	ND011000	¹

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<masculine>	00BA	MASCULINE ORDINAL INDICATOR	500/9B	SM200000	º
<guillemot-right>	00BB	RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK	500/8B	SP180000	»
<one-quarter>	00BC	FRACTION ONE QUARTER	500/B7	NF040000	¼
<one-half>	00BD	FRACTION ONE HALF	500/B8	NF010000	½
<three-quarters>	00BE	FRACTION THREE QUARTERS	500/B9	NF050000	¾
<question-down>	00BF	INVERTED QUESTION MARK	500/AB	SP160000	¿
<A-grave>	00C0	LATIN CAPITAL LETTER A GRAVE	500/64	LA140000	À
<A-acute>	00C1	LATIN CAPITAL LETTER A ACUTE	500/65	LA120000	Á
<A-circumflex>	00C2	LATIN CAPITAL LETTER A CIRCUMFLEX	500/62	LA160000	Â
<A-tilde>	00C3	LATIN CAPITAL LETTER A TILDE	500/66	LA200000	Ã
<A-diaresis>	00C4	LATIN CAPITAL LETTER A DIAERESIS	500/63	LA180000	Ä
<A-ring>	00C5	LATIN CAPITAL LETTER A RING ABOVE	500/67	LA280000	Å
<AE>	00C6	LATIN CAPITAL LIGATURE AE	500/9E	LA520000	Æ
<C-cedilla>	00C7	LATIN CAPITAL LETTER C CEDILLA	500/68	LC420000	Ç
<E-grave>	00C8	LATIN CAPITAL LETTER E GRAVE	500/74	LE140000	È
<E-acute>	00C9	LATIN CAPITAL LETTER E ACUTE	500/71	LE120000	É
<E-circumflex>	00CA	LATIN CAPITAL LETTER E CIRCUMFLEX	500/72	LE160000	Ê
<E-diaresis>	00CB	LATIN CAPITAL LETTER E DIAERESIS	500/73	LE180000	Ë
<I-grave>	00CC	LATIN CAPITAL LETTER I GRAVE	500/78	LI140000	Ì
<I-acute>	00CD	LATIN CAPITAL LETTER I ACUTE	500/75	LI120000	Í
<I-circumflex>	00CE	LATIN CAPITAL LETTER I CIRCUMFLEX	500/76	LI160000	Î
<I-diaresis>	00CF	LATIN CAPITAL LETTER I DIAERESIS	500/77	LI180000	Ï

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<Eth>	00D0	LATIN CAPITAL LETTER ETH (Icelandic)	500/AC	LD620000	Đ
<N-tilde>	00D1	LATIN CAPITAL LETTER N TILDE	500/69	LN200000	Ñ
<O-grave>	00D2	LATIN CAPITAL LETTER O GRAVE	500/ED	LO140000	Ò
<O-acute>	00D3	LATIN CAPITAL LETTER O ACUTE	500/EE	LO120000	Ó
<O-circumflex>	00D4	LATIN CAPITAL LETTER O CIRCUMFLEX	500/EB	LO160000	Ô
<O-tilde>	00D5	LATIN CAPITAL LETTER O TILDE	500/EF	LO200000	Õ
<O-diaresis>	00D6	LATIN CAPITAL LETTER O DIAERESIS	500/EC	LO180000	Ö
<multiply>	00D7	MULTIPLICATION SIGN	500/BF	SA070000	×
<O-slash>	00D8	LATIN CAPITAL LETTER O STROKE	500/80	LO620000	Ø
<U-grave>	00D9	LATIN CAPITAL LETTER U GRAVE	500/FD	LU140000	Ù
<U-acute>	00DA	LATIN CAPITAL LETTER U ACUTE	500/FE	LU120000	Ú
<U-circumflex>	00DB	LATIN CAPITAL LETTER U CIRCUMFLEX	500/FB	LU160000	Û
<U-diaresis>	00DC	LATIN CAPITAL LETTER U DIAERESIS	500/FC	LU180000	Ü
<Y-acute>	00DD	LATIN CAPITAL LETTER Y ACUTE	500/AD	LY120000	Ý
<Thorn>	00DE	LATIN CAPITAL LETTER THORN (Icelandic)	500/AE	LT640000	Þ
<s-sharp>	00DF	LATIN SMALL LETTER SHARP S (German)	500/59	LS610000	ß
<a-grave>	00E0	LATIN SMALL LETTER A GRAVE	500/44	LA130000	à
<a-acute>	00E1	LATIN SMALL LETTER A ACUTE	500/45	LA110000	á
<a-circumflex>	00E2	LATIN SMALL LETTER A CIRCUMFLEX	500/42	LA150000	â
<a-tilde>	00E3	LATIN SMALL LETTER A TILDE	500/46	LA190000	ã
<a-diaresis>	00E4	LATIN SMALL LETTER A DIAERESIS	500/43	LA170000	ä
<a-ring>	00E5	LATIN SMALL LETTER A RING ABOVE	500/47	LA270000	å

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<ae>	00E6	LATIN SMALL LIGATURE AE	500/9C	LA510000	æ
<c-cedilla>	00E7	LATIN SMALL LETTER C CEDILLA	500/48	LC410000	ç
<e-grave>	00E8	LATIN SMALL LETTER E GRAVE	500/54	LE130000	è
<e-acute>	00E9	LATIN SMALL LETTER E ACUTE	500/51	LE110000	é
<e-circumflex>	00EA	LATIN SMALL LETTER E CIRCUMFLEX	500/52	LE150000	ê
<e-diaresis>	00EB	LATIN SMALL LETTER E DIAERESIS	500/53	LE170000	ë
<i-grave>	00EC	LATIN SMALL LETTER I GRAVE	500/58	LI130000	ì
<i-acute>	00ED	LATIN SMALL LETTER I ACUTE	500/55	LI110000	í
<i-circumflex>	00EE	LATIN SMALL LETTER I CIRCUMFLEX	500/56	LI150000	î
<i-diaresis>	00EF	LATIN SMALL LETTER I DIAERESIS	500/57	LI170000	ï
<eth>	00F0	LATIN SMALL LETTER ETH (Icelandic)	500/8C	LD630000	ð
<n-tilde>	00F1	LATIN SMALL LETTER N TILDE	500/49	LN190000	ñ
<o-grave>	00F2	LATIN SMALL LETTER O GRAVE	500/CD	LO130000	ò
<o-acute>	00F3	LATIN SMALL LETTER O ACUTE	500/CE	LO110000	ó
<o-circumflex>	00F4	LATIN SMALL LETTER O CIRCUMFLEX	500/CB	LO150000	ô
<o-tilde>	00F5	LATIN SMALL LETTER O TILDE	500/CF	LO190000	õ
<o-diaresis>	00F6	LATIN SMALL LETTER O DIAERESIS	500/CC	LO170000	ö
<divide>	00F7	DIVISION SIGN	500/E1	SA060000	÷
<division>	00F7	DIVISION SIGN	500/E1	SA060000	÷
<o-slash>	00F8	LATIN SMALL LETTER O STROKE	500/70	LO610000	ø
<u-grave>	00F9	LATIN SMALL LETTER U GRAVE	500/DD	LU130000	ù
<u-acute>	00FA	LATIN SMALL LETTER U ACUTE	500/DE	LU110000	ú

Symbolic name	Unicode (ISO 10646) code point (hexadecimal)	Description	IBM code page and code point (xxx/xx)	IBM GCGID	GCGID illustration
<u-circumflex>	00FB	LATIN SMALL LETTER U CIRCUMFLEX	500/DB	LU150000	û
<u-diaresis>	00FC	LATIN SMALL LETTER U DIAERESIS	500/DC	LU170000	ü
<y-acute>	00FD	LATIN SMALL LETTER Y ACUTE	500/8D	LY110000	ý
<thorn>	00FE	LATIN SMALL LETTER THORN (Icelandic)	500/8E	LT630000	þ
<y-diaresis>	00FF	LATIN SMALL LETTER Y DIAERESIS	500/DF	LY170000	ÿ

Related concepts

"Locale symbolic names" on page 220
The i5/OS operating system supports locale symbolic names based on predefined names from the X/Open Standard portable character set.

Related information

Unicode Home Page

UCS-2 level-1 mapping tables

You can use the Convert Case API to convert characters encoded in universal character set 2 level 1 (UCS-2 level-1) between uppercase and lowercase.

Related information

Convert Case (QLGCNVCS, QlgConvertCase) API

Unicode uppercase to lowercase conversion mapping table

The table shows the mapping for the conversion of Unicode from uppercase to lowercase. You can refer to this table to see how the Convert Case API converts character data from uppercase to lowercase.

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
0041	0061	LATIN CAPITAL LETTER A	LATIN SMALL LETTER A
0042	0062	LATIN CAPITAL LETTER B	LATIN SMALL LETTER B
0043	0063	LATIN CAPITAL LETTER C	LATIN SMALL LETTER C
0044	0064	LATIN CAPITAL LETTER D	LATIN SMALL LETTER D
0045	0065	LATIN CAPITAL LETTER E	LATIN SMALL LETTER E
0046	0066	LATIN CAPITAL LETTER F	LATIN SMALL LETTER F
0047	0067	LATIN CAPITAL LETTER G	LATIN SMALL LETTER G
0048	0068	LATIN CAPITAL LETTER H	LATIN SMALL LETTER H
0049	0069	LATIN CAPITAL LETTER I	LATIN SMALL LETTER I
004A	006A	LATIN CAPITAL LETTER J	LATIN SMALL LETTER J
004B	006B	LATIN CAPITAL LETTER K	LATIN SMALL LETTER K
004C	006C	LATIN CAPITAL LETTER L	LATIN SMALL LETTER L
004D	006D	LATIN CAPITAL LETTER M	LATIN SMALL LETTER M
004E	006E	LATIN CAPITAL LETTER N	LATIN SMALL LETTER N
004F	006F	LATIN CAPITAL LETTER O	LATIN SMALL LETTER O
0050	0070	LATIN CAPITAL LETTER P	LATIN SMALL LETTER P

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
0051	0071	LATIN CAPITAL LETTER Q	LATIN SMALL LETTER Q
0052	0072	LATIN CAPITAL LETTER R	LATIN SMALL LETTER R
0053	0073	LATIN CAPITAL LETTER S	LATIN SMALL LETTER S
0054	0074	LATIN CAPITAL LETTER T	LATIN SMALL LETTER T
0055	0075	LATIN CAPITAL LETTER U	LATIN SMALL LETTER U
0056	0076	LATIN CAPITAL LETTER V	LATIN SMALL LETTER V
0057	0077	LATIN CAPITAL LETTER W	LATIN SMALL LETTER W
0058	0078	LATIN CAPITAL LETTER X	LATIN SMALL LETTER X
0059	0079	LATIN CAPITAL LETTER Y	LATIN SMALL LETTER Y
005A	007A	LATIN CAPITAL LETTER Z	LATIN SMALL LETTER Z
00C0	00E0	LATIN CAPITAL LETTER A GRAVE	LATIN SMALL LETTER A GRAVE
00C1	00E1	LATIN CAPITAL LETTER A ACUTE	LATIN SMALL LETTER A GRAVE
00C2	00E2	LATIN CAPITAL LETTER A CIRCUMFLEX	LATIN SMALL LETTER A GRAVE
00C3	00E3	LATIN CAPITAL LETTER A TILDE	LATIN SMALL LETTER A GRAVE
00C4	00E4	LATIN CAPITAL LETTER A DIAERESIS	LATIN SMALL LETTER A GRAVE
00C5	00E5	LATIN CAPITAL LETTER A RING	LATIN SMALL LETTER A GRAVE
00C6	00E6	LATIN CAPITAL LETTER A E	LATIN SMALL LETTER A GRAVE
00C7	00E7	LATIN CAPITAL LETTER C CEDILLA	LATIN SMALL LETTER A GRAVE
00C8	00E8	LATIN CAPITAL LETTER E GRAVE	LATIN SMALL LETTER A GRAVE
00C9	00E9	LATIN CAPITAL LETTER E ACUTE	LATIN SMALL LETTER A GRAVE
00CA	00EA	LATIN CAPITAL LETTER E CIRCUMFLEX	LATIN SMALL LETTER E CIRCUMFLEX
00CB	00EB	LATIN CAPITAL LETTER E DIAERESIS	LATIN SMALL LETTER E DIAERESIS
00CC	00EC	LATIN CAPITAL LETTER I GRAVE	LATIN SMALL LETTER I GRAVE
00CD	00ED	LATIN CAPITAL LETTER I ACUTE	LATIN SMALL LETTER I ACUTE
00CE	00EE	LATIN CAPITAL LETTER I CIRCUMFLEX	LATIN SMALL LETTER I CIRCUMFLEX
00CF	00EF	LATIN CAPITAL LETTER I DIAERESIS	LATIN SMALL LETTER I DIAERESIS
00D0	00F0	LATIN CAPITAL LETTER ETH	LATIN SMALL LETTER ETH
00D1	00F1	LATIN CAPITAL LETTER N TILDE	LATIN SMALL LETTER N TILDE
00D2	00F2	LATIN CAPITAL LETTER O GRAVE	LATIN SMALL LETTER O GRAVE
00D3	00F3	LATIN CAPITAL LETTER O ACUTE	LATIN SMALL LETTER O ACUTE
00D4	00F4	LATIN CAPITAL LETTER O CIRCUMFLEX	LATIN SMALL LETTER O CIRCUMFLEX
00D5	00F5	LATIN CAPITAL LETTER O TILDE	LATIN SMALL LETTER O TILDE
00D6	00F6	LATIN CAPITAL LETTER O DIAERESIS	LATIN SMALL LETTER O DIAERESIS
00D8	00F8	LATIN CAPITAL LETTER O SLASH	LATIN SMALL LETTER O SLASH
00D9	00F9	LATIN CAPITAL LETTER U GRAVE	LATIN SMALL LETTER U GRAVE
00DA	00FA	LATIN CAPITAL LETTER U ACUTE	LATIN SMALL LETTER U ACUTE
00DB	00FB	LATIN CAPITAL LETTER U CIRCUMFLEX	LATIN SMALL LETTER U CIRCUMFLEX
00DC	00FC	LATIN CAPITAL LETTER U DIAERESIS	LATIN SMALL LETTER U DIAERESIS

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
00DD	00FD	LATIN CAPITAL LETTER Y ACUTE	LATIN SMALL LETTER Y ACUTE
00DE	00FE	LATIN CAPITAL LETTER THORN	LATIN SMALL LETTER THORN
0100	0101	LATIN CAPITAL LETTER A WITH MACRON	LATIN SMALL LETTER A WITH MACRON
0102	0103	LATIN CAPITAL LETTER A WITH BREVE	LATIN SMALL LETTER A WITH BREVE
0104	0105	LATIN CAPITAL LETTER A WITH OGONEK	LATIN SMALL LETTER A WITH OGONEK
0106	0107	LATIN CAPITAL LETTER C WITH ACUTE	LATIN SMALL LETTER C WITH ACUTE
0108	0109	LATIN CAPITAL LETTER C WITH CIRCUMFLEX	LATIN SMALL LETTER C WITH CIRCUMFLEX
010A	010B	LATIN CAPITAL LETTER C WITH DOT ABOVE	LATIN SMALL LETTER C WITH DOT ABOVE
010C	010D	LATIN CAPITAL LETTER C WITH CARON	LATIN SMALL LETTER C WITH CARON
010E	010F	LATIN CAPITAL LETTER D WITH CARON	LATIN SMALL LETTER D WITH CARON
0110	0111	LATIN CAPITAL LETTER D WITH STROKE	LATIN SMALL LETTER D WITH STROKE
0112	0113	LATIN CAPITAL LETTER E WITH MACRON	LATIN SMALL LETTER E WITH MACRON
0114	0115	LATIN CAPITAL LETTER E WITH BREVE	LATIN SMALL LETTER E WITH BREVE
0116	0117	LATIN CAPITAL LETTER E WITH DOT ABOVE	LATIN SMALL LETTER E WITH DOT ABOVE
0118	0119	LATIN CAPITAL LETTER E WITH OGONEK	LATIN SMALL LETTER E WITH OGONEK
011A	011B	LATIN CAPITAL LETTER E WITH CARON	LATIN SMALL LETTER E WITH CARON
011C	011D	LATIN CAPITAL LETTER G WITH CIRCUMFLEX	LATIN SMALL LETTER G WITH CIRCUMFLEX
011E	011F	LATIN CAPITAL LETTER G WITH BREVE	LATIN SMALL LETTER G WITH BREVE
0120	0121	LATIN CAPITAL LETTER G WITH DOT ABOVE	LATIN SMALL LETTER G WITH DOT ABOVE
0122	0123	LATIN CAPITAL LETTER G WITH CEDILLA	LATIN SMALL LETTER G WITH CEDILLA
0124	0125	LATIN CAPITAL LETTER H WITH CIRCUMFLEX	LATIN SMALL LETTER H WITH CIRCUMFLEX
0126	0127	LATIN CAPITAL LETTER H WITH STROKE	LATIN SMALL LETTER H WITH STROKE
0128	0129	LATIN CAPITAL LETTER I WITH TILDE	LATIN SMALL LETTER I WITH TILDE
012A	012B	LATIN CAPITAL LETTER I WITH MACRON	LATIN SMALL LETTER I WITH MACRON
012C	012D	LATIN CAPITAL LETTER I WITH BREVE	LATIN SMALL LETTER I WITH BREVE
012E	012F	LATIN CAPITAL LETTER I WITH OGONEK	LATIN SMALL LETTER I WITH OGONEK
0130	0069	LATIN CAPITAL LETTER I WITH DOT ABOVE	LATIN SMALL LETTER I
0132	0133	LATIN CAPITAL LIGATURE IJ	LATIN SMALL LIGATURE IJ
0134	0135	LATIN CAPITAL LETTER J WITH CIRCUMFLEX	LATIN SMALL LETTER J WITH CIRCUMFLEX
0136	0137	LATIN CAPITAL LETTER K WITH CEDILLA	LATIN SMALL LETTER K WITH CEDILLA
0139	013A	LATIN CAPITAL LETTER L WITH ACUTE	LATIN SMALL LETTER L WITH ACUTE

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
013B	013C	LATIN CAPITAL LETTER L WITH CEDILLA	LATIN SMALL LETTER L WITH CEDILLA
013D	013E	LATIN CAPITAL LETTER L WITH CARON	LATIN SMALL LETTER L WITH CARON
013F	0140	LATIN CAPITAL LETTER L WITH MIDDLE DOT	LATIN SMALL LETTER L WITH MIDDLE DOT
0141	0142	LATIN CAPITAL LETTER L WITH STROKE	LATIN SMALL LETTER L WITH STROKE
0143	0144	LATIN CAPITAL LETTER N WITH ACUTE	LATIN SMALL LETTER N WITH ACUTE
0145	0146	LATIN CAPITAL LETTER N WITH CEDILLA	LATIN SMALL LETTER N WITH CEDILLA
0147	0148	LATIN CAPITAL LETTER N WITH CARON	LATIN SMALL LETTER N WITH CARON
014A	014B	LATIN CAPITAL LETTER ENG (SAMI)	LATIN SMALL LETTER ENG (SAMI)
014C	014D	LATIN CAPITAL LETTER O WITH MACRON	LATIN SMALL LETTER O WITH MACRON
014E	014F	LATIN CAPITAL LETTER O WITH BREVE	LATIN SMALL LETTER O WITH BREVE
0150	0151	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE	LATIN SMALL LETTER O WITH DOUBLE ACUTE
0152	0153	LATIN CAPITAL LIGATURE OE	LATIN SMALL LIGATURE OE
0154	0155	LATIN CAPITAL LETTER R WITH ACUTE	LATIN SMALL LETTER R WITH ACUTE
0156	0157	LATIN CAPITAL LETTER R WITH CEDILLA	LATIN SMALL LETTER R WITH CEDILLA
0158	0159	LATIN CAPITAL LETTER R WITH CARON	LATIN SMALL LETTER R WITH CARON
015A	015B	LATIN CAPITAL LETTER S WITH ACUTE	LATIN SMALL LETTER S WITH ACUTE
015C	015D	LATIN CAPITAL LETTER S WITH CIRCUMFLEX	LATIN SMALL LETTER S WITH CIRCUMFLEX
015E	015F	LATIN CAPITAL LETTER S WITH CEDILLA	LATIN SMALL LETTER S WITH CEDILLA
0160	0161	LATIN CAPITAL LETTER S WITH CARON	LATIN SMALL LETTER S WITH CARON
0162	0163	LATIN CAPITAL LETTER T WITH CEDILLA	LATIN SMALL LETTER T WITH CEDILLA
0164	0165	LATIN CAPITAL LETTER T WITH CARON	LATIN SMALL LETTER T WITH CARON
0166	0167	LATIN CAPITAL LETTER T WITH STROKE	LATIN SMALL LETTER T WITH STROKE
0168	0169	LATIN CAPITAL LETTER U WITH TILDE	LATIN SMALL LETTER U WITH TILDE
016A	016B	LATIN CAPITAL LETTER U WITH MACRON	LATIN SMALL LETTER U WITH MACRON
016C	016D	LATIN CAPITAL LETTER U WITH BREVE	LATIN SMALL LETTER U WITH BREVE
016E	016F	LATIN CAPITAL LETTER U WITH RING ABOVE	LATIN SMALL LETTER U WITH RING ABOVE
0170	0171	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE	LATIN SMALL LETTER U WITH DOUBLE ACUTE
0172	0173	LATIN CAPITAL LETTER U WITH OGONEK	LATIN SMALL LETTER U WITH OGONEK
0174	0175	LATIN CAPITAL LETTER W WITH CIRCUMFLEX	LATIN SMALL LETTER W WITH CIRCUMFLEX
0176	0177	LATIN CAPITAL LETTER Y WITH CIRCUMFLEX	LATIN SMALL LETTER Y WITH CIRCUMFLEX
0178	00FF	LATIN CAPITAL LETTER Y WITH DIAERESIS	LATIN SMALL LETTER Y DIAERESIS

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
0179	017A	LATIN CAPITAL LETTER Z WITH ACUTE	LATIN SMALL LETTER Z WITH ACUTE
017B	017C	LATIN CAPITAL LETTER Z WITH DOT ABOVE	LATIN SMALL LETTER Z WITH DOT ABOVE
017D	017E	LATIN CAPITAL LETTER Z WITH CARON	LATIN SMALL LETTER Z WITH CARON
0181	0253	LATIN CAPITAL LETTER B WITH HOOK	LATIN SMALL LETTER B WITH HOOK
0182	0183	LATIN CAPITAL LETTER B WITH TOPBAR	LATIN SMALL LETTER B WITH TOPBAR
0184	0185	LATIN CAPITAL LETTER TONE SIX	LATIN SMALL LETTER TONE SIX
0186	0254	LATIN CAPITAL LETTER OPEN O	LATIN SMALL LETTER OPEN O
0187	0188	LATIN CAPITAL LETTER C WITH HOOK	LATIN SMALL LETTER C WITH HOOK
018A	0257	LATIN CAPITAL LETTER D WITH HOOK	LATIN SMALL LETTER D WITH HOOK
018B	018C	LATIN CAPITAL LETTER D WITH TOPBAR	LATIN SMALL LETTER D WITH TOPBAR
018E	0258	LATIN CAPITAL LETTER REVERSED E	LATIN SMALL LETTER REVERSED E
018F	0259	LATIN CAPITAL LETTER SCHWA	LATIN SMALL LETTER SCHWA
0190	025B	LATIN CAPITAL LETTER OPEN E	LATIN SMALL LETTER OPEN E
0191	0192	LATIN CAPITAL LETTER F WITH HOOK	LATIN SMALL LETTER F WITH HOOK
0193	0260	LATIN CAPITAL LETTER G WITH HOOK	LATIN SMALL LETTER G WITH HOOK
0194	0263	LATIN CAPITAL LETTER GAMMA	LATIN SMALL LETTER GAMMA
0196	0269	LATIN CAPITAL LETTER IOTA	LATIN SMALL LETTER IOTA
0197	0268	LATIN CAPITAL LETTER I WITH STROKE	LATIN SMALL LETTER I WITH STROKE
0198	0199	LATIN CAPITAL LETTER K WITH HOOK	LATIN SMALL LETTER K WITH HOOK
019C	026f	LATIN CAPITAL LETTER TURNED M	LATIN SMALL LETTER TURNED M
019D	0272	LATIN CAPITAL LETTER N WITH LEFT HOOK	LATIN SMALL LETTER N WITH LEFT HOOK
019F	0275	LATIN CAPITAL LETTER O WITH MIDDLE TILDE	LATIN SMALL LETTER BARRED O
01A0	01A1	LATIN CAPITAL LETTER O WITH HORN	LATIN SMALL LETTER O WITH HORN
01A2	01A3	LATIN CAPITAL LETTER OI	LATIN SMALL LETTER OI
01A4	01A5	LATIN CAPITAL LETTER P WITH HOOK	LATIN SMALL LETTER P WITH HOOK
01A7	01A8	LATIN CAPITAL LETTER TONE TWO	LATIN SMALL LETTER TONE TWO
01A9	0283	LATIN CAPITAL LETTER ESH	LATIN SMALL LETTER ESH
01AC	01AD	LATIN CAPITAL LETTER T WITH HOOK	LATIN SMALL LETTER T WITH HOOK
01AE	0288	LATIN CAPITAL LETTER T WITH RETROFLEX HOOK	LATIN SMALL LETTER T WITH RETROFLEX HOOK
01AF	01B0	LATIN CAPITAL LETTER U WITH HORN	LATIN SMALL LETTER U WITH HORN
01B1	028A	LATIN CAPITAL LETTER UPSILON	LATIN SMALL LETTER UPSILON
01B2	028B	LATIN CAPITAL LETTER V WITH HOOK	LATIN SMALL LETTER V WITH HOOK
01B3	01B4	LATIN CAPITAL LETTER Y WITH HOOK	LATIN SMALL LETTER Y WITH HOOK

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
01B5	01B6	LATIN CAPITAL LETTER Z WITH STROKE	LATIN SMALL LETTER Z WITH STROKE
01B7	0292	LATIN CAPITAL LETTER EZH	LATIN SMALL LETTER EZH
01B8	01B9	LATIN CAPITAL LETTER EZH REVERSED	LATIN SMALL LETTER EZH REVERSED
01BC	01BD	LATIN CAPITAL LETTER TONE FIVE	LATIN SMALL LETTER TONE FIVE
01C4	01C6	LATIN CAPITAL LETTER DZ WITH CARON	LATIN SMALL LETTER DZ WITH CARON
01C5	01C6	LATIN CAPITAL LETTER D WITH SMALL LETTER Z WITH CARON	LATIN SMALL LETTER DZ WITH CARON
01C7	01C9	LATIN CAPITAL LETTER LJ	LATIN SMALL LETTER LJ
01C8	01C9	LATIN CAPITAL LETTER L WITH SMALL LETTER J	LATIN SMALL LETTER LJ
01CA	01CC	LATIN CAPITAL LETTER NJ	LATIN SMALL LETTER NJ
01CB	01CC	LATIN CAPITAL LETTER N WITH SMALL LETTER J	LATIN SMALL LETTER NJ
01CD	01CE	LATIN CAPITAL LETTER A WITH CARON	LATIN SMALL LETTER A WITH CARON
01CF	01D0	LATIN CAPITAL LETTER I WITH CARON	LATIN SMALL LETTER I WITH CARON
01D1	01D2	LATIN CAPITAL LETTER O WITH CARON	LATIN SMALL LETTER O WITH CARON
01D3	01D4	LATIN CAPITAL LETTER U WITH CARON	LATIN SMALL LETTER U WITH CARON
01D5	01D6	LATIN CAPITAL LETTER U WITH DIAERESIS AND MACRON	LATIN SMALL LETTER U WITH DIAERESIS AND MACRON
01D7	01D8	LATIN CAPITAL LETTER U WITH DIAERESIS AND ACUTE	LATIN SMALL LETTER U WITH DIAERESIS AND ACUTE
01D9	01DA	LATIN CAPITAL LETTER U WITH DIAERESIS AND CARON	LATIN SMALL LETTER U WITH DIAERESIS AND CARON
01DB	01DC	LATIN CAPITAL LETTER U WITH DIAERESIS AND GRAVE	LATIN SMALL LETTER U WITH DIAERESIS AND GRAVE
01DE	01DF	LATIN CAPITAL LETTER A WITH DIAERESIS AND MACRON	LATIN SMALL LETTER A WITH DIAERESIS AND MACRON
01E0	01E1	LATIN CAPITAL LETTER A WITH DOT ABOVE AND MACRON	LATIN SMALL LETTER A WITH DOT ABOVE AND MACRON
01E2	01E3	LATIN CAPITAL LIGATURE AE WITH MACRON	LATIN SMALL LIGATURE AE WITH MACRON
01E4	01E5	LATIN CAPITAL LETTER G WITH STROKE	LATIN SMALL LETTER G WITH STROKE
01E6	01E7	LATIN CAPITAL LETTER G WITH CARON	LATIN SMALL LETTER G WITH CARON
01E8	01E9	LATIN CAPITAL LETTER K WITH CARON	LATIN SMALL LETTER K WITH CARON
01EA	01EB	LATIN CAPITAL LETTER O WITH OGONEK	LATIN SMALL LETTER O WITH OGONEK
01EC	01ED	LATIN CAPITAL LETTER O WITH OGONEK AND MACRON	LATIN SMALL LETTER O WITH OGONEK AND MACRON
01EE	01EF	LATIN CAPITAL LETTER EZH WITH CARON	LATIN SMALL LETTER EZH WITH CARON
01F1	01F3	LATIN CAPITAL LETTER DZ	LATIN SMALL LETTER DZ
01F4	01F5	LATIN CAPITAL LETTER G WITH ACUTE	LATIN SMALL LETTER G WITH ACUTE
01FA	01FB	LATIN CAPITAL LETTER A WITH RING ABOVE AND ACUTE	LATIN SMALL LETTER A WITH RING ABOVE AND ACUTE
01FC	01FD	LATIN CAPITAL LIGATURE AE WITH ACUTE	LATIN SMALL LIGATURE AE WITH ACUTE
01FE	01FF	LATIN CAPITAL LETTER O WITH STROKE AND ACUTE	LATIN SMALL LETTER O WITH STROKE AND ACUTE

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
0200	0201	LATIN CAPITAL LETTER A WITH DOUBLE GRAVE	LATIN SMALL LETTER A WITH DOUBLE GRAVE
0202	0203	LATIN CAPITAL LETTER A WITH INVERTED BREVE	LATIN SMALL LETTER A WITH INVERTED BREVE
0204	0205	LATIN CAPITAL LETTER E WITH DOUBLE GRAVE	LATIN SMALL LETTER E WITH DOUBLE GRAVE
0206	0207	LATIN CAPITAL LETTER E WITH INVERTED BREVE	LATIN SMALL LETTER E WITH INVERTED BREVE
0208	0209	LATIN CAPITAL LETTER I WITH DOUBLE GRAVE	LATIN SMALL LETTER I WITH DOUBLE GRAVE
020A	020B	LATIN CAPITAL LETTER I WITH INVERTED BREVE	LATIN SMALL LETTER I WITH INVERTED BREVE
020C	020D	LATIN CAPITAL LETTER O WITH DOUBLE GRAVE	LATIN SMALL LETTER O WITH DOUBLE GRAVE
020E	020F	LATIN CAPITAL LETTER O WITH INVERTED BREVE	LATIN SMALL LETTER O WITH INVERTED BREVE
0210	0211	LATIN CAPITAL LETTER R WITH DOUBLE GRAVE	LATIN SMALL LETTER R WITH DOUBLE GRAVE
0212	0213	LATIN CAPITAL LETTER R WITH INVERTED BREVE	LATIN SMALL LETTER R WITH INVERTED BREVE
0214	0215	LATIN CAPITAL LETTER U WITH DOUBLE GRAVE	LATIN SMALL LETTER U WITH DOUBLE GRAVE
0216	0217	LATIN CAPITAL LETTER U WITH INVERTED BREVE	LATIN SMALL LETTER U WITH INVERTED BREVE
0386	03AC	GREEK CAPITAL LETTER ALPHA WITH TONOS	GREEK SMALL LETTER ALPHA WITH TONOS
0388	03AD	GREEK CAPITAL LETTER EPSILON WITH TONOS	GREEK SMALL LETTER EPSILON WITH TONOS
0389	03AE	GREEK CAPITAL LETTER ETA WITH TONOS	GREEK SMALL LETTER ETA WITH TONOS
038A	03AF	GREEK CAPITAL LETTER IOTA WITH TONOS	GREEK SMALL LETTER IOTA WITH TONOS
038C	03CC	GREEK CAPITAL LETTER OMICRON WITH TONOS	GREEK SMALL LETTER OMICRON WITH TONOS
038E	03CD	GREEK CAPITAL LETTER UPSILON WITH TONOS	GREEK SMALL LETTER UPSILON WITH TONOS
038F	03CE	GREEK CAPITAL LETTER OMEGA WITH TONOS	GREEK SMALL LETTER OMEGA WITH TONOS
0391	03B1	GREEK CAPITAL LETTER ALPHA	GREEK SMALL LETTER ALPHA
0392	03B2	GREEK CAPITAL LETTER BETA	GREEK SMALL LETTER BETA
0393	03B3	GREEK CAPITAL LETTER GAMMA	GREEK SMALL LETTER GAMMA
0394	03B4	GREEK CAPITAL LETTER DELTA	GREEK SMALL LETTER DELTA
0395	03B5	GREEK CAPITAL LETTER EPSILON	GREEK SMALL LETTER EPSILON
0396	03B6	GREEK CAPITAL LETTER ZETA	GREEK SMALL LETTER ZETA
0397	03B7	GREEK CAPITAL LETTER ETA	GREEK SMALL LETTER ETA
0398	03B8	GREEK CAPITAL LETTER THETA	GREEK SMALL LETTER THETA
0399	03B9	GREEK CAPITAL LETTER IOTA	GREEK SMALL LETTER IOTA
039A	03BA	GREEK CAPITAL LETTER KAPPA	GREEK SMALL LETTER KAPPA
039B	03BB	GREEK CAPITAL LETTER LAMDA	GREEK SMALL LETTER LAMDA
039C	03BC	GREEK CAPITAL LETTER MU	GREEK SMALL LETTER MU
039D	03BD	GREEK CAPITAL LETTER NU	GREEK SMALL LETTER NU
039E	03BE	GREEK CAPITAL LETTER XI	GREEK SMALL LETTER XI
039F	03BF	GREEK CAPITAL LETTER OMICRON	GREEK SMALL LETTER OMICRON
03A0	03C0	GREEK CAPITAL LETTER PI	GREEK SMALL LETTER PI
03A1	03C1	GREEK CAPITAL LETTER RHO	GREEK SMALL LETTER RHO

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
03A3	03C3	GREEK CAPITAL LETTER SIGMA	GREEK SMALL LETTER SIGMA
03A4	03C4	GREEK CAPITAL LETTER TAU	GREEK SMALL LETTER TAU
03A5	03C5	GREEK CAPITAL LETTER UPSILON	GREEK SMALL LETTER UPSILON
03A6	03C6	GREEK CAPITAL LETTER PHI	GREEK SMALL LETTER PHI
03A7	03C7	GREEK CAPITAL LETTER CHI	GREEK SMALL LETTER CHI
03A8	03C8	GREEK CAPITAL LETTER PSI	GREEK SMALL LETTER PSI
03A9	03C9	GREEK CAPITAL LETTER OMEGA	GREEK SMALL LETTER OMEGA
03AA	03CA	GREEK CAPITAL LETTER IOTA WITH DIALYTIKA	GREEK SMALL LETTER IOTA WITH DIALYTIKA
03AB	03CB	GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA	GREEK SMALL LETTER UPSILON WITH DIALYTIKA
03E2	03E3	COPTIC CAPITAL LETTER SHEI	COPTIC SMALL LETTER SHEI
03E4	03E5	COPTIC CAPITAL LETTER FEI	COPTIC SMALL LETTER FEI
03E6	03E7	COPTIC CAPITAL LETTER KHEI	COPTIC SMALL LETTER KHEI
03E8	03E9	COPTIC CAPITAL LETTER HORI	COPTIC SMALL LETTER HORI
03EA	03EB	COPTIC CAPITAL LETTER GANGIA	COPTIC SMALL LETTER GANGIA
03EC	03ED	COPTIC CAPITAL LETTER SHIMA	COPTIC SMALL LETTER SHIMA
03EE	03EF	COPTIC CAPITAL LETTER DEI	COPTIC SMALL LETTER DEI
0401	0451	CYRILLIC CAPITAL LETTER IO	CYRILLIC SMALL LETTER IO
0402	0452	CYRILLIC CAPITAL LETTER DJE (SERBOCROATIAN)	CYRILLIC SMALL LETTER DJE (SERBOCROATIAN)
0403	0453	CYRILLIC CAPITAL LETTER GJE	CYRILLIC SMALL LETTER GJE
0404	0454	CYRILLIC CAPITAL LETTER UKRAINIAN IE	CYRILLIC SMALL LETTER UKRAINIAN IE
0405	0455	CYRILLIC CAPITAL LETTER DZE	CYRILLIC SMALL LETTER DZE
0406	0456	CYRILLIC CAPITAL LETTER BYELORUSSIAN_UKRAINIAN I	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I
0407	0457	CYRILLIC CAPITAL LETTER YI (UKRAINIAN)	CYRILLIC SMALL LETTER YI (UKRAINIAN)
0408	0458	CYRILLIC CAPITAL LETTER JE	CYRILLIC SMALL LETTER JE
0409	0459	CYRILLIC CAPITAL LETTER LJE	CYRILLIC SMALL LETTER LJE
040A	045A	CYRILLIC CAPITAL LETTER NJE	CYRILLIC SMALL LETTER NJE
040B	045B	CYRILLIC CAPITAL LETTER TSHE (SERBOCROATIAN)	CYRILLIC SMALL LETTER TSHE (SERBOCROATIAN)
040C	045C	CYRILLIC CAPITAL LETTER KJE	CYRILLIC SMALL LETTER KJE
040E	045E	CYRILLIC CAPITAL LETTER SHORT U (BYELORUSSIAN)	CYRILLIC SMALL LETTER SHORT U (BYELORUSSIAN)
040F	045F	CYRILLIC CAPITAL LETTER DZHE	CYRILLIC SMALL LETTER DZHE
0410	0430	CYRILLIC CAPITAL LETTER A	CYRILLIC SMALL LETTER A
0411	0431	CYRILLIC CAPITAL LETTER BE	CYRILLIC SMALL LETTER BE
0412	0432	CYRILLIC CAPITAL LETTER VE	CYRILLIC SMALL LETTER VE
0413	0433	CYRILLIC CAPITAL LETTER GHE	CYRILLIC SMALL LETTER GHE
0414	0434	CYRILLIC CAPITAL LETTER DE	CYRILLIC SMALL LETTER DE
0415	0435	CYRILLIC CAPITAL LETTER IE	CYRILLIC SMALL LETTER IE
0416	0436	CYRILLIC CAPITAL LETTER ZHE	CYRILLIC SMALL LETTER ZHE
0417	0437	CYRILLIC CAPITAL LETTER ZE	CYRILLIC SMALL LETTER ZE
0418	0438	CYRILLIC CAPITAL LETTER I	CYRILLIC SMALL LETTER I
0419	0439	CYRILLIC CAPITAL LETTER SHORT I	CYRILLIC SMALL LETTER SHORT I
041A	043A	CYRILLIC CAPITAL LETTER KA	CYRILLIC SMALL LETTER KA
041B	043B	CYRILLIC CAPITAL LETTER EL	CYRILLIC SMALL LETTER EL
041C	043C	CYRILLIC CAPITAL LETTER EM	CYRILLIC SMALL LETTER EM

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
041D	043D	CYRILLIC CAPITAL LETTER EN	CYRILLIC SMALL LETTER EN
041E	043E	CYRILLIC CAPITAL LETTER O	CYRILLIC SMALL LETTER O
041F	043F	CYRILLIC CAPITAL LETTER PE	CYRILLIC SMALL LETTER PE
0420	0440	CYRILLIC CAPITAL LETTER ER	CYRILLIC SMALL LETTER ER
0421	0441	CYRILLIC CAPITAL LETTER ES	CYRILLIC SMALL LETTER ES
0422	0442	CYRILLIC CAPITAL LETTER TE	CYRILLIC SMALL LETTER TE
0423	0443	CYRILLIC CAPITAL LETTER U	CYRILLIC SMALL LETTER U
0424	0444	CYRILLIC CAPITAL LETTER EF	CYRILLIC SMALL LETTER EF
0425	0445	CYRILLIC CAPITAL LETTER HA	CYRILLIC SMALL LETTER HA
0426	0446	CYRILLIC CAPITAL LETTER TSE	CYRILLIC SMALL LETTER TSE
0427	0447	CYRILLIC CAPITAL LETTER CHE	CYRILLIC SMALL LETTER CHE
0428	0448	CYRILLIC CAPITAL LETTER SHA	CYRILLIC SMALL LETTER SHA
0429	0449	CYRILLIC CAPITAL LETTER SHCHA	CYRILLIC SMALL LETTER SHCHA
042A	044A	CYRILLIC CAPITAL LETTER HARD SIGN	CYRILLIC SMALL LETTER HARD SIGN
042B	044B	CYRILLIC CAPITAL LETTER YERU	CYRILLIC SMALL LETTER YERU
042C	044C	CYRILLIC CAPITAL LETTER SOFT SIGN	CYRILLIC SMALL LETTER SOFT SIGN
042D	044D	CYRILLIC CAPITAL LETTER E	CYRILLIC SMALL LETTER E
042E	044E	CYRILLIC CAPITAL LETTER YU	CYRILLIC SMALL LETTER YU
042F	044F	CYRILLIC CAPITAL LETTER YA	CYRILLIC SMALL LETTER YA
0460	0461	CYRILLIC CAPITAL LETTER OMEGA	CYRILLIC SMALL LETTER OMEGA
0462	0463	CYRILLIC CAPITAL LETTER YAT	CYRILLIC SMALL LETTER YAT
0464	0465	CYRILLIC CAPITAL LETTER IOTIFIED E	CYRILLIC SMALL LETTER IOTIFIED E
0466	0467	CYRILLIC CAPITAL LETTER LITTLE YUS	CYRILLIC SMALL LETTER LITTLE YUS
0468	0469	CYRILLIC CAPITAL LETTER IOTIFIED LITTLE YUS	CYRILLIC SMALL LETTER IOTIFIED LITTLE YUS
046A	046B	CYRILLIC CAPITAL LETTER BIG YUS	CYRILLIC SMALL LETTER BIG YUS
046C	046D	CYRILLIC CAPITAL LETTER IOTIFIED BIG YUS	CYRILLIC SMALL LETTER IOTIFIED BIG YUS
046E	046F	CYRILLIC CAPITAL LETTER KSI	CYRILLIC SMALL LETTER KSI
0470	0471	CYRILLIC CAPITAL LETTER PSI	CYRILLIC SMALL LETTER PSI
0472	0473	CYRILLIC CAPITAL LETTER FITA	CYRILLIC SMALL LETTER FITA
0474	0475	CYRILLIC CAPITAL LETTER IZHITSA	CYRILLIC SMALL LETTER IZHITSA
0476	0477	CYRILLIC CAPITAL LETTER IZHITSA WITH DOUBLE GRAVE ACCENT	CYRILLIC SMALL LETTER IZHITSA WITH DOUBLE GRAVE ACCENT
0478	0479	CYRILLIC CAPITAL LETTER UK	CYRILLIC SMALL LETTER UK
047A	047B	CYRILLIC CAPITAL LETTER ROUND OMEGA	CYRILLIC SMALL LETTER ROUND OMEGA
047C	047D	CYRILLIC CAPITAL LETTER OMEGA WITH TITLO	CYRILLIC SMALL LETTER OMEGA WITH TITLO
047E	047F	CYRILLIC CAPITAL LETTER OT	CYRILLIC SMALL LETTER OT
0480	0481	CYRILLIC CAPITAL LETTER KOPPA	CYRILLIC SMALL LETTER KOPPA
0490	0491	CYRILLIC CAPITAL LETTER GHE WITH UPTURN	CYRILLIC SMALL LETTER GHE WITH UPTURN
0492	0493	CYRILLIC CAPITAL LETTER GHE WITH STROKE	CYRILLIC SMALL LETTER GHE WITH STROKE

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
0494	0495	CYRILLIC CAPITAL LETTER GHE WITH MIDDLE HOOK	CYRILLIC SMALL LETTER GHE WITH MIDDLE HOOK
0496	0497	CYRILLIC CAPITAL LETTER ZHE WITH DESCENDER	CYRILLIC SMALL LETTER ZHE WITH DESCENDER
0498	0499	CYRILLIC CAPITAL LETTER ZE WITH DESCENDER	CYRILLIC SMALL LETTER ZE WITH DESCENDER
049A	049B	CYRILLIC CAPITAL LETTER KA WITH DESCENDER	CYRILLIC SMALL LETTER KA WITH DESCENDER
049C	049D	CYRILLIC CAPITAL LETTER KA WITH VERTICAL STROKE	CYRILLIC SMALL LETTER KA WITH VERTICAL STROKE
049E	049F	CYRILLIC CAPITAL LETTER KA WITH STROKE	CYRILLIC SMALL LETTER KA WITH STROKE
04A0	04A1	CYRILLIC CAPITAL LETTER BASHKIR KA	CYRILLIC SMALL LETTER EASHKIR KA
04A2	04A3	CYRILLIC CAPITAL LETTER EN WITH DESCENDER	CYRILLIC SMALL LETTER EN WITH DESCENDER
04A4	04A5	CYRILLIC CAPITAL LIGATURE EN GHF	CYRILLIC SMALL LIGATURE EN GHE
04A6	04A7	CYRILLIC CAPITAL LETTER PE WITH MIDDLE HOOK (ABKHASIAN)	CYRILLIC SMALL LETTER PE WITH MIDDLE HOOK (ABKHASIAN)
04A8	04A9	CYRILLIC CAPITAL LETTER ABKHASIAN HA	CYRILLIC SMALL LETTER ABKHASIAN HA
04AA	04AB	CYRILLIC CAPITAL LETTER ES WITH DESCENDER	CYRILLIC SMALL LETTER ES WITH DESCENDER
04AC	04AD	CYRILLIC CAPITAL LETTER TE WITH DESCENDER	CYRILLIC SMALL LETTER TE WITH DESCENDER
04AE	04AF	CYRILLIC CAPITAL LETTER STRAIGHT U	CYRILLIC SMALL LETTER STRAIGHT U
04B0	04B1	CYRILLIC CAPITAL LETTER STRAIGHT U WITH STROKE	CYRILLIC SMALL LETTER STRAIGHT U WITH STROKE
04B2	04B3	CYRILLIC CAPITAL LETTER HA WITH DESCENDER	CYRILLIC SMALL LETTER HA WITH DESCENDER
04B4	04B5	CYRILLIC CAPITAL LIGATURE TE TSE (ABKHASIAN)	CYRILLIC SMALL LIGATURE TE TSE (ABKHASIAN)
04B6	04B7	CYRILLIC CAPITAL LETTER CHE WITH DESCENDER	CYRILLIC SMALL LETTER CHE WITH DESCENDER
04B8	04B9	CYRILLIC CAPITAL LETTER CHE WITH VERTICAL STROKE	CYRILLIC SMALL LETTER CHE WITH VERTICAL STROKE
04BA	04BB	CYRILLIC CAPITAL LETTER SHHA	CYRILLIC SMALL LETTER SHHA
04BC	04BD	CYRILLIC CAPITAL LETTER ABKHASIAN CHE	CYRILLIC SMALL LETTER ABKHASIAN CHE
04BE	04BF	CYRILLIC CAPITAL LETTER ABKHASIAN CHE WITH DESCENDER	CYRILLIC SMALL LETTER ABKHASIAN CHE WITH DESCENDER
04C1	04C2	CYRILLIC CAPITAL LETTER ZHE WITH BREVE	CYRILLIC SMALL LETTER ZHE WITH BREVE
04C3	04C4	CYRILLIC CAPITAL LETTER KA WITH HOOK	CYRILLIC SMALL LETTER KA WITH HOOK
04C7	04C8	CYRILLIC CAPITAL LETTER EN WITH HOOK	CYRILLIC SMALL LETTER EN WITH HOOK
04CB	04CC	CYRILLIC CAPITAL LETTER KHAKASSIAN CHE	CYRILLIC SMALL LETTER KHAKASSIAN CHE
04D0	04D1	CYRILLIC CAPITAL LETTER A WITH BREVE	CYRILLIC SMALL LETTER A WITH BREVE
04D2	04D3	CYRILLIC CAPITAL LETTER A WITH DIAERESIS	CYRILLIC SMALL LETTER A WITH DIAERESIS
04D4	04D5	CYRILLIC CAPITAL LIGATURE A IE	CYRILLIC SMALL LIGATURE A IE
04D6	04D7	CYRILLIC CAPITAL LETTER IE WITH BREVE	CYRILLIC SMALL LETTER IE WITH BREVE

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
04D8	04D9	CYRILLIC CAPITAL LETTER SCHWA	CYRILLIC SMALL LETTER SCHWA
04DA	04DB	CYRILLIC CAPITAL LETTER SCHWA WITH DIAERESIS	CYRILLIC SMALL LETTER SCHWA WITH DIAERESIS
04DC	04DD	CYRILLIC CAPITAL LETTER ZHE WITH DIAERESIS	CYRILLIC SMALL LETTER ZHE WITH DIAERESIS
04DE	04DF	CYRILLIC CAPITAL LETTER ZE WITH DIAERESIS	CYRILLIC SMALL LETTER ZE WITH DIAERESIS
04E0	04E1	CYRILLIC CAPITAL LETTER ABKHASIAN DZE	CYRILLIC SMALL LETTER ABKHASIAN DZE
04E2	04E3	CYRILLIC CAPITAL LETTER I WITH MACRON	CYRILLIC SMALL LETTER I WITH MACRON
04E4	04E5	CYRILLIC CAPITAL LETTER I WITH DIAERESIS	CYRILLIC SMALL LETTER I WITH DIAERESIS
04E6	04E7	CYRILLIC CAPITAL LETTER O WITH DIAERESIS	CYRILLIC SMALL LETTER O WITH DIAERESIS
04E8	04E9	CYRILLIC CAPITAL LETTER BARRED O	CYRILLIC SMALL LETTER BARRED O
04EA	04EB	CYRILLIC CAPITAL LETTER BARRED O WITH DIAERESIS	CYRILLIC SMALL LETTER BARRED 0 WITH DIAERESIS
04EE	04EF	CYRILLIC CAPITAL LETTER U WITH MACRON	CYRILLIC SMALL LETTER U WITH MACRON
04F0	04F1	CYRILLIC CAPITAL LETTER U WITH DIAERESIS	CYRILLIC SMALL LETTER U WITH DIAERESIS
04F2	04F3	CYRILLIC CAPITAL LETTER U WITH DOUBLE ACUTE	CYRILLIC SMALL LETTER U WITH DOUBLE ACUTE
04F4	04F5	CYRILLIC CAPITAL LETTER CHE WITH DIAERESIS	CYRILLIC SMALL LETTER CHE WITH DIAERESIS
04F8	04F9	CYRILLIC CAPITAL LETTER YERU WITH DIAERESIS	CYRILLIC SMALL LETTER YERU WITH DIAERESIS
0531	0561	ARMENIAN CAPITAL LETTER AYB	ARMENIAN SMALL LETTER AYB
0532	0562	ARMENIAN CAPITAL LETTER BEN	ARMENIAN SMALL LETTER BEN
0533	0563	ARMENIAN CAPITAL LETTER GIM	ARMENIAN SMALL LETTER GIM
0534	0564	ARMENIAN CAPITAL LETTER DA	ARMENIAN SMALL LETTER DA
0535	0565	ARMENIAN CAPITAL LETTER ECH	ARMENIAN SMALL LETTER ECH
0536	0566	ARMENIAN CAPITAL LETTER ZA	ARMENIAN SMALL LETTER ZA
0537	0567	ARMENIAN CAPITAL LETTER EH	ARMENIAN SMALL LETTER EH
0538	0568	ARMENIAN CAPITAL LETTER ET	ARMENIAN SMALL LETTER ET
0539	0569	ARMENIAN CAPITAL LETTER TO	ARMENIAN SMALL LETTER TO
053A	056A	ARMENIAN CAPITAL LETTER ZHE	ARMENIAN SMALL LETTER ZHE
053B	056B	ARMENIAN CAPITAL LETTER INI	ARMENIAN SMALL LETTER INI
053C	056C	ARMENIAN CAPITAL LETTER LIWN	ARMENIAN SMALL LETTER LIWN
053D	056D	ARMENIAN CAPITAL LETTER XEH	ARMENIAN SMALL LETTER XEH
053E	056E	ARMENIAN CAPITAL LETTER CA	ARMENIAN SMALL LETTER CA
053F	056F	ARMENIAN CAPITAL LETTER KEN	ARMENIAN SMALL LETTER KEN
0540	0570	ARMENIAN CAPITAL LETTER HO	ARMENIAN SMALL LETTER HO
0541	0571	ARMENIAN CAPITAL LETTER JA	ARMENIAN SMALL LETTER JA
0542	0572	ARMENIAN CAPITAL LETTER GHAD	ARMENIAN SMALL LETTER GHAD

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
0543	0573	ARMENIAN CAPITAL LETTER CHEH	ARMENIAN SMALL LETTER CHEH
0544	0574	ARMENIAN CAPITAL LETTER MEN	ARMENIAN SMALL LETTER MEN
0545	0575	ARMENIAN CAPITAL LETTER YI	ARMENIAN SMALL LETTER YI
0546	0576	ARMENIAN CAPITAL LETTER NOW	ARMENIAN SMALL LETTER NOW
0547	0577	ARMENIAN CAPITAL LETTER SHA	ARMENIAN SMALL LETTER SNA
0548	0578	ARMENIAN CAPITAL LETTER VO	ARMENIAN SMALL LETTER VO
0549	0579	ARMENIAN CAPITAL LETTER CHA	ARMENIAN SMALL LETTER CHA
054A	057A	ARMENIAN CAPITAL LETTER PEH	ARMENIAN SMALL LETTER PEH
054B	057B	ARMENIAN CAPITAL LETTER JHEH	ARMENIAN SMALL LETTER JHEH
054C	057C	ARMENIAN CAPITAL LETTER RA	ARMENIAN SMALL LETTER RA
054D	057D	ARMENIAN CAPITAL LETTER SEH	ARMENIAN SMALL LETTER SEH
054E	057E	ARMENIAN CAPITAL LETTER VEW	ARMENIAN SMALL LETTER VEW
054F	057F	ARMENIAN CAPITAL LETTER TIWN	ARMENIAN SMALL LETTER TIWN
0550	0580	ARMENIAN CAPITAL LETTER REH	ARMENIAN SMALL LETTER REH
0551	0581	ARMENIAN CAPITAL LETTER CO	ARMENIAN SMALL LETTER CO
0552	0582	ARMENIAN CAPITAL LETTER YIWN	ARMENIAN SMALL LETTER YIWN
0553	0583	ARMENIAN CAPITAL LETTER PIWR	ARMENIAN SMALL LETTER PIWP
0554	0584	ARMENIAN CAPITAL LETTER KEH	ARMENIAN SMALL LETTER KEH
0555	0585	ARMENIAN CAPITAL LETTER OH	ARMENIAN SMALL LETTER OH
0556	0586	ARMENIAN CAPITAL LETTER FEH	ARMENIAN SMALL LETTER FEH
10A0	10D0	GEORGIAN CAPITAL LETTER AN (KHUTSURI)	GEORGIAN LETTER AN
10A1	10D1	GEORGIAN CAPITAL LETTER BAN (KHUTSURI)	GEORGIAN LETTER BAN
10A2	10D2	GEORGIAN CAPITAL LETTER GAN (KHUTSURI)	GEORGIAN LETTER GAN
10A3	10D3	GEORGIAN CAPITAL LETTER DON (KHUTSURI)	GEORGIAN LETTER DON
10A4	10D4	GEORGIAN CAPITAL LETTER EN (KHUTSURI)	GEORGIAN LETTER EN
10A5	10D5	GEORGIAN CAPITAL LETTER VIN (KHUTSURI)	GEORGIAN LETTER VIN
10A6	10D6	GEORGIAN CAPITAL LETTER ZEN (KHUTSURI)	GEORGIAN LETTER ZEN
10A7	10D7	GEORGIAN CAPITAL LETTER TAN (KHUTSURI)	GEORGIAN LETTER TAN
10A8	10D8	GEORGIAN CAPITAL LETTER IN (KHUTSURI)	GEORGIAN LETTER IN
10A9	10D9	GEORGIAN CAPITAL LETTER KAN (KHUTSURI)	GEORGIAN LETTER KAN
10AA	10DA	GEORGIAN CAPITAL LETTER LAS (KHUTSURI)	GEORGIAN LETTER LAS
10AB	10DB	GEORGIAN CAPITAL LETTER MAN (KHUTSURI)	GEORGIAN LETTER MAN

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
10AC	10DC	GEORGIAN CAPITAL LETTER NAR (KHUTSURI)	GEORGIAN LETTER NAR
10AD	10DD	GEORGIAN CAPITAL LETTER ON (KHUTSURI)	GEORGIAN LETTER ON
10AE	10DE	GEORGIAN CAPITAL LETTER PAR (KHUTSURI)	GEORGIAN LETTER PAR
10AF	10DF	GEORGIAN CAPITAL LETTER ZHAR (KHUTSURI)	GEORGIAN LETTER ZHAR
10B0	10E0	GEORGIAN CAPITAL LETTER RAE (KHUTSURI)	GEORGIAN LETTER RAE
10B1	10E1	GEORGIAN CAPITAL LETTER SAN (KHUTSURI)	GEORGIAN LETTER SAN
10B2	10E2	GEORGIAN CAPITAL LETTER TAR (KHUTSURI)	GEORGIAN LETTER TAR
10B3	10E3	GEORGIAN CAPITAL LETTER UN (KHUTSURI)	GEORGIAN LETTER UN
10B4	10E4	GEORGIAN CAPITAL LETTER PHAR (KHUTSURI)	GEORGIAN LETTER PHAR
10B5	10E5	GEORGIAN CAPITAL LETTER KHAR (KHUTSURI)	GEORGIAN LETTER KHAR
10B6	10E6	GEORGIAN CAPITAL LETTER GHAN (KHUTSURI)	GEORGIAN LETTER GHAN
10B7	10E7	GEORGIAN CAPITAL LETTER QAR (KHUTSURI)	GEORGIAN LETTER QAR
10B8	10E8	GEORGIAN CAPITAL LETTER SHIN (KHUTSURI)	GEORGIAN LETTER SHIN
10B9	10E9	GEORGIAN CAPITAL LETTER CHIN (KHUTSURI)	GEORGIAN LETTER CHIN
10BA	10EA	GEORGIAN CAPITAL LETTER CAN (KHUTSURI)	GEORGIAN LETTER CAN
10BB	10EB	GEORGIAN CAPITAL LETTER JIL (KHUTSURI)	GEORGIAN LETTER JIL
10BC	10EC	GEORGIAN CAPITAL LETTER CIL (KHUTSURI)	GEORGIAN LETTER CIL
10BD	10ED	GEORGIAN CAPITAL LETTER CHAR (KHUTSURI)	GEORGIAN LETTER CHAR
10BE	10EE	GEORGIAN CAPITAL LETTER XAN (KHUTSURI)	GEORGIAN LETTER XAN
10BF	10EF	GEORGIAN CAPITAL LETTER JHAN (KHUTSURI)	GEORGIAN LETTER JHAN
10C0	10F0	GEORGIAN CAPITAL LETTER HAE (KHUTSURI)	GEORGIAN LETTER HAE
10C1	10F1	GEORGIAN CAPITAL LETTER HE (KHUTSURI)	GEORGIAN LETTER HE
10C2	10F2	GEORGIAN CAPITAL LETTER HIE (KHUTSURI)	GEORGIAN LETTER HIE
10C3	10F3	GEORGIAN CAPITAL LETTER WE (KHUTSURI)	GEORGIAN LETTER WE
10C4	10F4	GEORGIAN CAPITAL LETTER HAR (KHUTSURI)	GEORGIAN LETTER HAR
10C5	10F5	GEORGIAN CAPITAL LETTER HOE (KHUTSURI)	GEORGIAN LETTER HOE
1E00	1E01	LATIN CAPITAL LETTER A WITH RING BELOW	LATIN SMALL LETTER A WITH RING BELOW
1E02	1E03	LATIN CAPITAL LETTER B WITH DOT ABOVE	LATIN SMALL LETTER B WITH DOT ABOVE
1E04	1E05	LATIN CAPITAL LETTER B WITH DOT BELOW	LATIN SMALL LETTER B WITH DOT BELOW
1E06	1E07	LATIN CAPITAL LETTER B WITH LINE BELOW	LATIN SMALL LETTER B WITH LINE BELOW

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1E08	1E09	LATIN CAPITAL LETTER C WITH CEDILLA AND ACUTE	LATIN SMALL LETTER C WITH CEDILLA AND ACUTE
1E0A	1E0B	LATIN CAPITAL LETTER D WITH DOT ABOVE	LATIN SMALL LETTER D WITH DOT ABOVE
1E0C	1E0D	LATIN CAPITAL LETTER D WITH DOT BELOW	LATIN SMALL LETTER D WITH DOT BELOW
1E0E	1E0F	LATIN CAPITAL LETTER D WITH LINE BELOW	LATIN SMALL LETTER D WITH LINE BELOW
1E10	1E11	LATIN CAPITAL LETTER D WITH CEDILLA	LATIN SMALL LETTER D WITH CEDILLA
1E12	1E13	LATIN CAPITAL LETTER D WITH CIRCUMFLEX BELOW	LATIN SMALL LETTER D WITH CIRCUMFLEX BELOW
1E14	1E15	LATIN CAPITAL LETTER E WITH MACRON AND GRAVE	LATIN SMALL LETTER E WITH MACRON AND GRAVE
1E16	1E17	LATIN CAPITAL LETTER E WITH MACRON AND ACUTE	LATIN SMALL LETTER E WITH MACRON AND ACUTE
1E18	1E19	LATIN CAPITAL LETTER E WITH CIRCUMFLEX BELOW	LATIN SMALL LETTER E WITH CIRCUMFLEX BELOW
1E1A	1E1B	LATIN CAPITAL LETTER E WITH TILDE BELOW	LATIN SMALL LETTER E WITH TILDE BELOW
1E1C	1E1D	LATIN CAPITAL LETTER E WITH CEDILLA AND BREVE	LATIN SMALL LETTER E WITH CEDILLA AND BREVE
1E1E	1E1F	LATIN CAPITAL LETTER F WITH DOT ABOVE	LATIN SMALL LETTER F WITH DOT ABOVE
1E20	1E21	LATIN CAPITAL LETTER G WITH MACRON	LATIN SMALL LETTER G WITH MACRON
1E22	1E23	LATIN CAPITAL LETTER H WITH DOT ABOVE	LATIN SMALL LETTER H WITH DOT ABOVE
1E24	1E25	LATIN CAPITAL LETTER H WITH DOT BELOW	LATIN SMALL LETTER H WITH DOT BELOW
1E26	1E27	LATIN CAPITAL LETTER H WITH DIAERESIS	LATIN SMALL LETTER H WITH DIAERESIS
1E28	1E29	LATIN CAPITAL LETTER H WITH CEDILLA	LATIN SMALL LETTER H WITH CEDILLA
1E2A	1E2B	LATIN CAPITAL LETTER H WITH BREVE BELOW	LATIN SMALL LETTER H WITH BREVE BELOW
1E2C	1E2D	LATIN CAPITAL LETTER I WITH TILDE BELOW	LATIN SMALL LETTER I WITH TILDE BELOW
1E2E	1E2F	LATIN CAPITAL LETTER I WITH DIAERESIS AND ACUTE	LATIN SMALL LETTER I WITH DIAERESIS AND ACUTE
1E30	1E31	LATIN CAPITAL LETTER K WITH ACUTE	LATIN SMALL LETTER K WITH ACUTE
1E32	1E33	LATIN CAPITAL LETTER K WITH DOT BELOW	LATIN SMALL LETTER K WITH DOT BELOW
1E34	1E35	LATIN CAPITAL LETTER K WITH LINE BELOW	LATIN SMALL LETTER K WITH LINE BELOW
1E36	1E37	LATIN CAPITAL LETTER L WITH DOT BELOW	LATIN SMALL LETTER L WITH DOT BELOW
1E38	1E39	LATIN CAPITAL LETTER L WITH DOT BELOW AND MACRON	LATIN SMALL LETTER L WITH DOT BELOW AND MACRON
1E3A	1E3B	LATIN CAPITAL LETTER L WITH LINE BELOW	LATIN SMALL LETTER L WITH LINE BELOW
1E3C	1E3D	LATIN CAPITAL LETTER L WITH CIRCUMFLEX BELOW	LATIN SMALL LETTER L WITH CIRCUMFLEX BELOW
1E3E	1E3F	LATIN CAPITAL LETTER M WITH ACUTE	LATIN SMALL LETTER M WITH ACUTE
1E40	1E41	LATIN CAPITAL LETTER M WITH DOT ABOVE	LATIN SMALL LETTER M WITH DOT ABOVE
1E42	1E43	LATIN CAPITAL LETTER M WITH DOT BELOW	LATIN SMALL LETTER M WITH DOT BELOW

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1E44	1E45	LATIN CAPITAL LETTER N WITH DOT ABOVE	LATIN SMALL LETTER N WITH DOT ABOVE
1E46	1E47	LATIN CAPITAL LETTER N WITH DOT BELOW	LATIN SMALL LETTER N WITH DOT BELOW
1E48	1E49	LATIN CAPITAL LETTER N WITH LINE BELOW	LATIN SMALL LETTER N WITH LINE BELOW
1E4A	1E4B	LATIN CAPITAL LETTER N WITH CIRCUMFLEX BELOW	LATIN SMALL LETTER N WITH CIRCUMFLEX BELOW
1E4C	1E4D	LATIN CAPITAL LETTER O WITH TILDE AND ACUTE	LATIN SMALL LETTER O WITH TILDE AND ACUTE
1E4E	1E4F	LATIN CAPITAL LETTER O WITH TILDE AND DIAERESIS	LATIN SMALL LETTER O WITH TILDE AND DIAERESIS
1E50	1E51	LATIN CAPITAL LETTER O WITH MACRON AND GRAVE	LATIN SMALL LETTER O WITH MACRON AND GRAVE
1E52	1E53	LATIN CAPITAL LETTER O WITH MACRON AND ACUTE	LATIN SMALL LETTER O WITH MACRON AND ACUTE
1E54	1E55	LATIN CAPITAL LETTER P WITH ACUTE	LATIN SMALL LETTER P WITH ACUTE
1E56	1E57	LATIN CAPITAL LETTER P WITH DOT ABOVE	LATIN SMALL LETTER P WITH DOT ABOVE
1E58	1E59	LATIN CAPITAL LETTER R WITH DOT ABOVE	LATIN SMALL LETTER R WITH DOT ABOVE
1E5A	1E5B	LATIN CAPITAL LETTER R WITH DOT BELOW	LATIN SMALL LETTER R WITH DOT BELOW
1E5C	1E5D	LATIN CAPITAL LETTER R WITH DOT BELOW AND MACRON	LATIN SMALL LETTER R WITH DOT BELOW AND MACRON
1E5E	1E5F	LATIN CAPITAL LETTER R WITH LINE BELOW	LATIN SMALL LETTER R WITH LINE BELOW
1E60	1E61	LATIN CAPITAL LETTER S WITH DOT ABOVE	LATIN SMALL LETTER S WITH DOT ABOVE
1E62	1E63	LATIN CAPITAL LETTER S WITH DOT BELOW	LATIN SMALL LETTER S WITH DOT BELOW
1E64	1E65	LATIN CAPITAL LETTER S WITH ACUTE AND DOT ABOVE	LATIN SMALL LETTER S WITH ACUTE AND DOT ABOVE
1E66	1E67	LATIN CAPITAL LETTER S WITH CARON AND DOT ABOVE	LATIN SMALL LETTER S WITH CARON AND DOT ABOVE
1E68	1E69	LATIN CAPITAL LETTER S WITH DOT BELOW AND DOT ABOVE	LATIN SMALL LETTER S WITH DOT BELOW AND DOT ABOVE
1E6A	1E6B	LATIN CAPITAL LETTER T WITH DOT ABOVE	LATIN SMALL LETTER T WITH DOT ABOVE
1E6C	1E6D	LATIN CAPITAL LETTER T WITH DOT BELOW	LATIN SMALL LETTER T WITH DOT BELOW
1E6E	1E6F	LATIN CAPITAL LETTER T WITH LINE BELOW	LATIN SMALL LETTER T WITH LINE BELOW
1E70	1E71	LATIN CAPITAL LETTER T WITH CIRCUMFLEX BELOW	LATIN SMALL LETTER T WITH CIRCUMFLEX BELOW
1E72	1E73	LATIN CAPITAL LETTER U WITH DIAERESIS BELOW	LATIN SMALL LETTER U WITH DIAERESIS BELOW
1E74	1E75	LATIN CAPITAL LETTER U WITH TILDE BELOW	LATIN SMALL LETTER U WITH TILDE BELOW
1E76	1E77	LATIN CAPITAL LETTER U WITH CIRCUMFLEX BELOW	LATIN SMALL LETTER U WITH CIRCUMFLEX BELOW
1E78	1E79	LATIN CAPITAL LETTER U WITH TILDE AND ACUTE	LATIN SMALL LETTER U WITH TILDE AND ACUTE
1E7A	1E7B	LATIN CAPITAL LETTER U WITH MACRON AND DIAERESIS	LATIN SMALL LETTER U WITH MACRON AND DIAERESIS
1E7C	1E7D	LATIN CAPITAL LETTER V WITH TILDE	LATIN SMALL LETTER V WITH TILDE
1E7E	1E7F	LATIN CAPITAL LETTER V WITH DOT BELOW	LATIN SMALL LETTER V WITH DOT BELOW

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1E80	1E81	LATIN CAPITAL LETTER W WITH GRAVE	LATIN SMALL LETTER W WITH GRAVE
1E82	1E83	LATIN CAPITAL LETTER W WITH ACUTE	LATIN SMALL LETTER W WITH ACUTE
1E84	1E85	LATIN CAPITAL LETTER W WITH DIAERESIS	LATIN SMALL LETTER W WITH DIAERESIS
1E86	1E87	LATIN CAPITAL LETTER W WITH DOT ABOVE	LATIN SMALL LETTER W WITH DOT ABOVE
1E88	1E89	LATIN CAPITAL LETTER W WITH DOT BELOW	LATIN SMALL LETTER W WITH DOT BELOW
1E8A	1E8B	LATIN CAPITAL LETTER X WITH DOT ABOVE	LATIN SMALL LETTER X WITH DOT ABOVE
1E8C	1E8D	LATIN CAPITAL LETTER X5 WITH DIAERESIS	LATIN SMALL LETTER X WITH DIAERESIS
1E8E	1E8F	LATIN CAPITAL LETTER Y WITH DOT ABOVE	LATIN SMALL LETTER Y WITH DOT ABOVE
1E90	1E91	LATIN CAPITAL LETTER Z WITH CIRCUMFLEX	LATIN SMALL LETTER Z WITH CIRCUMFLEX
1E92	1E93	LATIN CAPITAL LETTER Z WITH DOT BELOW	LATIN SMALL LETTER Z WITH DOT BELOW
1E94	1E95	LATIN CAPITAL LETTER Z WITH LINE BELOW	LATIN SMALL LETTER Z WITH LINE BELOW
1EA0	1EA1	LATIN CAPITAL LETTER A WITH DOT BELOW	LATIN SMALL LETTER A WITH DOT BELOW
1EA2	1EA3	LATIN CAPITAL LETTER A WITH HOOK ABOVE	LATIN SMALL LETTER A WITH HOOK ABOVE
1EA4	1EA5	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND ACUTE	LATIN SMALL LETTER A WITH CIRCUMFLEX AND ACUTE
1EA6	1EA7	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND GRAVE	LATIN SMALL LETTER A WITH CIRCUMFLEX AND GRAVE
1EA8	1EA9	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE	LATIN SMALL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE
1EAA	1EAB	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND TILDE	LATIN SMALL LETTER A WITH CIRCUMFLEX AND TILDE
1EAC	1EAD	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND DOT BELOW	LATIN SMALL LETTER A WITH CIRCUMFLEX AND DOT BELOW
1EAE	1EAF	LATIN CAPITAL LETTER A WITH BREVE AND ACUTE	LATIN SMALL LETTER A WITH BREVE AND ACUTE
1EB0	1EB1	LATIN CAPITAL LETTER A WITH BREVE AND GRAVE	LATIN SMALL LETTER A WITH BREVE AND GRAVE
1EB2	1EB3	LATIN CAPITAL LETTER A WITH BREVE AND HOOK ABOVE	LATIN SMALL LETTER A WITH BREVE AND HOOK ABOVE
1EB4	1EB5	LATIN CAPITAL LETTER A WITH BREVE AND TILDE	LATIN SMALL LETTER A WITH BREVE AND TILDE
1EB6	1EB7	LATIN CAPITAL LETTER A WITH BREVE AND DOT BELOW	LATIN SMALL LETTER A WITH BREVE AND DOT BELOW
1EB8	1EB9	LATIN CAPITAL LETTER E WITH DOT BELOW	LATIN SMALL LETTER E WITH DOT BELOW
1EBA	1EBB	LATIN CAPITAL LETTER E WITH HOOK ABOVE	LATIN SMALL LETTER E WITH HOOK ABOVE
1EBC	1EBD	LATIN CAPITAL LETTER E WITH TILDE	LATIN SMALL LETTER E WITH TILDE
1EBE	1EBF	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND ACUTE	LATIN SMALL LETTER E WITH CIRCUMFLEX AND ACUTE
1EC0	1EC1	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND GRAVE	LATIN SMALL LETTER E WITH CIRCUMFLEX AND GRAVE
1EC2	1EC3	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE	LATIN SMALL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
1EC4	1EC5	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND TILDE	LATIN SMALL LETTER E WITH CIRCUMFLEX AND TILDE

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1EC6	1EC7	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND DOT BELOW	LATIN SMALL LETTER E WITH CIRCUMFLEX AND DOT BELOW
1EC8	1EC9	LATIN CAPITAL LETTER I WITH HOOK ABOVE	LATIN SMALL LETTER I WITH HOOK ABOVE
1ECA	1ECB	LATIN CAPITAL LETTER I WITH DOT BELOW	LATIN SMALL LETTER I WITH DOT BELOW
1ECC	1ECD	LATIN CAPITAL LETTER O WITH DOT BELOW	LATIN SMALL LETTER O WITH DOT BELOW
1ECE	1ECF	LATIN CAPITAL LETTER O WITH HOOK ABOVE	LATIN SMALL LETTER O WITH HOOK ABOVE
1ED0	1ED1	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND ACUTE	LATIN SMALL LETTER O WITH CIRCUMFLEX AND ACUTE
1ED2	1ED3	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND GRAVE	LATIN SMALL LETTER O WITH CIRCUMFLEX AND GRAVE
1ED4	1ED5	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE	LATIN SMALL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE
1ED6	1ED7	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND TILDE	LATIN SMALL LETTER O WITH CIRCUMFLEX AND TILDE
1ED8	1ED9	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND DOT BELOW	LATIN SMALL LETTER O WITH CIRCUMFLEX AND DOT BELOW
1EDA	1EDB	LATIN CAPITAL LETTER O WITH HORN AND ACUTE	LATIN SMALL LETTER O WITH HORN AND ACUTE
1EDC	1EDD	LATIN CAPITAL LETTER O WITH HORN AND GRAVE	LATIN SMALL LETTER O WITH HORN AND GRAVE
1EDE	1EDF	LATIN CAPITAL LETTER O WITH HORN AND HOOK ABOVE	LATIN SMALL LETTER O WITH HORN AND HOOK ABOVE
1EE0	1EE1	LATIN CAPITAL LETTER O WITH HORN AND TILDE	LATIN SMALL LETTER O WITH HORN AND TILDE
1EE2	1EE3	LATIN CAPITAL LETTER O WITH HORN AND DOT BELOW	LATIN SMALL LETTER O WITH HORN AND DOT BELOW
1EE4	1EE5	LATIN CAPITAL LETTER U WITH DOT BELOW	LATIN SMALL LETTER U WITH DOT BELOW
1EE6	1EE7	LATIN CAPITAL LETTER U WITH HOOK ABOVE	LATIN SMALL LETTER U WITH HOOK ABOVE
1EE8	1EE9	LATIN CAPITAL LETTER U WITH HORN AND ACUTE	LATIN SMALL LETTER U WITH HORN AND ACUTE
1EEA	1EEB	LATIN CAPITAL LETTER U WITH HORN AND GRAVE	LATIN SMALL LETTER U WITH HORN AND GRAVE
1EEC	1EED	LATIN CAPITAL LETTER U WITH HORN AND HOOK ABOVE	LATIN SMALL LETTER U WITH HORN AND HOOK ABOVE
1EEE	1EEF	LATIN CAPITAL LETTER U WITH HORN AND TILDE	LATIN SMALL LETTER U WITH HORN AND TILDE
1EF0	1EF1	LATIN CAPITAL LETTER U WITH HORN AND DOT BELOW	LATIN SMALL LETTER U WITH HORN AND DOT BELOW
1EF2	1EF3	LATIN CAPITAL LETTER Y WITH GRAVE	LATIN SMALL LETTER Y WITH GRAVE
1EF4	1EF5	LATIN CAPITAL LETTER Y WITH DOT BELOW	LATIN SMALL LETTER Y WITH DOT BELOW
1EF6	1EF7	LATIN CAPITAL LETTER Y WITH HOOK ABOVE	LATIN SMALL LETTER Y WITH HOOK ABOVE
1EF8	1EF9	LATIN CAPITAL LETTER Y WITH TILDE	LATIN SMALL LETTER Y WITH TILDE
1F08	1F00	GREEK CAPITAL LETTER ALPHA WITH PSILI	GREEK SMALL LETTER ALPHA WITH PSILI
1F09	1F01	GREEK CAPITAL LETTER ALPHA WITH DASIA	GREEK SMALL LETTER ALPHA WITH DASIA
1F0A	1F02	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA
1F0B	1F03	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1F0C	1F04	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA
1F0D	1F05	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA
1F0E	1F06	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI
1F0F	1F07	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI
1F18	1F10	GREEK CAPITAL LETTER EPSILON WITH PSILI	GREEK SMALL LETTER EPSILON WITH PSILI
1F19	1F11	GREEK CAPITAL LETTER EPSILON WITH DASIA	GREEK SMALL LETTER EPSILON WITH DASIA
1F1A	1F12	GREEK CAPITAL LETTER EPSILON WITH PSILI AND VARIA	GREEK SMALL LETTER EPSILON WITH PSILI AND VARIA
1F1B	1F13	GREEK CAPITAL LETTER EPSILON WITH DASIA AND VARIA	GREEK SMALL LETTER EPSILON WITH DASIA AND VARIA
1F1C	1F14	GREEK CAPITAL LETTER EPSILON WITH PSILI AND OXIA	GREEK SMALL LETTER EPSILON WITH PSILI AND OXIA
1F1D	1F15	GREEK CAPITAL LETTER EPSILON WITH DASIA AND OXIA	GREEK SMALL LETTER EPSILON WITH DASIA AND OXIA
1F28	1F20	GREEK CAPITAL LETTER ETA WITH PSILI	GREEK SMALL LETTER ETA WITH PSILI
1F29	1F21	GREEK CAPITAL LETTER ETA WITH DASIA	GREEK SMALL LETTER ETA WITH DASIA
1F2A	1F22	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA	GREEK SMALL LETTER ETA WITH PSILI AND VARIA
1F2B	1F23	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA	GREEK SMALL LETTER ETA WITH DASIA AND VARIA
1F2C	1F24	GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA	GREEK SMALL LETTER ETA WITH PSILI AND OXIA
1F2D	1F25	GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA	GREEK SMALL LETTER ETA WITH DASIA AND OXIA
1F2E	1F26	GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI
1F2F	1F27	GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI
1F38	1F30	GREEK CAPITAL LETTER IOTA WITH PSILI	GREEK SMALL LETTER IOTA WITH PSILI
1F39	1F31	GREEK CAPITAL LETTER IOTA WITH DASIA	GREEK SMALL LETTER IOTA WITH DASIA
1F3A	1F32	GREEK CAPITAL LETTER IOTA WITH PSILI AND VARIA	GREEK SMALL LETTER IOTA WITH PSILI AND VARIA
1F3B	1F33	GREEK CAPITAL LETTER IOTA WITH DASIA AND VARIA	GREEK SMALL LETTER IOTA WITH DASIA AND VARIA
1F3C	1F34	GREEK CAPITAL LETTER IOTA WITH PSILI AND OXIA	GREEK SMALL LETTER IOTA WITH PSILI AND OXIA
1F3D	1F35	GREEK CAPITAL LETTER IOTA WITH DASIA AND OXIA	GREEK SMALL LETTER IOTA WITH DASIA AND OXIA
1F3E	1F36	GREEK CAPITAL LETTER IOTA WITH PSILI AND PERISPOMENI	GREEK SMALL LETTER IOTA WITH PSILI AND PERISPOMENI
1F3F	1F37	GREEK CAPITAL LETTER IOTA WITH DASIA AND PERISPOMENI	GREEK SMALL LETTER IOTA WITH DASIA AND PERISPOMENI
1F48	1F40	GREEK CAPITAL LETTER OMICRON WITH PSILI	GREEK SMALL LETTER OMICRON WITH PSILI
1F49	1F41	GREEK CAPITAL LETTER OMICRON WITH DASIA	GREEK SMALL LETTER OMICRON WITH DASIA
1F4A	1F42	GREEK CAPITAL LETTER OMICRON WITH PSILI AND VARIA	GREEK SMALL LETTER OMICRON WITH PSILI AND VARIA

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1F4B	1F43	GREEK CAPITAL LETTER OMICRON WITH DASIA AND VARIA	GREEK SMALL LETTER OMICRON WITH DASIA AND VARIA
1F4C	1F44	GREEK CAPITAL LETTER OMICRON WITH PSILI AND OXIA	GREEK SMALL LETTER OMICRON WITH PSILI AND OXIA
1F4D	1F45	GREEK CAPITAL LETTER OMICRON WITH DASIA AND OXIA	GREEK SMALL LETTER OMICRON WITH DASIA AND OXIA
1F59	1F51	GREEK CAPITAL LETTER UPSILON WITH OASIS	GREEK SMALL LETTER UPSILON WITH DASIA
1F5B	1F53	GREEK CAPITAL LETTER UPSILON WITH DASIA AND VARIA	GREEK SMALL LETTER UPSILON WITH DASIA AND VARIA
1F5D	1F55	GREEK CAPITAL LETTER UPSILON WITH DASIA AND OXIA	GREEK SMALL LETTER UPSILON WITH DASIA AND OXIA
1F5F	1F57	GREEK CAPITAL LETTER UPSILON WITH DASIA AND PERISPOMENI	GREEK SMALL LETTER UPSILON WITH DASIA AND PERISPOMENI
1F68	1F60	GREEK CAPITAL LETTER OMEGA WITH PSILI	GREEK SMALL LETTER OMEGA WITH PSILI
1F69	1F61	GREEK CAPITAL LETTER OMEGA WITH DASIA	GREEK SMALL LETTER OMEGA WITH DASIA
1F6A	1F62	GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA
1F6B	1F63	GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA
1F6C	1F64	GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA
1F6D	1F65	GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA
1F6E	1F66	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI
1F6F	1F67	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PERISPOMENI	GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI
1F88	1F80	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PROSGEGRAMMENI	GREEK SMALL LETTER ALPHA WITH PSILI AND YPOGEGRAMMENI
1F89	1F81	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ALPHA WITH DASIA AND YPOGEGRAMMENI
1F8A	1F82	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA AND YPOGEGRAMMENI
1F8B	1F83	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA AND YPOGEGRAMMENI
1F8C	1F84	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA AND PROSGEGRAMMEN	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA AND YPOGEGRAMMENI
1F8D	1F85	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA AND PROSGEGRAMMEN	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA AND YPOGEGRAMMENI
1F8E	1F86	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
1F8F	1F87	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
1F98	1F90	GREEK CAPITAL LETTER ETA WITH PSILI AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH PSILI AND YPOGEGRAMMENI

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
1F99	1F91	GREEK CAPITAL LETTER ETA WITH DASIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH DASIA AND YPOGEGRAMMENI
1F9A	1F92	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH PSILI AND VARIA AND YPOGEGRAMMENI
1F9B	1F93	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH DASIA AND VARIA AND YPOGEGRAMMENI
1F9C	1F94	GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH PSILI AND OXIA AND YPOGEGRAMMENI
1F9D	1F95	GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH DASIA AND OXIA AND YPOGEGRAMMENI
1F9E	1F96	GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
1F9F	1F97	GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI
1FA8	1FA0	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH PSILI AND YPOGEGRAMMENI
1FA9	1FA1	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH DASIA AND YPOGEGRAMMENI
1FAA	1FA2	GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA AND YPOGEGRAMMENI
1FAB	1FA3	GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA AND YPOGEGRAMMENI
1FAC	1FA4	GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA AND YPOGEGRAMMENI
1FAD	1FA5	GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA AND YPOGEGRAMMENI
1FAE	1FA6	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI
1FAF	1FA7	GREEK CAPITAL LETTER OMECA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI	GREEK SMALL LETTER OMEGA WITH DASIA AND PEPISPOMENI AND YPOGEGRAMMENI
1FB8	1FB0	GREEK CAPITAL LETTER ALPHA WITH VRACHY	GREEK SMALL LETTER ALPHA WITH VRACHY
1FB9	1FB1	GREEK CAPITAL LETTER ALPHA WITH MACRON	GREEK SMALL LETTER ALPHA WITH MACRON
1FD8	1FD0	GREEK CAPITAL LETTER IOTA WITH VRACHY	GREEK SMALL LETTER IOTA WITH VRACHY
1FD9	1FD1	GREEK CAPITAL LETTER IOTA WITH MACRON	GREEK SMALL LETTER IOTA WITH MACRON
1FE8	1FE0	GREEK CAPITAL LETTER UPSILON WITH VRACHY	GREEK SMALL LETTER UPSILON WITH VRACHY
1FE9	1FE1	GREEK CAPITAL LETTER UPSILON WITH MACRON	GREEK SMALL LETTER UPSILON WITH MACRON
24B6	24D0	CIRCLED LATIN CAPITAL LETTER A	CIRCLED LATIN SMALL LETTER A
24B7	24D1	CIRCLED LATIN CAPITAL LETTER B	CIRCLED LATIN SMALL LETTER B
24B8	24D2	CIRCLED LATIN CAPITAL LETTER C	CIRCLED LATIN SMALL LETTER C

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
24B9	24D3	CIRCLED LATIN CAPITAL LETTER D	CIRCLED LATIN SMALL LETTER D
24BA	24D4	CIRCLED LATIN CAPITAL LETTER E	CIRCLED LATIN SMALL LETTER E
24BB	24D5	CIRCLED LATIN CAPITAL LETTER F	CIRCLED LATIN SMALL LETTER F
24BC	24D6	CIRCLED LATIN CAPITAL LETTER G	CIRCLED LATIN SMALL LETTER G
24BD	24D7	CIRCLED LATIN CAPITAL LETTER H	CIRCLED LATIN SMALL LETTER H
24BE	24D8	CIRCLED LATIN CAPITAL LETTER I	CIRCLED LATIN SMALL LETTER I
24BF	24D9	CIRCLED LATIN CAPITAL LETTER J	CIRCLED LATIN SMALL LETTER J
24C0	24DA	CIRCLED LATIN CAPITAL LETTER K	CIRCLED LATIN SMALL LETTER K
24C1	24DB	CIRCLED LATIN CAPITAL LETTER L	CIRCLED LATIN SMALL LETTER L
24C2	24DC	CIRCLED LATIN CAPITAL LETTER M	CIRCLED LATIN SMALL LETTER M
24C3	24DD	CIRCLED LATIN CAPITAL LETTER N	CIRCLED LATIN SMALL LETTER N
24C4	24DE	CIRCLED LATIN CAPITAL LETTER O	CIRCLED LATIN SMALL LETTER O
24C5	24DF	CIRCLED LATIN CAPITAL LETTER P	CIRCLED LATIN SMALL LETTER P
24C6	24E0	CIRCLED LATIN CAPITAL LETTER Q	CIRCLED LATIN SMALL LETTER Q
24C7	24E1	CIRCLED LATIN CAPITAL LETTER R	CIRCLED LATIN SMALL LETTER R
24C8	24E2	CIRCLED LATIN CAPITAL LETTER S	CIRCLED LATIN SMALL LETTER S
24C9	24E3	CIRCLED LATIN CAPITAL LETTER T	CIRCLED LATIN SMALL LETTER T
24CA	24E4	CIRCLED LATIN CAPITAL LETTER U	CIRCLED LATIN SMALL LETTER U
24CB	24E5	CIRCLED LATIN CAPITAL LETTER V	CIRCLED LATIN SMALL LETTER V
24CC	24E6	CIRCLED LATIN CAPITAL LETTER W	CIRCLED LATIN SMALL LETTER W
24CD	24E7	CIRCLED LATIN CAPITAL LETTER X	CIRCLED LATIN SMALL LETTER X
24CE	24E8	CIRCLED LATIN CAPITAL LETTER Y	CIRCLED LATIN SMALL LETTER Y
24CF	24E9	CIRCLED LATIN CAPITAL LETTER Z	CIRCLED LATIN SMALL LETTER Z
FF21	FF41	FULLWIDTH LATIN CAPITAL LETTER A	FULLWIDTH LATIN SMALL LETTER A
FF22	FF42	FULLWIDTH LATIN CAPITAL LETTER B	FULLWIDTH LATIN SMALL LETTER B
FF23	FF43	FULLWIDTH LATIN CAPITAL LETTER C	FULLWIDTH LATIN SMALL LETTER C
FF24	FF44	FULLWIDTH LATIN CAPITAL LETTER D	FULLWIDTH LATIN SMALL LETTER D
FF25	FF45	FULLWIDTH LATIN CAPITAL LETTER E	FULLWIDTH LATIN SMALL LETTER E
FF26	FF46	FULLWIDTH LATIN CAPITAL LETTER F	FULLWIDTH LATIN SMALL LETTER F
FF27	FF47	FULLWIDTH LATIN CAPITAL LETTER G	FULLWIDTH LATIN SMALL LETTER G

Uppercase code point	Lowercase code point	Uppercase character description	Lowercase character description
FF28	FF48	FULLWIDTH LATIN CAPITAL LETTER H	FULLWIDTH LATIN SMALL LETTER H
FF29	FF49	FULLWIDTH LATIN CAPITAL LETTER I	FULLWIDTH LATIN SMALL LETTER I
FF2A	FF4A	FULLWIDTH LATIN CAPITAL LETTER J	FULLWIDTH LATIN SMALL LETTER J
FF2B	FF4B	FULLWIDTH LATIN CAPITAL LETTER K	FULLWIDTH LATIN SMALL LETTER K
FF2C	FF4C	FULLWIDTH LATIN CAPITAL LETTER L	FULLWIDTH LATIN SMALL LETTER L
FF2D	FF4D	FULLWIDTH LATIN CAPITAL LETTER M	FULLWIDTH LATIN SMALL LETTER M
FF2E	FF4E	FULLWIDTH LATIN CAPITAL LETTER N	FULLWIDTH LATIN SMALL LETTER N
FF2F	FF4F	FULLWIDTH LATIN CAPITAL LETTER O	FULLWIDTH LATIN SMALL LETTER O
FF30	FF50	FULLWIDTH LATIN CAPITAL LETTER P	FULLWIDTH LATIN SMALL LETTER P
FF31	FF51	FULLWIDTH LATIN CAPITAL LETTER Q	FULLWIDTH LATIN SMALL LETTER Q
FF32	FF52	FULLWIDTH LATIN CAPITAL LETTER R	FULLWIDTH LATIN SMALL LETTER R
FF33	FF53	FULLWIDTH LATIN CAPITAL LETTER S	FULLWIDTH LATIN SMALL LETTER S
FF34	FF54	FULLWIDTH LATIN CAPITAL LETTER T	FULLWIDTH LATIN SMALL LETTER T
FF35	FF55	FULLWIDTH LATIN CAPITAL LETTER U	FULLWIDTH LATIN SMALL LETTER U
FF36	FF56	FULLWIDTH LATIN CAPITAL LETTER V	FULLWIDTH LATIN SMALL LETTER V
FF37	FF57	FULLWIDTH LATIN CAPITAL LETTER W	FULLWIDTH LATIN SMALL LETTER W
FF38	FF58	FULLWIDTH LATIN CAPITAL LETTER X	FULLWIDTH LATIN SMALL LETTER X
FF39	FF59	FULLWIDTH LATIN CAPITAL LETTER Y	FULLWIDTH LATIN SMALL LETTER Y
FF3A	FF5A	FULLWIDTH LATIN CAPITAL LETTER Z	FULLWIDTH LATIN SMALL LETTER Z

Unicode lowercase to uppercase conversion mapping table

| The table shows the mapping for the conversion of Unicode from uppercase to lowercase. You can refer
| to this table to see how the Convert Case API converts character data from lowercase to uppercase.

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0061	0041	LATIN SMALL LETTER A	LATIN CAPITAL LETTER A
0062	0042	LATIN SMALL LETTER B	LATIN CAPITAL LETTER B
0063	0043	LATIN SMALL LETTER C	LATIN CAPITAL LETTER C
0064	0044	LATIN SMALL LETTER D	LATIN CAPITAL LETTER D
0065	0045	LATIN SMALL LETTER E	LATIN CAPITAL LETTER E
0066	0046	LATIN SMALL LETTER F	LATIN CAPITAL LETTER F
0067	0047	LATIN SMALL LETTER G	LATIN CAPITAL LETTER G
0068	0048	LATIN SMALL LETTER H	LATIN CAPITAL LETTER H
0069	0049	LATIN SMALL LETTER I	LATIN CAPITAL LETTER I
006A	004A	LATIN SMALL LETTER J	LATIN CAPITAL LETTER J
006B	004B	LATIN SMALL LETTER K	LATIN CAPITAL LETTER K
006C	004C	LATIN SMALL LETTER L	LATIN CAPITAL LETTER L
006D	004D	LATIN SMALL LETTER M	LATIN CAPITAL LETTER M

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
006E	004E	LATIN SMALL LETTER N	LATIN CAPITAL LETTER N
006F	004F	LATIN SMALL LETTER O	LATIN CAPITAL LETTER O
0070	0050	LATIN SMALL LETTER P	LATIN CAPITAL LETTER P
0071	0051	LATIN SMALL LETTER Q	LATIN CAPITAL LETTER Q
0072	0052	LATIN SMALL LETTER R	LATIN CAPITAL LETTER R
0073	0053	LATIN SMALL LETTER S	LATIN CAPITAL LETTER S
0074	0054	LATIN SMALL LETTER T	LATIN CAPITAL LETTER T
0075	0055	LATIN SMALL LETTER U	LATIN CAPITAL LETTER U
0076	0056	LATIN SMALL LETTER V	LATIN CAPITAL LETTER V
0077	0057	LATIN SMALL LETTER W	LATIN CAPITAL LETTER W
0078	0058	LATIN SMALL LETTER X	LATIN CAPITAL LETTER X
0079	0059	LATIN SMALL LETTER Y	LATIN CAPITAL LETTER Y
007A	005A	LATIN SMALL LETTER Z	LATIN CAPITAL LETTER Z
00E0	00C0	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A GRAVE
00E1	00C1	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A ACUTE
00E2	00C2	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A CIRCUMFLEX
00E3	00C3	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A TILDE
00E4	00C4	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A DIAERESIS
00E5	00C5	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A RING
00E6	00C6	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER A E
00E7	00C7	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER C CEDILLA
00E8	00C8	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER E GRAVE
00E9	00C9	LATIN SMALL LETTER A GRAVE	LATIN CAPITAL LETTER E ACUTE
00EA	00CA	LATIN SMALL LETTER E CIRCUMFLEX	LATIN CAPITAL LETTER E CIRCUMFLEX
00EB	00CB	LATIN SMALL LETTER E DIAERESIS	LATIN CAPITAL LETTER E DIAERESIS
00EC	00CC	LATIN SMALL LETTER I GRAVE	LATIN CAPITAL LETTER I GRAVE
00ED	00CD	LATIN SMALL LETTER I ACUTE	LATIN CAPITAL LETTER I ACUTE
00EE	00CE	LATIN SMALL LETTER I CIRCUMFLEX	LATIN CAPITAL LETTER I CIRCUMFLEX
00EF	00CF	LATIN SMALL LETTER I DIAERESIS	LATIN CAPITAL LETTER I DIAERESIS
00F0	00D0	LATIN SMALL LETTER ETH	LATIN CAPITAL LETTER ETH
00F1	00D1	LATIN SMALL LETTER N TILDE	LATIN CAPITAL LETTER N TILDE
00F2	00D2	LATIN SMALL LETTER O GRAVE	LATIN CAPITAL LETTER O GRAVE
00F3	00D3	LATIN SMALL LETTER O ACUTE	LATIN CAPITAL LETTER O ACUTE
00F4	00D4	LATIN SMALL LETTER O CIRCUMFLEX	LATIN CAPITAL LETTER O CIRCUMFLEX
00F5	00D5	LATIN SMALL LETTER O TILDE	LATIN CAPITAL LETTER O TILDE
00F6	00D6	LATIN SMALL LETTER O DIAERESIS	LATIN CAPITAL LETTER O DIAERESIS
00F8	00D8	LATIN SMALL LETTER O SLASH	LATIN CAPITAL LETTER O SLASH
00F9	00D9	LATIN SMALL LETTER U GRAVE	LATIN CAPITAL LETTER U GRAVE
00FA	00DA	LATIN SMALL LETTER U ACUTE	LATIN CAPITAL LETTER U ACUTE

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
00FB	00DB	LATIN SMALL LETTER U CIRCUMFLEX	LATIN CAPITAL LETTER U CIRCUMFLEX
00FC	00DC	LATIN SMALL LETTER U DIAERESIS	LATIN CAPITAL LETTER U DIAERESIS
00FD	00DD	LATIN SMALL LETTER Y ACUTE	LATIN CAPITAL LETTER Y ACUTE
00FE	00DE	LATIN SMALL LETTER THORN	LATIN CAPITAL LETTER THORN
00FF	0178	LATIN SMALL LETTER Y DIAERESIS	LATIN CAPITAL LETTER Y WITH DIAERESIS
0101	0100	LATIN SMALL LETTER A WITH MACRON	LATIN CAPITAL LETTER A WITH MACRON
0103	0102	LATIN SMALL LETTER A WITH BREVE	LATIN CAPITAL LETTER A WITH BREVE
0105	0104	LATIN SMALL LETTER A WITH OGONEK	LATIN CAPITAL LETTER A WITH OGONEK
0107	0106	LATIN SMALL LETTER C WITH ACUTE	LATIN CAPITAL LETTER C WITH ACUTE
0109	0108	LATIN SMALL LETTER C WITH CIRCUMFLEX	LATIN CAPITAL LETTER C WITH CIRCUMFLEX
010B	010A	LATIN SMALL LETTER C WITH DOT ABOVE	LATIN CAPITAL LETTER C WITH DOT ABOVE
010D	010C	LATIN SMALL LETTER C WITH CARON	LATIN CAPITAL LETTER C WITH CARON
010F	010E	LATIN SMALL LETTER D WITH CARON	LATIN CAPITAL LETTER D WITH CARON
0111	0110	LATIN SMALL LETTER D WITH STROKE	LATIN CAPITAL LETTER D WITH STROKE
0113	0112	LATIN SMALL LETTER E WITH MACRON	LATIN CAPITAL LETTER E WITH MACRON
0115	0114	LATIN SMALL LETTER E WITH BREVE	LATIN CAPITAL LETTER E WITH BREVE
0117	0116	LATIN SMALL LETTER E WITH DOT ABOVE	LATIN CAPITAL LETTER E WITH DOT ABOVE
0119	0118	LATIN SMALL LETTER E WITH OGONEK	LATIN CAPITAL LETTER E WITH OGONEK
011B	011A	LATIN SMALL LETTER E WITH CARON	LATIN CAPITAL LETTER E WITH CARON
011D	011C	LATIN SMALL LETTER G WITH CIRCUMFLEX	LATIN CAPITAL LETTER G WITH CIRCUMFLEX
011F	011E	LATIN SMALL LETTER G WITH BREVE	LATIN CAPITAL LETTER G WITH BREVE
0121	0120	LATIN SMALL LETTER G WITH DOT ABOVE	LATIN CAPITAL LETTER G WITH DOT ABOVE
0123	0122	LATIN SMALL LETTER G WITH CEDILLA	LATIN CAPITAL LETTER G WITH CEDILLA
0125	0124	LATIN SMALL LETTER H WITH CIRCUMFLEX	LATIN CAPITAL LETTER H WITH CIRCUMFLEX
0127	0126	LATIN SMALL LETTER H WITH STROKE	LATIN CAPITAL LETTER H WITH STROKE
0129	0128	LATIN SMALL LETTER I WITH TILDE	LATIN CAPITAL LETTER I WITH TILDE
012B	012A	LATIN SMALL LETTER I WITH MACRON	LATIN CAPITAL LETTER I WITH MACRON
012D	012C	LATIN SMALL LETTER I WITH BREVE	LATIN CAPITAL LETTER I WITH BREVE
012F	012E	LATIN SMALL LETTER I WITH OGONEK	LATIN CAPITAL LETTER I WITH OGONEK
0131	0049	LATIN SMALL LETTER DOTLESS I	LATIN CAPITAL LETTER I
0133	0132	LATIN SMALL LIGATURE IJ	LATIN CAPITAL LIGATURE IJ

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0135	0134	LATIN SMALL LETTER J WITH CIRCUMFLEX	LATIN CAPITAL LETTER J WITH CIRCUMFLEX
0137	0136	LATIN SMALL LETTER K WITH CEDILLA	LATIN CAPITAL LETTER K WITH CEDILLA
013A	0139	LATIN SMALL LETTER L WITH ACUTE	LATIN CAPITAL LETTER L WITH ACUTE
013C	013B	LATIN SMALL LETTER L WITH CEDILLA	LATIN CAPITAL LETTER L WITH CEDILLA
013E	013D	LATIN SMALL LETTER L WITH CARON	LATIN CAPITAL LETTER L WITH CARON
0140	013F	LATIN SMALL LETTER L WITH MIDDLE DOT	LATIN CAPITAL LETTER L WITH MIDDLE DOT
0142	0141	LATIN SMALL LETTER L WITH STROKE	LATIN CAPITAL LETTER L WITH STROKE
0144	0143	LATIN SMALL LETTER N WITH ACUTE	LATIN CAPITAL LETTER N WITH ACUTE
0146	0145	LATIN SMALL LETTER N WITH CEDILLA	LATIN CAPITAL LETTER N WITH CEDILLA
0148	0147	LATIN SMALL LETTER N WITH CARON	LATIN CAPITAL LETTER N WITH CARON
014B	014A	LATIN SMALL LETTER ENG (SAMI)	LATIN CAPITAL LETTER ENG (SAMI)
014D	014C	LATIN SMALL LETTER O WITH MACRON	LATIN CAPITAL LETTER O WITH MACRON
014F	014E	LATIN SMALL LETTER O WITH BREVE	LATIN CAPITAL LETTER O WITH BREVE
0151	0150	LATIN SMALL LETTER O WITH DOUBLE ACUTE	LATIN CAPITAL LETTER O WITH DOUBLE ACUTE
0153	0152	LATIN SMALL LIGATURE OE	LATIN CAPITAL LIGATURE OE
0155	0154	LATIN SMALL LETTER R WITH ACUTE	LATIN CAPITAL LETTER R WITH ACUTE
0157	0156	LATIN SMALL LETTER R WITH CEDILLA	LATIN CAPITAL LETTER R WITH CEDILLA
0159	0158	LATIN SMALL LETTER R WITH CARON	LATIN CAPITAL LETTER R WITH CARON
015B	015A	LATIN SMALL LETTER S WITH ACUTE	LATIN CAPITAL LETTER S WITH ACUTE
015D	015C	LATIN SMALL LETTER S WITH CIRCUMFLEX	LATIN CAPITAL LETTER S WITH CIRCUMFLEX
015F	015E	LATIN SMALL LETTER S WITH CEDILLA	LATIN CAPITAL LETTER S WITH CEDILLA
0161	0160	LATIN SMALL LETTER S WITH CARON	LATIN CAPITAL LETTER S WITH CARON
0163	0162	LATIN SMALL LETTER T WITH CEDILLA	LATIN CAPITAL LETTER T WITH CEDILLA
0165	0164	LATIN SMALL LETTER T WITH CARON	LATIN CAPITAL LETTER T WITH CARON
0167	0166	LATIN SMALL LETTER T WITH STROKE	LATIN CAPITAL LETTER T WITH STROKE
0169	0168	LATIN SMALL LETTER U WITH TILDE	LATIN CAPITAL LETTER U WITH TILDE
016B	016A	LATIN SMALL LETTER U WITH MACRON	LATIN CAPITAL LETTER U WITH MACRON
016D	016C	LATIN SMALL LETTER U WITH BREVE	LATIN CAPITAL LETTER U WITH BREVE
016F	016E	LATIN SMALL LETTER U WITH RING ABOVE	LATIN CAPITAL LETTER U WITH RING ABOVE
0171	0170	LATIN SMALL LETTER U WITH DOUBLE ACUTE	LATIN CAPITAL LETTER U WITH DOUBLE ACUTE
0173	0172	LATIN SMALL LETTER U WITH OGONEK	LATIN CAPITAL LETTER U WITH OGONEK

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0175	0174	LATIN SMALL LETTER W WITH CIRCUMFLEX	LATIN CAPITAL LETTER W WITH CIRCUMFLEX
0177	0176	LATIN SMALL LETTER Y WITH CIRCUMFLEX	LATIN CAPITAL LETTER Y WITH CIRCUMFLEX
017A	0179	LATIN SMALL LETTER Z WITH ACUTE	LATIN CAPITAL LETTER Z WITH ACUTE
017C	017B	LATIN SMALL LETTER Z WITH DOT ABOVE	LATIN CAPITAL LETTER Z WITH DOT ABOVE
017E	017D	LATIN SMALL LETTER Z WITH CARON	LATIN CAPITAL LETTER Z WITH CARON
0183	0182	LATIN SMALL LETTER B WITH TOPBAR	LATIN CAPITAL LETTER B WITH TOPBAR
0185	0184	LATIN SMALL LETTER TONE SIX	LATIN CAPITAL LETTER TONE SIX
0188	0187	LATIN SMALL LETTER C WITH HOOK	LATIN CAPITAL LETTER C WITH HOOK
018C	018B	LATIN SMALL LETTER D WITH TOPBAR	LATIN CAPITAL LETTER D WITH TOPBAR
0192	0191	LATIN SMALL LETTER F WITH HOOK	LATIN CAPITAL LETTER F WITH HOOK
0199	0198	LATIN SMALL LETTER K WITH HOOK	LATIN CAPITAL LETTER K WITH HOOK
01A1	01A0	LATIN SMALL LETTER O WITH HORN	LATIN CAPITAL LETTER O WITH HORN
01A3	01A2	LATIN SMALL LETTER OI	LATIN CAPITAL LETTER OI
01A5	01A4	LATIN SMALL LETTER P WITH HOOK	LATIN CAPITAL LETTER P WITH HOOK
01A8	01A7	LATIN SMALL LETTER TONE TWO	LATIN CAPITAL LETTER TONE TWO
01AD	01AC	LATIN SMALL LETTER T WITH HOOK	LATIN CAPITAL LETTER T WITH HOOK
01B0	01AF	LATIN SMALL LETTER U WITH HORN	LATIN CAPITAL LETTER U WITH HORN
01B4	01B3	LATIN SMALL LETTER Y WITH HOOK	LATIN CAPITAL LETTER Y WITH HOOK
01B6	01B5	LATIN SMALL LETTER Z WITH STROKE	LATIN CAPITAL LETTER Z WITH STROKE
01B9	01B8	LATIN SMALL LETTER EZH REVERSED	LATIN CAPITAL LETTER EZH REVERSED
01BD	01BC	LATIN SMALL LETTER TONE FIVE	LATIN CAPITAL LETTER TONE FIVE
01C6	01C4	LATIN SMALL LETTER DZ WITH CARON	LATIN CAPITAL LETTER DZ WITH CARON
01C9	01C7	LATIN SMALL LETTER LJ	LATIN CAPITAL LETTER LJ
01CC	01CA	LATIN SMALL LETTER NJ	LATIN CAPITAL LETTER NJ
01CE	01CD	LATIN SMALL LETTER A WITH CARON	LATIN CAPITAL LETTER A WITH CARON
01D0	01CF	LATIN SMALL LETTER I WITH CARON	LATIN CAPITAL LETTER I WITH CARON
01D2	01D1	LATIN SMALL LETTER O WITH CARON	LATIN CAPITAL LETTER O WITH CARON
01D4	01D3	LATIN SMALL LETTER U WITH CARON	LATIN CAPITAL LETTER U WITH CARON
01D6	01D5	LATIN SMALL LETTER U WITH DIAERESIS AND MACRON	LATIN CAPITAL LETTER U WITH DIAERESIS AND MACRON
01D8	01D7	LATIN SMALL LETTER U WITH DIAERESIS AND ACUTE	LATIN CAPITAL LETTER U WITH DIAERESIS AND ACUTE
01DA	01D9	LATIN SMALL LETTER U WITH DIAERESIS AND CARON	LATIN CAPITAL LETTER U WITH DIAERESIS AND CARON
01DC	01DB	LATIN SMALL LETTER U WITH DIAERESIS AND GRAVE	LATIN CAPITAL LETTER U WITH DIAERESIS AND GRAVE

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
01DF	01DE	LATIN SMALL LETTER A WITH DIAERESIS AND MACRON	LATIN CAPITAL LETTER A WITH DIAERESIS AND MACRON
01E1	01E0	LATIN SMALL LETTER A WITH DOT ABOVE AND MACRON	LATIN CAPITAL LETTER A WITH DOT ABOVE AND MACRON
01E3	01E2	LATIN SMALL LIGATURE AE WITH MACRON	LATIN CAPITAL LIGATURE AE MTH MACRON
01E5	01E4	LATIN SMALL LETTER G WITH STROKE	LATIN CAPITAL LETTER G WITH STROKE
01E7	01E6	LATIN SMALL LETTER G WITH CARON	LATIN CAPITAL LETTER G WITH CARON
01E9	01E8	LATIN SMALL LETTER K WITH CARON	LATIN CAPITAL LETTER K WITH CARON
01EB	01EA	LATIN SMALL LETTER O WITH OGONEK	LATIN CAPITAL LETTER O WITH OGONEK
01ED	01EC	LATIN SMALL LETTER O WITH OGONEK AND MACRON	LATIN CAPITAL LETTER O WITH OGONEK AND MACRON
01EF	01EE	LATIN SMALL LETTER EZH WITH CARON	LATIN CAPITAL LETTER EZH WITH CARON
01F3	01F1	LATIN SMALL LETTER DZ	LATIN CAPITAL LETTER DZ
01F5	01F4	LATIN SMALL LETTER G WITH ACUTE	LATIN CAPITAL LETTER G WITH ACUTE
01FB	01FA	LATIN SMALL LETTER A WITH RING ABOVE AND ACUTE	LATIN CAPITAL LETTER A WITH RING ABOVE AND ACUTE
01FD	01FC	LATIN SMALL LIGATURE AE WITH ACUTE	LATIN CAPITAL LIGATURE AE WITH ACUTE
01FF	01FE	LATIN SMALL LETTER O WITH STROKE AND ACUTE	LATIN CAPITAL LETTER O WITH STROKE AND ACUTE
0201	0200	LATIN SMALL LETTER A WITH DOUBLE GRAVE	LATIN CAPITAL LETTER A WITH DOUBLE GRAVE
0203	0202	LATIN SMALL LETTER A WITH INVERTED BREVE	LATIN CAPITAL LETTER A WITH INVERTED BREVE
0205	0204	LATIN SMALL LETTER E WITH DOUBLE GRAVE	LATIN CAPITAL LETTER E WITH DOUBLE GRAVE
0207	0206	LATIN SMALL LETTER E WITH INVERTED BREVE	LATIN CAPITAL LETTER E WITH INVERTED BREVE
0209	0208	LATIN SMALL LETTER I WITH DOUBLE GRAVE	LATIN CAPITAL LETTER I WITH DOUBLE GRAVE
020B	020A	LATIN SMALL LETTER I WITH INVERTED BREVE	LATIN CAPITAL LETTER I WITH INVERTED BREVE
020D	020C	LATIN SMALL LETTER O WITH DOUBLE GRAVE	LATIN CAPITAL LETTER O WITH DOUBLE GRAVE
020F	020E	LATIN SMALL LETTER O WITH INVERTED BREVE	LATIN CAPITAL LETTER O WITH INVERTED BREVE
0211	0210	LATIN SMALL LETTER R WITH DOUBLE GRAVE	LATIN CAPITAL LETTER R WITH DOUBLE GRAVE
0213	0212	LATIN SMALL LETTER R WITH INVERTED BREVE	LATIN CAPITAL LETTER R WITH INVERTED BREVE
0215	0214	LATIN SMALL LETTER U WITH DOUBLE GRAVE	LATIN CAPITAL LETTER U WITH DOUBLE GRAVE
0217	0216	LATIN SMALL LETTER U WITH INVERTED BREVE	LATIN CAPITAL LETTER U WITH INVERTED BREVE
0253	0181	LATIN SMALL LETTER B WITH HOOK	LATIN CAPITAL LETTER B WITH HOOK
0254	0186	LATIN SMALL LETTER OPEN O	LATIN CAPITAL LETTER OPEN O
0257	018A	LATIN SMALL LETTER D WITH HOOK	LATIN CAPITAL LETTER D WITH HOOK
0258	018E	LATIN SMALL LETTER REVERSED E	LATIN CAPITAL LETTER REVERSED E
0259	018F	LATIN SMALL LETTER SCHWA	LATIN CAPITAL LETTER SCHWA
025B	0190	LATIN SMALL LETTER OPEN E	LATIN CAPITAL LETTER OPEN E

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0260	0193	LATIN SMALL LETTER G WITH HOOK	LATIN CAPITAL LETTER G WITH HOOK
0263	0194	LATIN SMALL LETTER GAMMA	LATIN CAPITAL LETTER GAMMA
0268	0197	LATIN SMALL LETTER I WITH STROKE	LATIN CAPITAL LETTER I WITH STROKE
0269	0196	LATIN SMALL LETTER IOTA	LATIN CAPITAL LETTER IOTA
026F	019C	LATIN SMALL LETTER TURNED M	LATIN CAPITAL LETTER TURNED M
0272	019D	LATIN SMALL LETTER N WITH LEFT HOOK	LATIN CAPITAL LETTER N WITH LEFT HOOK
0275	019F	LATIN SMALL LETTER BARRED O	LATIN CAPITAL LETTER O WITH MIDDLE TILDE
0283	01A9	LATIN SMALL LETTER ESH	LATIN CAPITAL LETTER ESH
0288	01AE	LATIN SMALL LETTER T WITH RETROFLEX HOOK	LATIN CAPITAL LETTER T WITH RETROFLEX HOOK
028A	01B1	LATIN SMALL LETTER UPSILON	LATIN CAPITAL LETTER UPSILON
028B	01B2	LATIN SMALL LETTER V WITH HOOK	LATIN CAPITAL LETTER V WITH HOOK
0292	01B7	LATIN SMALL LETTER EZH	LATIN CAPITAL LETTER EZH
03AC	0386	GREEK SMALL LETTER ALPHA WITH TONOS	GREEK CAPITAL LETTER ALPHA WITH TONOS
03AD	0388	GREEK SMALL LETTER EPSILON WITH TONOS	GREEK CAPITAL LETTER EPSILON WITH TONOS
03AE	0389	GREEK SMALL LETTER ETA WITH TONOS	GREEK CAPITAL LETTER ETA WITH TONOS
03AF	038A	GREEK SMALL LETTER IOTA WITH TONOS	GREEK CAPITAL LETTER IOTA WITH TONOS
03B1	0391	GREEK SMALL LETTER ALPHA	GREEK CAPITAL LETTER ALPHA
03B2	0392	GREEK SMALL LETTER BETA	GREEK CAPITAL LETTER BETA
03B3	0393	GREEK SMALL LETTER GAMMA	GREEK CAPITAL LETTER GAMMA
03B4	0394	GREEK SMALL LETTER DELTA	GREEK CAPITAL LETTER DELTA
03B5	0395	GREEK SMALL LETTER EPSILON	GREEK CAPITAL LETTER EPSILON
03B6	0396	GREEK SMALL LETTER ZETA	GREEK CAPITAL LETTER ZETA
03B7	0397	GREEK SMALL LETTER ETA	GREEK CAPITAL LETTER ETA
03B8	0398	GREEK SMALL LETTER THETA	GREEK CAPITAL LETTER THETA
03B9	0399	GREEK SMALL LETTER IOTA	GREEK CAPITAL LETTER IOTA
03BA	039A	GREEK SMALL LETTER KAPPA	GREEK CAPITAL LETTER KAPPA
03BB	039B	GREEK SMALL LETTER LAMDA	GREEK CAPITAL LETTER LAMDA
03BC	039C	GREEK SMALL LETTER MU	GREEK CAPITAL LETTER MU
03BD	039D	GREEK SMALL LETTER NU	GREEK CAPITAL LETTER NU
03BE	039E	GREEK SMALL LETTER XI	GREEK CAPITAL LETTER XI
03BF	039F	GREEK SMALL LETTER OMICRON	GREEK CAPITAL LETTER OMICRON
03C0	03A0	GREEK SMALL LETTER PI	GREEK CAPITAL LETTER PI
03C1	03A1	GREEK SMALL LETTER RHO	GREEK CAPITAL LETTER RHO
03C3	03A3	GREEK SMALL LETTER SIGMA	GREEK CAPITAL LETTER SIGMA
03C4	03A4	GREEK SMALL LETTER TAU	GREEK CAPITAL LETTER TAU
03C5	03A5	GREEK SMALL LETTER UPSILON	GREEK CAPITAL LETTER UPSILON
03C6	03A6	GREEK SMALL LETTER PHI	GREEK CAPITAL LETTER PHI
03C7	03A7	GREEK SMALL LETTER CHI	GREEK CAPITAL LETTER CHI
03C8	03A8	GREEK SMALL LETTER PSI	GREEK CAPITAL LETTER PSI
03C9	03A9	GREEK SMALL LETTER OMEGA	GREEK CAPITAL LETTER OMEGA

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
03CA	03AA	GREEK SMALL LETTER IOTA WITH DIALYTIKA	GREEK CAPITAL LETTER IOTA WITH DIALYTIKA
03CB	03AB	GREEK SMALL LETTER UPSILON WITH DIALYTIKA	GREEK CAPITAL LETTER UPSILON WITH DIALYTIKA
03CC	038C	GREEK SMALL LETTER OMICRON WITH TONOS	GREEK CAPITAL LETTER OMICRON WITH TONOS
03CD	038E	GREEK SMALL LETTER UPSILON WITH TONOS	GREEK CAPITAL LETTER UPSILON WITH TONOS
03CE	038F	GREEK SMALL LETTER OMEGA WITH TONOS	GREEK CAPITAL LETTER OMEGA WITH TONOS
03E3	03E2	COPTIC SMALL LETTER SHEI	COPTIC CAPITAL LETTER SHEI
03E5	03E4	COPTIC SMALL LETTER FEI	COPTIC CAPITAL LETTER FEI
03E7	03E6	COPTIC SMALL LETTER KHEI	COPTIC CAPITAL LETTER KHEI
03E9	03E8	COPTIC SMALL LETTER HORI	COPTIC CAPITAL LETTER HORI
03EB	03EA	COPTIC SMALL LETTER GANGIA	COPTIC CAPITAL LETTER GANGIA
03ED	03EC	COPTIC SMALL LETTER SHIMA	COPTIC CAPITAL LETTER SHIMA
03EF	03EE	COPTIC SMALL LETTER DEI	COPTIC CAPITAL LETTER DEI
0430	0410	CYRILLIC SMALL LETTER A	CYRILLIC CAPITAL LETTER A
0431	0411	CYRILLIC SMALL LETTER BE	CYRILLIC CAPITAL LETTER BE
0432	0412	CYRILLIC SMALL LETTER VE	CYRILLIC CAPITAL LETTER VE
0433	0413	CYRILLIC SMALL LETTER GHE	CYRILLIC CAPITAL LETTER GHE
0434	0414	CYRILLIC SMALL LETTER DE	CYRILLIC CAPITAL LETTER DE
0435	0415	CYRILLIC SMALL LETTER IE	CYRILLIC CAPITAL LETTER IE
0436	0416	CYRILLIC SMALL LETTER ZHE	CYRILLIC CAPITAL LETTER ZHE
0437	0417	CYRILLIC SMALL LETTER ZE	CYRILLIC CAPITAL LETTER ZE
0438	0418	CYRILLIC SMALL LETTER I	CYRILLIC CAPITAL LETTER I
0439	0419	CYRILLIC SMALL LETTER SHORT I	CYRILLIC CAPITAL LETTER SHORT I
043A	041A	CYRILLIC SMALL LETTER KA	CYRILLIC CAPITAL LETTER KA
043B	041B	CYRILLIC SMALL LETTER EL	CYRILLIC CAPITAL LETTER EL
043C	041C	CYRILLIC SMALL LETTER EM	CYRILLIC CAPITAL LETTER EM
043D	041D	CYRILLIC SMALL LETTER EN	CYRILLIC CAPITAL LETTER EN
043E	041E	CYRILLIC SMALL LETTER O	CYRILLIC CAPITAL LETTER O
043F	041F	CYRILLIC SMALL LETTER PE	CYRILLIC CAPITAL LETTER PE
0440	0420	CYRILLIC SMALL LETTER ER	CYRILLIC CAPITAL LETTER ER
0441	0421	CYRILLIC SMALL LETTER ES	CYRILLIC CAPITAL LETTER ES
0442	0422	CYRILLIC SMALL LETTER TE	CYRILLIC CAPITAL LETTER TE
0443	0423	CYRILLIC SMALL LETTER U	CYRILLIC CAPITAL LETTER U
0444	0424	CYRILLIC SMALL LETTER EF	CYRILLIC CAPITAL LETTER EF
0445	0425	CYRILLIC SMALL LETTER HA	CYRILLIC CAPITAL LETTER HA
0446	0426	CYRILLIC SMALL LETTER TSE	CYRILLIC CAPITAL LETTER TSE
0447	0427	CYRILLIC SMALL LETTER CHE	CYRILLIC CAPITAL LETTER CHE
0448	0428	CYRILLIC SMALL LETTER SHA	CYRILLIC CAPITAL LETTER SHA
0449	0429	CYRILLIC SMALL LETTER SHCHA	CYRILLIC CAPITAL LETTER SHCHA
044A	042A	CYRILLIC SMALL LETTER HARD SIGN	CYRILLIC CAPITAL LETTER HARD SIGN
044B	042B	CYRILLIC SMALL LETTER YERU	CYRILLIC CAPITAL LETTER YERU
044C	042C	CYRILLIC SMALL LETTER SOFT SIGN	CYRILLIC CAPITAL LETTER SOFT SIGN
044D	042D	CYRILLIC SMALL LETTER E	CYRILLIC CAPITAL LETTER E
044E	042E	CYRILLIC SMALL LETTER YU	CYRILLIC CAPITAL LETTER YU
044F	042F	CYRILLIC SMALL LETTER YA	CYRILLIC CAPITAL LETTER YA

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0451	0401	CYRILLIC SMALL LETTER IO	CYRILLIC CAPITAL LETTER IO
0452	0402	CYRILLIC SMALL LETTER DJE (SERBOCROATIAN)	CYRILLIC CAPITAL LETTER DJE (SERBOCROATIAN)
0453	0403	CYRILLIC SMALL LETTER GJE	CYRILLIC CAPITAL LETTER GJE
0454	0404	CYRILLIC SMALL LETTER UKRAINIAN IE	CYRILLIC CAPITAL LETTER UKRAINIAN IE
0455	0405	CYRILLIC SMALL LETTER DZE	CYRILLIC CAPITAL LETTER DZE
0456	0406	CYRILLIC SMALL LETTER BYELORUSSIAN-UKRAINIAN I	CYRILLIC CAPITAL LETTER BYELORUSSIAN_UKRAINIAN I
0457	0407	CYRILLIC SMALL LETTER YI (UKRAINIAN)	CYRILLIC CAPITAL LETTER YI (UKRAINIAN)
0458	0408	CYRILLIC SMALL LETTER JE	CYRILLIC CAPITAL LETTER JE
0459	0409	CYRILLIC SMALL LETTER LJE	CYRILLIC CAPITAL LETTER LJE
045A	040A	CYRILLIC SMALL LETTER NJE	CYRILLIC CAPITAL LETTER NJE
045B	040B	CYRILLIC SMALL LETTER TSHE (SERBOCROATIAN)	CYRILLIC CAPITAL LETTER TSHE (SERBOCROATIAN)
045C	040C	CYRILLIC SMALL LETTER KJE	CYRILLIC CAPITAL LETTER KJE
045E	040E	CYRILLIC SMALL LETTER SHORT U (BYELORUSSIAN)	CYRILLIC CAPITAL LETTER SHORT U (BYELORUSSIAN)
045F	040F	CYRILLIC SMALL LETTER DZHE	CYRILLIC CAPITAL LETTER DZHE
0461	0460	CYRILLIC SMALL LETTER OMEGA	CYRILLIC CAPITAL LETTER OMEGA
0463	0462	CYRILLIC SMALL LETTER YAT	CYRILLIC CAPITAL LETTER YAT
0465	0464	CYRILLIC SMALL LETTER IOTIFIED E	CYRILLIC CAPITAL LETTER IOTIFIED E
0467	0466	CYRILLIC SMALL LETTER LITTLE YUS	CYRILLIC CAPITAL LETTER LITTLE YUS
0469	0468	CYRILLIC SMALL LETTER IOTIFIED LITTLE YUS	CYRILLIC CAPITAL LETTER IOTIFIED LITTLE YUS
046B	046A	CYRILLIC SMALL LETTER BIG YUS	CYRILLIC CAPITAL LETTER BIG YUS
046D	046C	CYRILLIC SMALL LETTER IOTIFIED BIG YUS	CYRILLIC CAPITAL LETTER IOTIFIED BIG YUS
046F	046E	CYRILLIC SMALL LETTER KSI	CYRILLIC CAPITAL LETTER KSI
0471	0470	CYRILLIC SMALL LETTER PSI	CYRILLIC CAPITAL LETTER PSI
0473	0472	CYRILLIC SMALL LETTER FITA	CYRILLIC CAPITAL LETTER FITA
0475	0474	CYRILLIC SMALL LETTER IZHITSA	CYRILLIC CAPITAL LETTER IZHITSA
0477	0476	CYRILLIC SMALL LETTER IZHITSA WITH DOUBLE GRAVE ACCENT	CYRILLIC CAPITAL LETTER IZHITSA WITH DOUBLE GRAVE ACCENT
0479	0478	CYRILLIC SMALL LETTER UK	CYRILLIC CAPITAL LETTER UK
047B	047A	CYRILLIC SMALL LETTER ROUND OMEGA	CYRILLIC CAPITAL LETTER ROUND OMEGA
047D	047C	CYRILLIC SMALL LETTER OMEGA WITH TITLO	CYRILLIC CAPITAL LETTER OMEGA WITH TITLO
047F	047E	CYRILLIC SMALL LETTER OT	CYRILLIC CAPITAL LETTER OT
0481	0480	CYRILLIC SMALL LETTER KOPPA	CYRILLIC CAPITAL LETTER KOPPA
0491	0490	CYRILLIC SMALL LETTER GHE WITH UPTURN	CYRILLIC CAPITAL LETTER GHE WITH UPTURN
0493	0492	CYRILLIC SMALL LETTER GHE WITH STROKE	CYRILLIC CAPITAL LETTER GHE WITH STROKE
0495	0494	CYRILLIC SMALL LETTER GHE WITH MIDDLE HOOK	CYRILLIC CAPITAL LETTER GHE WITH MIDDLE HOOK
0497	0496	CYRILLIC SMALL LETTER ZHE WITH DESCENDER	CYRILLIC CAPITAL LETTER ZHE WITH DESCENDER

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0499	0498	CYRILLIC SMALL LETTER ZE WITH DESCENDER	CYRILLIC CAPITAL LETTER ZE WITH DESCENDER
049B	049A	CYRILLIC SMALL LETTER KA WITH DESCENDER	CYRILLIC CAPITAL LETTER KA WITH DESCENDER
049D	049C	CYRILLIC SMALL LETTER KA WITH VERTICAL STROKE	CYRILLIC CAPITAL LETTER KA WITH VERTICAL STROKE
049F	049E	CYRILLIC SMALL LETTER KA WITH STROKE	CYRILLIC CAPITAL LETTER KA WITH STROKE
04A1	04A0	CYRILLIC SMALL LETTER EASHKIR KA	CYRILLIC CAPITAL LETTER BASHKIR KA
04A3	04A2	CYRILLIC SMALL LETTER EN WITH DESCENOER	CYRILLIC CAPITAL LETTER EN WITH DESCENDER
04A5	04A4	CYRILLIC SMALL LIGATURE EN GHE	CYRILLIC CAPITAL LIGATURE EN GHF
04A7	04A6	CYRILLIC SMALL LETTER PE WITH MIDDLE HOOK (ABKHASIAN)	CYRILLIC CAPITAL LETTER PE WITH MIDDLE HOOK (ABKHASIAN)
04A9	04A8	CYRILLIC SMALL LETTER ABKHASIAN HA	CYRILLIC CAPITAL LETTER ABKHASIAN HA
04AB	04AA	CYRILLIC SMALL LETTER ES WITH DESCENDER	CYRILLIC CAPITAL LETTER ES WITH DESCENDER
04AD	04AC	CYRILLIC SMALL LETTER TE WITH DESCENDER	CYRILLIC CAPITAL LETTER TE WITH DESCENDER
04AF	04AE	CYRILLIC SMALL LETTER STRAIGHT U	CYRILLIC CAPITAL LETTER STRAIGHT U
04B1	04B0	CYRILLIC SMALL LETTER STRAIGHT U WITH STROKE	CYRILLIC CAPITAL LETTER STRAIGHT U WITH STROKE
04B3	04B2	CYRILLIC SMALL LETTER HA WITH DESCENDER	CYRILLIC CAPITAL LETTER HA WITH DESCENDER
04B5	04B4	CYRILLIC SMALL LIGATURE TE TSE (ABKHASIAN)	CYRILLIC CAPITAL LIGATURE TE TSE (ABKHASIAN)
04B7	04B6	CYRILLIC SMALL LETTER CHE WITH DESCENDER	CYRILLIC CAPITAL LETTER CHE WITH DESCENDER
04B9	04B8	CYRILLIC SMALL LETTER CHE WITH VERTICAL STROKE	CYRILLIC CAPITAL LETTER CHE WITH VERTICAL STROKE
04BB	04BA	CYRILLIC SMALL LETTER SHHA	CYRILLIC CAPITAL LETTER SHHA
04BD	04BC	CYRILLIC SMALL LETTER ABKHASIAN CHE	CYRILLIC CAPITAL LETTER ABKHASIAN CHE
04BF	04BE	CYRILLIC SMALL LETTER ABKHASIAN CHE WITH DESCENDER	CYRILLIC CAPITAL LETTER ABKHASIAN CHE WITH DESCENDER
04C2	04C1	CYRILLIC SMALL LETTER ZHE WITH BREVE	CYRILLIC CAPITAL LETTER ZHE WITH BREVE
04C4	04C3	CYRILLIC SMALL LETTER KA WITH HOOK	CYRILLIC CAPITAL LETTER KA WITH HOOK
04C8	04C7	CYRILLIC SMALL LETTER EN WITH HOOK	CYRILLIC CAPITAL LETTER EN WITH HOOK
04CC	04CB	CYRILLIC SMALL LETTER KHAKASSIAN CHE	CYRILLIC CAPITAL LETTER KHAKASSIAN CHE
04D1	04D0	CYRILLIC SMALL LETTER A WITH BREVE	CYRILLIC CAPITAL LETTER A WITH BREVE
04D3	04D2	CYRILLIC SMALL LETTER A WITH DIAERESIS	CYRILLIC CAPITAL LETTER A WITH DIAERESIS
04D5	04D4	CYRILLIC SMALL LIGATURE A IE	CYRILLIC CAPITAL LIGATURE A IE
04D7	04D6	CYRILLIC SMALL LETTER IE WITH BREVE	CYRILLIC CAPITAL LETTER IE WITH BREVE
04D9	04D8	CYRILLIC SMALL LETTER SCHWA	CYRILLIC CAPITAL LETTER SCHWA
04DB	04DA	CYRILLIC SMALL LETTER SCHWA WITH DIAERESIS	CYRILLIC CAPITAL LETTER SCHWA WITH DIAERESIS

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
04DD	04DC	CYRILLIC SMALL LETTER ZHE WITH DIAERESIS	CYRILLIC CAPITAL LETTER ZHE WITH DIAERESIS
04DF	04DE	CYRILLIC SMALL LETTER ZE WITH DIAERESIS	CYRILLIC CAPITAL LETTER ZE WITH DIAERESIS
04E1	04E0	CYRILLIC SMALL LETTER ABKHASIAN DZE	CYRILLIC CAPITAL LETTER ABKHASIAN DZE
04E3	04E2	CYRILLIC SMALL LETTER I WITH MACRON	CYRILLIC CAPITAL LETTER I WITH MACRON
04E5	04E4	CYRILLIC SMALL LETTER I WITH DIAERESIS	CYRILLIC CAPITAL LETTER I WITH DIAERESIS
04E7	04E6	CYRILLIC SMALL LETTER O WITH DIAERESIS	CYRILLIC CAPITAL LETTER O WITH DIAERESIS
04E9	04E8	CYRILLIC SMALL LETTER BARRED O	CYRILLIC CAPITAL LETTER BARRED O
04EB	04EA	CYRILLIC SMALL LETTER BARRED O WITH DIAERESIS	CYRILLIC CAPITAL LETTER BARRED O WITH DIAERESIS
04EF	04EE	CYRILLIC SMALL LETTER U WITH MACRON	CYRILLIC CAPITAL LETTER U WITH MACRON
04F1	04F0	CYRILLIC SMALL LETTER U WITH DIAERESIS	CYRILLIC CAPITAL LETTER U WITH DIAERESIS
04F3	04F2	CYRILLIC SMALL LETTER U WITH DOUBLE ACUTE	CYRILLIC CAPITAL LETTER U WITH DOUBLE ACUTE
04F5	04F4	CYRILLIC SMALL LETTER CHE AITH DIAERESIS	CYRILLIC CAPITAL LETTER CHE WITH DIAERESIS
04F9	04F8	CYRILLIC SMALL LETTER YERU WITH DIAERESIS	CYRILLIC CAPITAL LETTER YERU WITH DIAERESIS
0561	0531	ARMENIAN SMALL LETTER AYB	ARMENIAN CAPITAL LETTER AYB
0562	0532	ARMENIAN SMALL LETTER BEN	ARMENIAN CAPITAL LETTER BEN
0563	0533	ARMENIAN SMALL LETTER GIM	ARMENIAN CAPITAL LETTER GIM
0564	0534	ARMENIAN SMALL LETTER DA	ARMENIAN CAPITAL LETTER DA
0565	0535	ARMENIAN SMALL LETTER ECH	ARMENIAN CAPITAL LETTER ECH
0566	0536	ARMENIAN SMALL LETTER ZA	ARMENIAN CAPITAL LETTER ZA
0567	0537	ARMENIAN SMALL LETTER EH	ARMENIAN CAPITAL LETTER EH
0568	0538	ARMENIAN SMALL LETTER ET	ARMENIAN CAPITAL LETTER ET
0569	0539	ARMENIAN SMALL LETTER TO	ARMENIAN CAPITAL LETTER TO
056A	053A	ARMENIAN SMALL LETTER ZHE	ARMENIAN CAPITAL LETTER ZHE
056B	053B	ARMENIAN SMALL LETTER INI	ARMENIAN CAPITAL LETTER INI
056C	053C	ARMENIAN SMALL LETTER LIWN	ARMENIAN CAPITAL LETTER LIWN
056D	053D	ARMENIAN SMALL LETTER XEH	ARMENIAN CAPITAL LETTER XEH
056E	053E	ARMENIAN SMALL LETTER CA	ARMENIAN CAPITAL LETTER CA
056F	053F	ARMENIAN SMALL LETTER KEN	ARMENIAN CAPITAL LETTER KEN
0570	0540	ARMENIAN SMALL LETTER HO	ARMENIAN CAPITAL LETTER HO
0571	0541	ARMENIAN SMALL LETTER JA	ARMENIAN CAPITAL LETTER JA
0572	0542	ARMENIAN SMALL LETTER GHAD	ARMENIAN CAPITAL LETTER GHAD
0573	0543	ARMENIAN SMALL LETTER CHEH	ARMENIAN CAPITAL LETTER CHEH
0574	0544	ARMENIAN SMALL LETTER MEN	ARMENIAN CAPITAL LETTER MEN

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
0575	0545	ARMENIAN SMALL LETTER YI	ARMENIAN CAPITAL LETTER YI
0576	0546	ARMENIAN SMALL LETTER NOW	ARMENIAN CAPITAL LETTER NOW
0577	0547	ARMENIAN SMALL LETTER SNA	ARMENIAN CAPITAL LETTER SHA
0578	0548	ARMENIAN SMALL LETTER VO	ARMENIAN CAPITAL LETTER VO
0579	0549	ARMENIAN SMALL LETTER CHA	ARMENIAN CAPITAL LETTER CHA
057A	054A	ARMENIAN SMALL LETTER PEH	ARMENIAN CAPITAL LETTER PEH
057B	054B	ARMENIAN SMALL LETTER JHEH	ARMENIAN CAPITAL LETTER JHEH
057C	054C	ARMENIAN SMALL LETTER RA	ARMENIAN CAPITAL LETTER RA
057D	054D	ARMENIAN SMALL LETTER SEH	ARMENIAN CAPITAL LETTER SEH
057E	054E	ARMENIAN SMALL LETTER VEW	ARMENIAN CAPITAL LETTER VEW
057F	054F	ARMENIAN SMALL LETTER TIWN	ARMENIAN CAPITAL LETTER TIWN
0580	0550	ARMENIAN SMALL LETTER REH	ARMENIAN CAPITAL LETTER REH
0581	0551	ARMENIAN SMALL LETTER CO	ARMENIAN CAPITAL LETTER CO
0582	0552	ARMENIAN SMALL LETTER YIWN	ARMENIAN CAPITAL LETTER YIWN
0583	0553	ARMENIAN SMALL LETTER PIWP	ARMENIAN CAPITAL LETTER PIWR
0584	0554	ARMENIAN SMALL LETTER KEH	ARMENIAN CAPITAL LETTER KEH
0585	0555	ARMENIAN SMALL LETTER OH	ARMENIAN CAPITAL LETTER OH
0586	0556	ARMENIAN SMALL LETTER FEH	ARMENIAN CAPITAL LETTER FEH
10D0	10A0	GEORGIAN LETTER AN	GEORGIAN CAPITAL LETTER AN (KHUTSURI)
10D1	10A1	GEORGIAN LETTER BAN	GEORGIAN CAPITAL LETTER BAN (KHUTSURI)
10D2	10A2	GEORGIAN LETTER GAN	GEORGIAN CAPITAL LETTER GAN (KHUTSURI)
10D3	10A3	GEORGIAN LETTER DON	GEORGIAN CAPITAL LETTER DON (KHUTSURI)
10D4	10A4	GEORGIAN LETTER EN	GEORGIAN CAPITAL LETTER EN (KHUTSURI)
10D5	10A5	GEORGIAN LETTER VIN	GEORGIAN CAPITAL LETTER VIN (KHUTSURI)
10D6	10A6	GEORGIAN LETTER ZEN	GEORGIAN CAPITAL LETTER ZEN (KHUTSURI)
10D7	10A7	GEORGIAN LETTER TAN	GEORGIAN CAPITAL LETTER TAN (KHUTSURI)
10D8	10A8	GEORGIAN LETTER IN	GEORGIAN CAPITAL LETTER IN (KHUTSURI)
10D9	10A9	GEORGIAN LETTER KAN	GEORGIAN CAPITAL LETTER KAN (KHUTSURI)
10DA	10AA	GEORGIAN LETTER LAS	GEORGIAN CAPITAL LETTER LAS (KHUTSURI)
10DB	10AB	GEORGIAN LETTER MAN	GEORGIAN CAPITAL LETTER MAN (KHUTSURI)
10DC	10AC	GEORGIAN LETTER NAR	GEORGIAN CAPITAL LETTER NAR (KHUTSURI)
10DD	10AD	GEORGIAN LETTER ON	GEORGIAN CAPITAL LETTER ON (KHUTSURI)

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
10DE	10AE	GEORGIAN LETTER PAR	GEORGIAN CAPITAL LETTER PAR (KHUTSURI)
10DF	10AF	GEORGIAN LETTER ZHAR	GEORGIAN CAPITAL LETTER ZHAR (KHUTSURI)
10E0	10B0	GEORGIAN LETTER RAE	GEORGIAN CAPITAL LETTER RAE (KHUTSURI)
10E1	10B1	GEORGIAN LETTER SAN	GEORGIAN CAPITAL LETTER SAN (KHUTSURI)
10E2	10B2	GEORGIAN LETTER TAR	GEORGIAN CAPITAL LETTER TAR (KHUTSURI)
10E3	10B3	GEORGIAN LETTER UN	GEORGIAN CAPITAL LETTER UN (KHUTSURI)
10E4	10B4	GEORGIAN LETTER PHAR	GEORGIAN CAPITAL LETTER PHAR (KHUTSURI)
10E5	10B5	GEORGIAN LETTER KHAR	GEORGIAN CAPITAL LETTER KHAR (KHUTSURI)
10E6	10B6	GEORGIAN LETTER GHAN	GEORGIAN CAPITAL LETTER GHAN (KHUTSURI)
10E7	10B7	GEORGIAN LETTER QAR	GEORGIAN CAPITAL LETTER QAR (KHUTSURI)
10E8	10B8	GEORGIAN LETTER SHIN	GEORGIAN CAPITAL LETTER SHIN (KHUTSURI)
10E9	10B9	GEORGIAN LETTER CHIN	GEORGIAN CAPITAL LETTER CHIN (KHUTSURI)
10EA	10BA	GEORGIAN LETTER CAN	GEORGIAN CAPITAL LETTER CAN (KHUTSURI)
10EB	10BB	GEORGIAN LETTER JIL	GEORGIAN CAPITAL LETTER JIL (KHUTSURI)
10EC	10BC	GEORGIAN LETTER CIL	GEORGIAN CAPITAL LETTER CIL (KHUTSURI)
10ED	10BD	GEORGIAN LETTER CHAR	GEORGIAN CAPITAL LETTER CHAR (KHUTSURI)
10EE	10BE	GEORGIAN LETTER XAN	GEORGIAN CAPITAL LETTER XAN (KHUTSURI)
10EF	10BF	GEORGIAN LETTER JHAN	GEORGIAN CAPITAL LETTER JHAN (KHUTSURI)
10F0	10C0	GEORGIAN LETTER HAE	GEORGIAN CAPITAL LETTER HAE (KHUTSURI)
10F1	10C1	GEORGIAN LETTER HE	GEORGIAN CAPITAL LETTER HE (KHUTSURI)
10F2	10C2	GEORGIAN LETTER HIE	GEORGIAN CAPITAL LETTER HIE (KHUTSURI)
10F3	10C3	GEORGIAN LETTER WE	GEORGIAN CAPITAL LETTER WE (KHUTSURI)
10F4	10C4	GEORGIAN LETTER HAR	GEORGIAN CAPITAL LETTER HAR (KHUTSURI)
10F5	10C5	GEORGIAN LETTER HOE	GEORGIAN CAPITAL LETTER HOE (KHUTSURI)
1E01	1E00	LATIN SMALL LETTER A WITH RING BELOW	LATIN CAPITAL LETTER A WITH RING BELOW
1E03	1E02	LATIN SMALL LETTER B WITH DOT ABOVE	LATIN CAPITAL LETTER B WITH DOT ABOVE
1E05	1E04	LATIN SMALL LETTER B WITH DOT BELOW	LATIN CAPITAL LETTER B WITH DOT BELOW
1E07	1E06	LATIN SMALL LETTER B WITH LINE BELOW	LATIN CAPITAL LETTER B WITH LINE BELOW
1E09	1E08	LATIN SMALL LETTER C WITH CEDILLA AND ACUTE	LATIN CAPITAL LETTER C WITH CEDILLA AND ACUTE
1E0B	1E0A	LATIN SMALL LETTER D WITH DOT ABOVE	LATIN CAPITAL LETTER D WITH DOT ABOVE

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1E0D	1E0C	LATIN SMALL LETTER D WITH DOT BELOW	LATIN CAPITAL LETTER D WITH DOT BELOW
1E0F	1E0E	LATIN SMALL LETTER D WITH LINE BELOW	LATIN CAPITAL LETTER D WITH LINE BELOW
1E11	1E10	LATIN SMALL LETTER D WITH CEDILLA	LATIN CAPITAL LETTER D WITH CEDILLA
1E13	1E12	LATIN SMALL LETTER D WITH CIRCUMFLEX BELOW	LATIN CAPITAL LETTER D WITH CIRCUMFLEX BELOW
1E15	1E14	LATIN SMALL LETTER E WITH MACRON AND GRAVE	LATIN CAPITAL LETTER E WITH MACRON AND GRAVE
1E17	1E16	LATIN SMALL LETTER E WITH MACRON AND ACUTE	LATIN CAPITAL LETTER E WITH MACRON AND ACUTE
1E19	1E18	LATIN SMALL LETTER E WITH CIRCUMFLEX BELOW	LATIN CAPITAL LETTER E WITH CIRCUMFLEX BELOW
1E1B	1E1A	LATIN SMALL LETTER E WITH TILDE BELOW	LATIN CAPITAL LETTER E WITH TILDE BELOW
1E1D	1E1C	LATIN SMALL LETTER E WITH CEDILLA AND BREVE	LATIN CAPITAL LETTER E WITH CEDILLA AND BREVE
1E1F	1E1E	LATIN SMALL LETTER F WITH DOT ABOVE	LATIN CAPITAL LETTER F WITH DOT ABOVE
1E21	1E20	LATIN SMALL LETTER G WITH MACRON	LATIN CAPITAL LETTER G WITH MACRON
1E23	1E22	LATIN SMALL LETTER H WITH DOT ABOVE	LATIN CAPITAL LETTER H WITH DOT ABOVE
1E25	1E24	LATIN SMALL LETTER H WITH DOT BELOW	LATIN CAPITAL LETTER H WITH DOT BELOW
1E27	1E26	LATIN SMALL LETTER H WITH DIAERESIS	LATIN CAPITAL LETTER H WITH DIAERESIS
1E29	1E28	LATIN SMALL LETTER H WITH CEDILLA	LATIN CAPITAL LETTER H WITH CEDILLA
1E2B	1E2A	LATIN SMALL LETTER H WITH BREVE BELOW	LATIN CAPITAL LETTER H WITH BREVE BELOW
1E2D	1E2C	LATIN SMALL LETTER I WITH TILDE BELOW	LATIN CAPITAL LETTER I WITH TILDE BELOW
1E2F	1E2E	LATIN SMALL LETTER I WITH DIAERESIS AND ACUTE	LATIN CAPITAL LETTER I WITH DIAERESIS AND ACUTE
1E31	1E30	LATIN SMALL LETTER K WITH ACUTE	LATIN CAPITAL LETTER K WITH ACUTE
1E33	1E32	LATIN SMALL LETTER K WITH DOT BELOW	LATIN CAPITAL LETTER K WITH DOT BELOW
1E35	1E34	LATIN SMALL LETTER K WITH LINE BELOW	LATIN CAPITAL LETTER K WITH LINE BELOW
1E37	1E36	LATIN SMALL LETTER L WITH DOT BELOW	LATIN CAPITAL LETTER L WITH DOT BELOW
1E39	1E38	LATIN SMALL LETTER L WITH DOT BELOW AND MACRON	LATIN CAPITAL LETTER L WITH DOT BELOW AND MACRON
1E3B	1E3A	LATIN SMALL LETTER L WITH LINE BELOW	LATIN CAPITAL LETTER L WITH LINE BELOW
1E3D	1E3C	LATIN SMALL LETTER L WITH CIRCUMFLEX BELOW	LATIN CAPITAL LETTER L WITH CIRCUMFLEX BELOW
1E3F	1E3E	LATIN SMALL LETTER M WITH ACUTE	LATIN CAPITAL LETTER M WITH ACUTE
1E41	1E40	LATIN SMALL LETTER M WITH DOT ABOVE	LATIN CAPITAL LETTER M WITH DOT ABOVE
1E43	1E42	LATIN SMALL LETTER M WITH DOT BELOW	LATIN CAPITAL LETTER M WITH DOT BELOW
1E45	1E44	LATIN SMALL LETTER N WITH DOT ABOVE	LATIN CAPITAL LETTER N WITH DOT ABOVE
1E47	1E46	LATIN SMALL LETTER N WITH DOT BELOW	LATIN CAPITAL LETTER N WITH DOT BELOW

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1E49	1E48	LATIN SMALL LETTER N WITH LINE BELOW	LATIN CAPITAL LETTER N WITH LINE BELOW
1E4B	1E4A	LATIN SMALL LETTER N WITH CIRCUMFLEX BELOW	LATIN CAPITAL LETTER N WITH CIRCUMFLEX BELOW
1E4D	1E4C	LATIN SMALL LETTER O WITH TILDE AND ACUTE	LATIN CAPITAL LETTER O WITH TILDE AND ACUTE
1E4F	1E4E	LATIN SMALL LETTER O WITH TILDE AND DIAERESIS	LATIN CAPITAL LETTER O WITH TILDE AND DIAERESIS
1E51	1E50	LATIN SMALL LETTER O WITH MACRON AND GRAVE	LATIN CAPITAL LETTER O WITH MACRON AND GRAVE
1E53	1E52	LATIN SMALL LETTER O WITH MACRON AND ACUTE	LATIN CAPITAL LETTER O WITH MACRON AND ACUTE
1E55	1E54	LATIN SMALL LETTER P WITH ACUTE	LATIN CAPITAL LETTER P WITH ACUTE
1E57	1E56	LATIN SMALL LETTER P WITH DOT ABOVE	LATIN CAPITAL LETTER P WITH DOT ABOVE
1E59	1E58	LATIN SMALL LETTER R WITH DOT ABOVE	LATIN CAPITAL LETTER R WITH DOT ABOVE
1E5B	1E5A	LATIN SMALL LETTER R WITH DOT BELOW	LATIN CAPITAL LETTER R WITH DOT BELOW
1E5D	1E5C	LATIN SMALL LETTER R WITH DOT BELOW AND MACRON	LATIN CAPITAL LETTER R WITH DOT BELOW AND MACRON
1E5F	1E5E	LATIN SMALL LETTER R WITH LINE BELOW	LATIN CAPITAL LETTER R WITH LINE BELOW
1E61	1E60	LATIN SMALL LETTER S WITH DOT ABOVE	LATIN CAPITAL LETTER S WITH DOT ABOVE
1E63	1E62	LATIN SMALL LETTER S WITH DOT BELOW	LATIN CAPITAL LETTER S WITH DOT BELOW
1E65	1E64	LATIN SMALL LETTER S WITH ACUTE AND DOT ABOVE	LATIN CAPITAL LETTER S WITH ACUTE AND DOT ABOVE
1E67	1E66	LATIN SMALL LETTER S WITH CARON AND DOT ABOVE	LATIN CAPITAL LETTER S WITH CARON AND DOT ABOVE
1E69	1E68	LATIN SMALL LETTER S WITH DOT BELOW AND DOT ABOVE	LATIN CAPITAL LETTER S WITH DOT BELOW AND DOT ABOVE
1E6B	1E6A	LATIN SMALL LETTER T WITH DOT ABOVE	LATIN CAPITAL LETTER T WITH DOT ABOVE
1E6D	1E6C	LATIN SMALL LETTER T WITH DOT BELOW	LATIN CAPITAL LETTER T WITH DOT BELOW
1E6F	1E6E	LATIN SMALL LETTER T WITH LINE BELOW	LATIN CAPITAL LETTER T WITH LINE BELOW
1E71	1E70	LATIN SMALL LETTER T WITH CIRCUMFLEX BELOW	LATIN CAPITAL LETTER T WITH CIRCUMFLEX BELOW
1E73	1E72	LATIN SMALL LETTER U WITH DIAERESIS BELOW	LATIN CAPITAL LETTER U WITH DIAERESIS BELOW
1E75	1E74	LATIN SMALL LETTER U WITH TILDE BELOW	LATIN CAPITAL LETTER U WITH TILDE BELOW
1E77	1E76	LATIN SMALL LETTER U WITH CIRCUMFLEX BELOW	LATIN CAPITAL LETTER U WITH CIRCUMFLEX BELOW
1E79	1E78	LATIN SMALL LETTER U WITH TILDE AND ACUTE	LATIN CAPITAL LETTER U WITH TILDE AND ACUTE
1E7B	1E7A	LATIN SMALL LETTER U WITH MACRON AND DIAERESIS	LATIN CAPITAL LETTER U WITH MACRON AND DIAERESIS
1E7D	1E7C	LATIN SMALL LETTER V WITH TILDE	LATIN CAPITAL LETTER V WITH TILDE
1E7F	1E7E	LATIN SMALL LETTER V WITH DOT BELOW	LATIN CAPITAL LETTER V WITH DOT BELOW
1E81	1E80	LATIN SMALL LETTER W WITH GRAVE	LATIN CAPITAL LETTER W WITH GRAVE
1E83	1E82	LATIN SMALL LETTER W WITH ACUTE	LATIN CAPITAL LETTER W WITH ACUTE

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1E85	1E84	LATIN SMALL LETTER W WITH DIAERESIS	LATIN CAPITAL LETTER W WITH DIAERESIS
1E87	1E86	LATIN SMALL LETTER W WITH DOT ABOVE	LATIN CAPITAL LETTER W WITH DOT ABOVE
1E89	1E88	LATIN SMALL LETTER W WITH DOT BELOW	LATIN CAPITAL LETTER W WITH DOT BELOW
1E8B	1E8A	LATIN SMALL LETTER X WITH DOT ABOVE	LATIN CAPITAL LETTER X WITH DOT ABOVE
1E8D	1E8C	LATIN SMALL LETTER X WITH DIAERESIS	LATIN CAPITAL LETTER X5 WITH DIAERESIS
1E8F	1E8E	LATIN SMALL LETTER Y WITH DOT ABOVE	LATIN CAPITAL LETTER Y WITH DOT ABOVE
1E91	1E90	LATIN SMALL LETTER Z WITH CIRCUMFLEX	LATIN CAPITAL LETTER Z WITH CIRCUMFLEX
1E93	1E92	LATIN SMALL LETTER Z WITH DOT BELOW	LATIN CAPITAL LETTER Z WITH DOT BELOW
1E95	1E94	LATIN SMALL LETTER Z WITH LINE BELOW	LATIN CAPITAL LETTER Z WITH LINE BELOW
1EA1	1EA0	LATIN SMALL LETTER A WITH DOT BELOW	LATIN CAPITAL LETTER A WITH DOT BELOW
1EA3	1EA2	LATIN SMALL LETTER A WITH HOOK ABOVE	LATIN CAPITAL LETTER A WITH HOOK ABOVE
1EA5	1EA4	LATIN SMALL LETTER A WITH CIRCUMFLEX AND ACUTE	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND ACUTE
1EA7	1EA6	LATIN SMALL LETTER A WITH CIRCUMFLEX AND GRAVE	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND GRAVE
1EA9	1EA8	LATIN SMALL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND HOOK ABOVE
1EAB	1EAA	LATIN SMALL LETTER A WITH CIRCUMFLEX AND TILDE	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND TILDE
1EAD	1EAC	LATIN SMALL LETTER A WITH CIRCUMFLEX AND DOT BELOW	LATIN CAPITAL LETTER A WITH CIRCUMFLEX AND DOT BELOW
1EAF	1EAE	LATIN SMALL LETTER A WITH BREVE AND ACUTE	LATIN CAPITAL LETTER A WITH BREVE AND ACUTE
1EB1	1EB0	LATIN SMALL LETTER A WITH BREVE AND GRAVE	LATIN CAPITAL LETTER A WITH BREVE AND GRAVE
1EB3	1EB2	LATIN SMALL LETTER A WITH BREVE AND HOOK ABOVE	LATIN CAPITAL LETTER A WITH BREVE AND HOOK ABOVE
1EB5	1EB4	LATIN SMALL LETTER A WITH BREVE AND TILDE	LATIN CAPITAL LETTER A WITH BREVE AND TILDE
1EB7	1EB6	LATIN SMALL LETTER A WITH BREVE AND DOT BELOW	LATIN CAPITAL LETTER A WITH BREVE AND DOT BELOW
1EB9	1EB8	LATIN SMALL LETTER E WITH DOT BELOW	LATIN CAPITAL LETTER E WITH DOT BELOW
1EBB	1EBA	LATIN SMALL LETTER E WITH HOOK ABOVE	LATIN CAPITAL LETTER E WITH HOOK ABOVE
1EBD	1EBC	LATIN SMALL LETTER E WITH TILDE	LATIN CAPITAL LETTER E WITH TILDE
1EBF	1EBE	LATIN SMALL LETTER E WITH CIRCUMFLEX AND ACUTE	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND ACUTE
1EC1	1EC0	LATIN SMALL LETTER E WITH CIRCUMFLEX AND GRAVE	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND GRAVE
1EC3	1EC2	LATIN SMALL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND HOOK ABOVE
1EC5	1EC4	LATIN SMALL LETTER E WITH CIRCUMFLEX AND TILDE	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND TILDE
1EC7	1EC6	LATIN SMALL LETTER E WITH CIRCUMFLEX AND DOT BELOW	LATIN CAPITAL LETTER E WITH CIRCUMFLEX AND DOT BELOW
1EC9	1EC8	LATIN SMALL LETTER I WITH HOOK ABOVE	LATIN CAPITAL LETTER I WITH HOOK ABOVE

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1ECB	1ECA	LATIN SMALL LETTER I WITH DOT BELOW	LATIN CAPITAL LETTER I WITH DOT BELOW
1ECD	1ECC	LATIN SMALL LETTER O WITH DOT BELOW	LATIN CAPITAL LETTER O WITH DOT BELOW
1ECF	1ECE	LATIN SMALL LETTER O WITH HOOK ABOVE	LATIN CAPITAL LETTER O WITH HOOK ABOVE
1ED1	1ED0	LATIN SMALL LETTER O WITH CIRCUMFLEX AND ACUTE	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND ACUTE
1ED3	1ED2	LATIN SMALL LETTER O WITH CIRCUMFLEX AND GRAVE	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND GRAVE
1ED5	1ED4	LATIN SMALL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND HOOK ABOVE
1ED7	1ED6	LATIN SMALL LETTER O WITH CIRCUMFLEX AND TILDE	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND TILDE
1ED9	1ED8	LATIN SMALL LETTER O WITH CIRCUMFLEX AND DOT BELOW	LATIN CAPITAL LETTER O WITH CIRCUMFLEX AND DOT BELOW
1EDB	1EDA	LATIN SMALL LETTER O WITH HORN AND ACUTE	LATIN CAPITAL LETTER O WITH HORN AND ACUTE
1EDD	1EDC	LATIN SMALL LETTER O WITH HORN AND GRAVE	LATIN CAPITAL LETTER O WITH HORN AND GRAVE
1EDF	1EDE	LATIN SMALL LETTER O WITH HORN AND HOOK ABOVE	LATIN CAPITAL LETTER O WITH HORN AND HOOK ABOVE
1EE1	1EE0	LATIN SMALL LETTER O WITH HORN AND TILDE	LATIN CAPITAL LETTER O WITH HORN AND TILDE
1EE3	1EE2	LATIN SMALL LETTER O WITH HORN AND DOT BELOW	LATIN CAPITAL LETTER O WITH HORN AND DOT BELOW
1EE5	1EE4	LATIN SMALL LETTER U WITH DOT BELOW	LATIN CAPITAL LETTER U WITH DOT BELOW
1EE7	1EE6	LATIN SMALL LETTER U WITH HOOK ABOVE	LATIN CAPITAL LETTER U WITH HOOK ABOVE
1EE9	1EE8	LATIN SMALL LETTER U WITH HORN AND ACUTE	LATIN CAPITAL LETTER U WITH HORN AND ACUTE
1EEB	1EEA	LATIN SMALL LETTER U WITH HORN AND GRAVE	LATIN CAPITAL LETTER U WITH HORN AND GRAVE
1EED	1EEC	LATIN SMALL LETTER U WITH HORN AND HOCK ABOVE	LATIN CAPITAL LETTER U WITH HORN AND HOOK ABOVE
1EEF	1EEE	LATIN SMALL LETTER U WITH HORN AND TILDE	LATIN CAPITAL LETTER U WITH HORN AND TILDE
1EF1	1EF0	LATIN SMALL LETTER U WITH HORN AND DOT BELOW	LATIN CAPITAL LETTER U WITH HORN AND DOT BELOW
1EF3	1EF2	LATIN SMALL LETTER Y WITH GRAVE	LATIN CAPITAL LETTER Y WITH GRAVE
1EF5	1EF4	LATIN SMALL LETTER Y WITH DOT BELOW	LATIN CAPITAL LETTER Y WITH DOT BELOW
1EF7	1EF6	LATIN SMALL LETTER Y WITH HOOK ABOVE	LATIN CAPITAL LETTER Y WITH HOOK ABOVE
1EF9	1EF8	LATIN SMALL LETTER Y WITH TILDE	LATIN CAPITAL LETTER Y WITH TILDE
1F00	1F08	GREEK SMALL LETTER ALPHA WITH PSILI	GREEK CAPITAL LETTER ALPHA WITH PSILI
1F01	1F09	GREEK SMALL LETTER ALPHA WITH DASIA	GREEK CAPITAL LETTER ALPHA WITH DASIA
1F02	1F0A	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA
1F03	1F0B	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA
1F04	1F0C	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA
1F05	1F0D	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1F06	1F0E	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI
1F07	1F0F	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI
1F10	1F18	GREEK SMALL LETTER EPSILON WITH PSILI	GREEK CAPITAL LETTER EPSILON WITH PSILI
1F11	1F19	GREEK SMALL LETTER EPSILON WITH DASIA	GREEK CAPITAL LETTER EPSILON WITH DASIA
1F12	1F1A	GREEK SMALL LETTER EPSILON WITH PSILI AND VARIA	GREEK CAPITAL LETTER EPSILON WITH PSILI AND VARIA
1F13	1F1B	GREEK SMALL LETTER EPSILON WITH DASIA AND VARIA	GREEK CAPITAL LETTER EPSILON WITH DASIA AND VARIA
1F14	1F1C	GREEK SMALL LETTER EPSILON WITH PSILI AND OXIA	GREEK CAPITAL LETTER EPSILON WITH PSILI AND OXIA
1F15	1F1D	GREEK SMALL LETTER EPSILON WITH DASIA AND OXIA	GREEK CAPITAL LETTER EPSILON WITH DASIA AND OXIA
1F20	1F28	GREEK SMALL LETTER ETA WITH PSILI	GREEK CAPITAL LETTER ETA WITH PSILI
1F21	1F29	GREEK SMALL LETTER ETA WITH DASIA	GREEK CAPITAL LETTER ETA WITH DASIA
1F22	1F2A	GREEK SMALL LETTER ETA WITH PSILI AND VARIA	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA
1F23	1F2B	GREEK SMALL LETTER ETA WITH DASIA AND VARIA	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA
1F24	1F2C	GREEK SMALL LETTER ETA WITH PSILI AND OXIA	GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA
1F25	1F2D	GREEK SMALL LETTER ETA WITH DASIA AND OXIA	GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA
1F26	1F2E	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI	GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI
1F27	1F2F	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI	GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI
1F30	1F38	GREEK SMALL LETTER IOTA WITH PSILI	GREEK CAPITAL LETTER IOTA WITH PSILI
1F31	1F39	GREEK SMALL LETTER IOTA WITH DASIA	GREEK CAPITAL LETTER IOTA WITH DASIA
1F32	1F3A	GREEK SMALL LETTER IOTA WITH PSILI AND VARIA	GREEK CAPITAL LETTER IOTA WITH PSILI AND VARIA
1F33	1F3B	GREEK SMALL LETTER IOTA WITH DASIA AND VARIA	GREEK CAPITAL LETTER IOTA WITH DASIA AND VARIA
1F34	1F3C	GREEK SMALL LETTER IOTA WITH PSILI AND OXIA	GREEK CAPITAL LETTER IOTA WITH PSILI AND OXIA
1F35	1F3D	GREEK SMALL LETTER IOTA WITH DASIA AND OXIA	GREEK CAPITAL LETTER IOTA WITH DASIA AND OXIA
1F36	1F3E	GREEK SMALL LETTER IOTA WITH PSILI AND PERISPOMENI	GREEK CAPITAL LETTER IOTA WITH PSILI AND PERISPOMENI
1F37	1F3F	GREEK SMALL LETTER IOTA WITH DASIA AND PERISPOMENI	GREEK CAPITAL LETTER IOTA WITH DASIA AND PERISPOMENI
1F40	1F48	GREEK SMALL LETTER OMICRON WITH PSILI	GREEK CAPITAL LETTER OMICRON WITH PSILI
1F41	1F49	GREEK SMALL LETTER OMICRON WITH DASIA	GREEK CAPITAL LETTER OMICRON WITH DASIA
1F42	1F4A	GREEK SMALL LETTER OMICRON WITH PSILI AND VARIA	GREEK CAPITAL LETTER OMICRON WITH PSILI AND VARIA
1F43	1F4B	GREEK SMALL LETTER OMICRON WITH DASIA AND VARIA	GREEK CAPITAL LETTER OMICRON WITH DASIA AND VARIA

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1F44	1F4C	GREEK SMALL LETTER OMICRON WITH PSILI AND OXIA	GREEK CAPITAL LETTER OMICRON WITH PSILI AND OXIA
1F45	1F4D	GREEK SMALL LETTER OMICRON WITH DASIA AND OXIA	GREEK CAPITAL LETTER OMICRON WITH DASIA AND OXIA
1F51	1F59	GREEK SMALL LETTER UPSILON WITH DASIA	GREEK CAPITAL LETTER UPSILON WITH OASIS
1F53	1F5B	GREEK SMALL LETTER UPSILON WITH DASIA AND VARIA	GREEK CAPITAL LETTER UPSILON WITH DASIA AND VARIA
1F55	1F5D	GREEK SMALL LETTER UPSILON WITH DASIA AND OXIA	GREEK CAPITAL LETTER UPSILON WITH DASIA AND OXIA
1F57	1F5F	GREEK SMALL LETTER UPSILON WITH DASIA AND PERISPOMENI	GREEK CAPITAL LETTER UPSILON WITH DASIA AND PERISPOMENI
1F60	1F68	GREEK SMALL LETTER OMEGA WITh PSILI	GREEK CAPITAL LETTER OMEGA WITH PSILI
1F61	1F69	GREEK SMALL LETTER OMEGA WITH DASIA	GREEK CAPITAL LETTER OMEGA WITH DASIA
1F62	1F6A	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA	GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA
1F63	1F6B	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA	GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA
1F64	1F6C	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA	GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA
1F65	1F6D	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA	GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA
1F66	1F6E	GREEK SMALL LETTER OMEGA WITH PSILI AND PERISPOMENI	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI
1F67	1F6F	GREEK SMALL LETTER OMEGA WITH DASIA AND PERISPOMENI	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PERISPOMENI
1F80	1F88	GREEK SMALL LETTER ALPHA WITH PSILI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITh PSILI AND PROSGEGRAMMENI
1F81	1F89	GREEK SMALL LETTER ALPHA WITH DASIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PROSGEGRAMMENI
1F82	1F8A	GREEK SMALL LETTER ALPHA WITH PSILI AND VARIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH PSILI AND VARIA AND PROSGEGRAMMENI
1F83	1F8B	GREEK SMALL LETTER ALPHA WITH DASIA AND VARIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH DASIA AND VARIA AND PROSGEGRAMMENI
1F84	1F8C	GREEK SMALL LETTER ALPHA WITH PSILI AND OXIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH PSILI AND OXIA AND PROSGEGRAMMEN
1F85	1F8D	GREEK SMALL LETTER ALPHA WITH DASIA AND OXIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH DASIA AND OXIA AND PROSGEGRAMMEN
1F86	1F8E	GREEK SMALL LETTER ALPHA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
1F87	1F8F	GREEK SMALL LETTER ALPHA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ALPHA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
1F90	1F98	GREEK SMALL LETTER ETA WITH PSILI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH PSILI AND PROSGEGRAMMENI
1F91	1F99	GREEK SMALL LETTER ETA WITH DASIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH DASIA AND PROSGEGRAMMENI

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
1F92	1F9A	GREEK SMALL LETTER ETA WITH PSILI AND VARIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH PSILI AND VARIA AND PROSGEGRAMMENI
1F93	1F9B	GREEK SMALL LETTER ETA WITH DASIA AND VARIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH DASIA AND VARIA AND PROSGEGRAMMENI
1F94	1F9C	GREEK SMALL LETTER ETA WITH PSILI AND OXIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH PSILI AND OXIA AND PROSGEGRAMMENI
1F95	1F9D	GREEK SMALL LETTER ETA WITH DASIA AND OXIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH DASIA AND OXIA AND PROSGEGRAMMENI
1F96	1F9E	GREEK SMALL LETTER ETA WITH PSILI AND PERISPOMENI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
1F97	1F9F	GREEK SMALL LETTER ETA WITH DASIA AND PERISPOMENI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER ETA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
1FA0	1FA8	GREEK SMALL LETTER OMEGA WITH PSILI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PROSGEGRAMMENI
1FA1	1FA9	GREEK SMALL LETTER OMEGA WITH DASIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH DASIA AND PROSGEGRAMMENI
1FA2	1FAA	GREEK SMALL LETTER OMEGA WITH PSILI AND VARIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH PSILI AND VARIA AND PROSGEGRAMMENI
1FA3	1FAB	GREEK SMALL LETTER OMEGA WITH DASIA AND VARIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH DASIA AND VARIA AND PROSGEGRAMMENI
1FA4	1FAC	GREEK SMALL LETTER OMEGA WITH PSILI AND OXIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH PSILI AND OXIA AND PROSGEGRAMMENI
1FA5	1FAD	GREEK SMALL LETTER OMEGA WITH DASIA AND OXIA AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH DASIA AND OXIA AND PROSGEGRAMMENI
1FA6	1FAE	GREEK SMALL LETTER OMEGA WITh PSILI AND PERISPOMENI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMEGA WITH PSILI AND PERISPOMENI AND PROSGEGRAMMENI
1FA7	1FAF	GREEK SMALL LETTER OMEGA WITH DASIA AND PEPISPOMENI AND YPOGEGRAMMENI	GREEK CAPITAL LETTER OMECA WITH DASIA AND PERISPOMENI AND PROSGEGRAMMENI
1FB0	1FB8	GREEK SMALL LETTER ALPHA WITH VRACHY	GREEK CAPITAL LETTER ALPHA WITH VRACHY
1FB1	1FB9	GREEK SMALL LETTER ALPHA WITH MACRON	GREEK CAPITAL LETTER ALPHA WITH MACRON
1FD0	1FD8	GREEK SMALL LETTER IOTA WITH VRACHY	GREEK CAPITAL LETTER IOTA WITH VRACHY
1FD1	1FD9	GREEK SMALL LETTER IOTA WITH MACRON	GREEK CAPITAL LETTER IOTA WITH MACRON
1FE0	1FE8	GREEK SMALL LETTER UPSILON WITH VRACHY	GREEK CAPITAL LETTER UPSILON WITH VRACHY
1FE1	1FE9	GREEK SMALL LETTER UPSILON WITH MACRON	GREEK CAPITAL LETTER UPSILON WITH MACRON
24D0	24B6	CIRCLED LATIN SMALL LETTER A	CIRCLED LATIN CAPITAL LETTER A
24D1	24B7	CIRCLED LATIN SMALL LETTER B	CIRCLED LATIN CAPITAL LETTER B
24D2	24B8	CIRCLED LATIN SMALL LETTER C	CIRCLED LATIN CAPITAL LETTER C
24D3	24B9	CIRCLED LATIN SMALL LETTER D	CIRCLED LATIN CAPITAL LETTER D
24D4	24BA	CIRCLED LATIN SMALL LETTER E	CIRCLED LATIN CAPITAL LETTER E

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
24D5	24BB	CIRCLED LATIN SMALL LETTER F	CIRCLED LATIN CAPITAL LETTER F
24D6	24BC	CIRCLED LATIN SMALL LETTER G	CIRCLED LATIN CAPITAL LETTER G
24D7	24BD	CIRCLED LATIN SMALL LETTER H	CIRCLED LATIN CAPITAL LETTER H
24D8	24BE	CIRCLED LATIN SMALL LETTER I	CIRCLED LATIN CAPITAL LETTER I
24D9	24BF	CIRCLED LATIN SMALL LETTER J	CIRCLED LATIN CAPITAL LETTER J
24DA	24C0	CIRCLED LATIN SMALL LETTER K	CIRCLED LATIN CAPITAL LETTER K
24DB	24C1	CIRCLED LATIN SMALL LETTER L	CIRCLED LATIN CAPITAL LETTER L
24DC	24C2	CIRCLED LATIN SMALL LETTER M	CIRCLED LATIN CAPITAL LETTER M
24DD	24C3	CIRCLED LATIN SMALL LETTER N	CIRCLED LATIN CAPITAL LETTER N
24DE	24C4	CIRCLED LATIN SMALL LETTER O	CIRCLED LATIN CAPITAL LETTER O
24DF	24C5	CIRCLED LATIN SMALL LETTER P	CIRCLED LATIN CAPITAL LETTER P
24E0	24C6	CIRCLED LATIN SMALL LETTER Q	CIRCLED LATIN CAPITAL LETTER Q
24E1	24C7	CIRCLED LATIN SMALL LETTER R	CIRCLED LATIN CAPITAL LETTER R
24E2	24C8	CIRCLED LATIN SMALL LETTER S	CIRCLED LATIN CAPITAL LETTER S
24E3	24C9	CIRCLED LATIN SMALL LETTER T	CIRCLED LATIN CAPITAL LETTER T
24E4	24CA	CIRCLED LATIN SMALL LETTER U	CIRCLED LATIN CAPITAL LETTER U
24E5	24CB	CIRCLED LATIN SMALL LETTER V	CIRCLED LATIN CAPITAL LETTER V
24E6	24CC	CIRCLED LATIN SMALL LETTER W	CIRCLED LATIN CAPITAL LETTER W
24E7	24CD	CIRCLED LATIN SMALL LETTER X	CIRCLED LATIN CAPITAL LETTER X
24E8	24CE	CIRCLED LATIN SMALL LETTER Y	CIRCLED LATIN CAPITAL LETTER Y
24E9	24CF	CIRCLED LATIN SMALL LETTER Z	CIRCLED LATIN CAPITAL LETTER Z
FF41	FF21	FULLWIDTH LATIN SMALL LETTER A	FULLWIDTH LATIN CAPITAL LETTER A
FF42	FF22	FULLWIDTH LATIN SMALL LETTER B	FULLWIDTH LATIN CAPITAL LETTER B
FF43	FF23	FULLWIDTH LATIN SMALL LETTER C	FULLWIDTH LATIN CAPITAL LETTER C
FF44	FF24	FULLWIDTH LATIN SMALL LETTER D	FULLWIDTH LATIN CAPITAL LETTER D
FF45	FF25	FULLWIDTH LATIN SMALL LETTER E	FULLWIDTH LATIN CAPITAL LETTER E
FF46	FF26	FULLWIDTH LATIN SMALL LETTER F	FULLWIDTH LATIN CAPITAL LETTER F
FF47	FF27	FULLWIDTH LATIN SMALL LETTER G	FULLWIDTH LATIN CAPITAL LETTER G
FF48	FF28	FULLWIDTH LATIN SMALL LETTER H	FULLWIDTH LATIN CAPITAL LETTER H
FF49	FF29	FULLWIDTH LATIN SMALL LETTER I	FULLWIDTH LATIN CAPITAL LETTER I

Lowercase code point	Uppercase code point	Lowercase character description	Uppercase character description
FF4A	FF2A	FULLWIDTH LATIN SMALL LETTER J	FULLWIDTH LATIN CAPITAL LETTER J
FF4B	FF2B	FULLWIDTH LATIN SMALL LETTER K	FULLWIDTH LATIN CAPITAL LETTER K
FF4C	FF2C	FULLWIDTH LATIN SMALL LETTER L	FULLWIDTH LATIN CAPITAL LETTER L
FF4D	FF2D	FULLWIDTH LATIN SMALL LETTER M	FULLWIDTH LATIN CAPITAL LETTER M
FF4E	FF2E	FULLWIDTH LATIN SMALL LETTER N	FULLWIDTH LATIN CAPITAL LETTER N
FF4F	FF2F	FULLWIDTH LATIN SMALL LETTER O	FULLWIDTH LATIN CAPITAL LETTER O
FF50	FF30	FULLWIDTH LATIN SMALL LETTER P	FULLWIDTH LATIN CAPITAL LETTER P
FF51	FF31	FULLWIDTH LATIN SMALL LETTER Q	FULLWIDTH LATIN CAPITAL LETTER Q
FF52	FF32	FULLWIDTH LATIN SMALL LETTER R	FULLWIDTH LATIN CAPITAL LETTER R
FF53	FF33	FULLWIDTH LATIN SMALL LETTER S	FULLWIDTH LATIN CAPITAL LETTER S
FF54	FF34	FULLWIDTH LATIN SMALL LETTER T	FULLWIDTH LATIN CAPITAL LETTER T
FF55	FF35	FULLWIDTH LATIN SMALL LETTER U	FULLWIDTH LATIN CAPITAL LETTER U
FF56	FF36	FULLWIDTH LATIN SMALL LETTER V	FULLWIDTH LATIN CAPITAL LETTER V
FF57	FF37	FULLWIDTH LATIN SMALL LETTER W	FULLWIDTH LATIN CAPITAL LETTER W
FF58	FF38	FULLWIDTH LATIN SMALL LETTER X	FULLWIDTH LATIN CAPITAL LETTER X
FF59	FF39	FULLWIDTH LATIN SMALL LETTER Y	FULLWIDTH LATIN CAPITAL LETTER Y
FF5A	FF3A	FULLWIDTH LATIN SMALL LETTER Z	FULLWIDTH LATIN CAPITAL LETTER Z

REXX extension characters

These tables show the REXX™ extension characters that are supported on the i5/OS operating system.

REXX/400 extension characters: Axxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with A.

GCGID	Description	Token type	Token flag
AA010000	Aleph (A/F/U) - isolated	NAME	
AA010002	Aleph (A/F/U) - final	NAME	
AA010006	Aleph (after Lam) (A/F/U) - final	NAME	
AA020000	Aleph Maksura (A) - isolated	NAME	
AA020002	Aleph Maksura (A) - final	NAME	
AA070009	Fathatan (A) - intrinsic	NAME	
AA210000	Aleph Madda (A), Aleph Maddey (F), Aleph Madd (U) - Isolated	NAME	

GCGID	Description	Token type	Token flag
AA210002	Aleph Madda (A), Aleph Maddey (F) - final	NAME	
AA210006	Aleph Madda (after Lam) (A), Aleph Maddey (after Lam) (F) - final	NAME	
AA310000	Aleph Hamza (A), Aleph Hamzey (F) - isolated	NAME	
AA310002	Aleph Hamza (A), Aleph Hamzey (F) - final	NAME	
AA310006	Aleph Hamza (after Lam) (A), Aleph Hamzey (after Lam) (F) - final	NAME	
AB010000	Beh (A/F/U) - isolated-final	NAME	
AB010003	Beh (A/F/U) - initial-middle	NAME	
AC210000	Tcheh (F/U) - isolated-final	NAME	
AC210003	Tcheh (F/U) - initial-middle	NAME	
AC470000	Ayn (A/F/U) - isolated	NAME	
AC470002	Ayn (A/F/U) - final	NAME	
AC470003	Ayn (A/F/U) - initial	NAME	
AC470004	Ayn (A/F/U) - middle	NAME	
AD010000	Dal (A/F/U) - isolated-final	NAME	
AD450000	Dud (1st part) (A) - isolated-final	NAME	
AD450003	Dud (A), Zad (F), Duad (U) - initial-middle	NAME	
AD450006	Dud (A), Zad (F), Duad (U) - isolated-final	NAME	
AD470000	Thal (A), Zal (F/U) - isolated-final	NAME	
AF010000	Feh (A/F/U) - isolated-final	NAME	
AF010003	Feh (A/F/U) - initial-middle	NAME	
AG010000	Gaf (F/U) - isolated-final	NAME	
AG010003	Gaf (F/U) - initial-middle	NAME	
AG230000	Jeem (A/F/U) - isolated-final	NAME	
AG230003	Jeem (A/F/U) - initial-middle	NAME	
AG310000	Ghayn (A/F/U) - isolated	NAME	
AG310002	Ghayn (A/F/U) - final	NAME	
AG310003	Ghayn (A/F/U) - initial	NAME	
AG310004	Ghayn (A/F/U) - middle	NAME	
AH010000	Heh (A/F) - isolated-final	NAME	

GCGID	Description	Token type	Token flag
AH010003	Heh (A/F) - initial	NAME	
AH010004	Heh (A/F) - middle	NAME	
AH210000	Heh Yey (F) - isolated-final	NAME	
AH450000	Hah (A), Hey (F), Heh (U) - Isolated-Final	NAME	
AH450003	Hah (A), Hey (F), Heh (U) - initial-middle	NAME	
AH470000	Khah (A), Khey (F), Kheh (U) - Isolated-Final	NAME	
AH470003	Khah (A), Khey (F), Kheh (U) - initial-middle	NAME	
AK010000	Caf (A) - isolated-final	NAME	
AK010003	Caf (A/F/U) - initial-middle	NAME	
AK010006	Caf (F/U) - isolated-final	NAME	
AL010000	Lam (A/F/U) - isolated-final	NAME	
AL010003	Lam (A/F) - initial-middle	NAME	
AL020000	Lamaleph (A/F) - isolated	NAME	
AL020003	Lamaleph (A/F) - final	NAME	
AL220000	Lamaleph Madda (A), Lamaleph Maddey (F) - Isolated	NAME	
AL220003	Lamaleph Madda (A), Lamaleph Maddey (F) - final	NAME	
AL320000	Lamaleph Hamza (A), Lamaleph Hamzey (F) - isolated	NAME	
AL320003	Lamaleph Hamza (A), Lamaleph Hamzey (F) - final	NAME	
AM010000	Meem (A/F/U) - isolated-final	NAME	
AM010003	Meem (A/F/U) - initial-middle	NAME	
AN010000	Noon (A/F/U) - isolated-final	NAME	
AN010003	Noon (A/F/U) - initial-middle	NAME	
AP010000	Peh (F/U) - isolated-final	NAME	
AP010003	Peh (F/U) - initial-middle	NAME	
AQ010000	Qaf (A/F/U) - isolated-final	NAME	
AQ010003	Qaf (A/F/U) - initial-middle	NAME	

GCGID	Description	Token type	Token flag
AR010000	Reh (A/F/U) - isolated-final	NAME	
AS010000	Seen (1st part) (A) - isolated-final	NAME	
AS010003	Seen (A/F/U) - initial-middle	NAME	
AS010006	Seen (A/F/U) - isolated-final	NAME	
AS230000	Sheen (1st part) (A) - isolated-final	NAME	
AS230003	Sheen (A/F/U) - initial-middle	NAME	
AS230006	Sheen (A/F/U) - isolated-final	NAME	
AS450000	Sad (1st part) (A) - isolated-final	NAME	
AS450003	Sad (A/F), Suad (U) - initial-middle	NAME	
AS450006	Sad (A/F), Suad (U) - isolated-final	NAME	
AT010000	Teh (A/F/U) - isolated-final	NAME	
AT010003	Teh (A/F/U) - initial-middle	NAME	
AT020000	Teh Marbuta (A), Teh Mudawara (U) - isolated-final	NAME	
AT450000	Tah (A/F), Toey (U) - isolated-final-initial-middle	NAME	
AT450001	Tah (A/F), Toey (U) - isolated-final	NAME	
AT450002	Tah (A/F), Toey (U) - initial-middle	NAME	
AT470000	Theh (A/F/U) - isolated-final	NAME	
AT470003	Theh (A/F/U) - initial-middle	NAME	
AW010000	Waw (A), Vav (F), Waow (U) - isolated-final	NAME	
AW310000	Waw Hamza (A), Vav Hamzey (F), Waow Hamza (U) - isolated-final	NAME	
AX100000	Shadda (A/F), Shadd (U) - isolated	NAME	
AX100004	Shadda (A/F), Shadd (U) - middle	NAME	
AX300000	Hamza (A), Hamzey (F), Hamza (U) - isolated	NAME	

GCGID	Description	Token type	Token flag
AY010000	Yeh (A) - isolated	NAME	
AY010002	Yeh (A) - final	NAME	
AY010003	Yeh (A) - initial-middle	NAME	
AY020000	Yey (F), Yeh Chotee (U) - isolated	NAME	
AY020002	Yey (F), Yeh Chotee (U) - final	NAME	
AY020003	Yey (F), Yeh Chotee (U) - initial-middle	NAME	
AY310000	Yeh Hamza (A) - initial-middle	NAME	
AY320003	Yey Hamzey (F), Yeh Chotee Hamza (U) - initial-middle	NAME	
AZ010000	Zayn (A), Zey (F), Zeh (U) - isolated-final	NAME	
AZ210000	Jey (F), Zzeh (U) - isolated-final	NAME	
AZ450000	Zah (F), Zoey (U) - isolated-final-initial-middle	NAME	
AZ450001	Zah (A/F), Zoey (U) - isolated-final	NAME	
AZ450002	Zah (A/F), Zoey (U) - initial-middle	NAME	

REXX/400 extension characters: Bxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with B.

GCGID	Description	Token type	Token flag
BA100000	a - (upper vowel)	NAME	
BA200000	a - (middle vowel)	NAME	
BA300000	a - (middle vowel)	NAME	
BA400000	am - (middle vowel)	NAME	
BA500000	ai - (middle vowel)	NAME	
BA600000	ai - (middle vowel)	NAME	
BA700000	a - (middle vowel)	NAME	
BB100000	Bo	NAME	
BC100000	Cho	NAME	
BD100000	Do	NAME	
BD200000	Do	NAME	
BE100000	e/a - (upper vowel)	NAME	
BE200000	e - (middle vowel)	NAME	
BE300000	e - (middle vowel)	NAME	
BE400000	Yamakkan	NAME	

GCGID	Description	Token type	Token flag
BF100000	Fo	NAME	
BF200000	Fo	NAME	
BH100000	Ho	NAME	
BH200000	Ho	NAME	
BI100000	i - (upper vowel)	NAME	
BI200000	i - (upper vowel)	NAME	
BK100000	Ko	NAME	
BK200000	Kho	NAME	
BK300000	Kho	NAME	
BK400000	Kho	NAME	
BK500000	Kho	NAME	
BK600000	Kho	NAME	
BL100000	Lo	NAME	
BL200000	Lu	NAME	
BL300000	Lo	NAME	
BM100000	Mo	NAME	
BN100000	Ngo	NAME	
BN200000	No	NAME	
BN300000	No	NAME	
BN400000	a - (upper vowel)	NAME	
BO100000	o	NAME	
BO200000	o - (middle vowel)	NAME	
BP100000	Po	NAME	
BP200000	Pho	NAME	
BP300000	Pho	NAME	
BP400000	Pho	NAME	
BQ100000	Thai repeat sign	NAME	
BQ200000	Thai Ellipsis	NAME	
BQ300000	a - (lower vowel)	NAME	
BQ400000	Fongmann	NAME	
BQ500000	Angkhankhu	NAME	
BQ600000	Komut	NAME	
BR100000	Ro	NAME	
BR200000	Ro	NAME	
BS100000	So	NAME	
BS200000	So	NAME	
BS300000	So	NAME	
BS400000	So	NAME	
BT100000	To	NAME	
BT200000	Tho	NAME	
BT300000	Tho	NAME	

GCGID	Description	Token type	Token flag
BT400000	Tho	NAME	
BT500000	To	NAME	
BT600000	Tho	NAME	
BT700000	Tho	NAME	
BT800000	Tho	NAME	
BU100000	u - (upper vowel)	NAME	
BU200000	u - (upper vowel)	NAME	
BU300000	u - (lower vowel)	NAME	
BU400000	u - (lower vowel)	NAME	
BW100000	Wo	NAME	
BX100000	Xo	NAME	
BX200000	Xo	NAME	
BX300000	Xo	NAME	
BY100000	Jo	NAME	
BY200000	Yo	NAME	
BZ100000	1st tone mark	NAME	
BZ100300	1st tone mark, low position	NAME	
BZ200000	2nd tone mark	NAME	
BZ200300	2nd tone mark, low position	NAME	
BZ300000	3rd tone mark	NAME	
BZ300300	3rd tone mark, low position	NAME	
BZ400000	4th tone mark	NAME	
BZ400300	4th tone mark, low position	NAME	
BZ500000	5th tone mark	NAME	
BZ500300	5th tone mark, low position	NAME	

REXX/400 extension characters: Gxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with G.

GCGID	Description	Token type	Token flag
GA010000	Alpha small	NAME	
GA020000	Alpha capital	NAME	
GA110000	Alpha acute small	NAME	
GA120000	Alpha acute capital	NAME	
GB010000	Beta small	NAME	
GB020000	Beta capital	NAME	
GD010000	Delta small	NAME	
GD020000	Delta capital	NAME	
GE010000	Epsilon small	NAME	
GE020000	Epsilon capital	NAME	

GCGID	Description	Token type	Token flag
GE110000	Epsilon acute small	NAME	
GE120000	Epsilon acute capital	NAME	
GE310000	Eta small	NAME	
GE320000	Eta capital	NAME	
GE710000	Eta acute small	NAME	
GE720000	Eta acute capital	NAME	
GF010000	Phi small	NAME	
GF020000	Phi capital	NAME	
GG010000	Gamma small	NAME	
GG020000	Gamma capital	NAME	
GH010000	Chi small	NAME	
GH020000	Chi capital	NAME	
GI010000	Iota small	NAME	
GI020000	Iota capital	NAME	
GI110000	Iota acute small	NAME	
GI120000	Iota acute capital	NAME	
GI170000	Iota diaeresis small	NAME	
GI180000	Iota diaeresis capital	NAME	
GI730000	Iota acute and diaeresis small	NAME	
GK010000	Kappa small	NAME	
GK020000	Kappa capital	NAME	
GL010000	Lambda small	NAME	
GL020000	Lambda capital	NAME	
GM010000	Mu small	NAME	
GM020000	Mu capital	NAME	
GN010000	Nu small	NAME	
GN020000	Nu capital	NAME	
GO010000	Omicron small	NAME	
GO020000	Omicron capital	NAME	
GO110000	Omicron acute small	NAME	
GO120000	Omicron acute capital	NAME	
GO310000	Omega small	NAME	
GO320000	Omega capital	NAME	
GO710000	Omega acute small	NAME	
GO720000	Omega acute capital	NAME	
GP010000	Pi small	NAME	
GP020000	Pi capital	NAME	
GP610000	Psi small	NAME	
GP620000	Psi capital	NAME	
GR010000	Rho small	NAME	

GCGID	Description	Token type	Token flag
GR020000	Rho capital	NAME	
GS010000	Sigma small	NAME	
GS020000	Sigma capital	NAME	
GS610000	Sigma small (final form)	NAME	
GT010000	Tau small	NAME	
GT020000	Tau capital	NAME	
GT610000	Theta small	NAME	
GT620000	Theta capital	NAME	
GU010000	Upsilon small	NAME	
GU020000	Upsilon capital	NAME	
GU110000	Upsilon acute small	NAME	
GU120000	Upsilon acute capital	NAME	
GU170000	Upsilon diaeresis small	NAME	
GU180000	Upsilon diaeresis capital	NAME	
GU730000	Upsilon acute and diaeresis small	NAME	
GX010000	Xi small	NAME	
GX020000	Xi capital	NAME	
GZ010000	Zeta small	NAME	
GZ020000	Zeta capital	NAME	

REXX/400 extension characters: Hxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with H.

GCGID	Description	Token type	Token flag
HB010000	Bet	NAME	
HD010000	Dalet	NAME	
HG010000	Gimel	NAME	
HH010000	He	NAME	
HH450000	Het	NAME	
HK010000	Kaf	NAME	
HK610000	Kaf (Final Form)	NAME	
HL010000	Lamed	NAME	
HM010000	Mem	NAME	
HM610000	Mem (final form)	NAME	
HN010000	Nun	NAME	
HN610000	Nun (final form)	NAME	
HP010000	Pe	NAME	
HP610000	Pe (final form)	NAME	
HQ010000	Qof	NAME	
HR010000	Resh	NAME	

GCGID	Description	Token type	Token flag
HS010000	Samech	NAME	
HS210000	Shin	NAME	
HS450000	Zadi	NAME	
HS610000	Zadi (Final Form)	NAME	
HT010000	Tav	NAME	
HT450000	Tet	NAME	
HW010000	Waw	NAME	
HX330000	Alef	NAME	
HX350000	Ayin	NAME	
HY010000	Yod	NAME	
HZ010000	Zayin	NAME	

REXX/400 extension characters: Jxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with J.

GCGID	Description	Token type	Token flag
JA000000	A	NAME	
JA010000	a	NAME	
JE000000	E	NAME	
JE010000	e	NAME	
JH100000	HA	NAME	
JH200000	HI	NAME	
JH300000	HU or FU	NAME	
JH400000	HE	NAME	
JH500000	HO	NAME	
JI000000	I	NAME	
JI010000	i	NAME	
JK100000	KA	NAME	
JK200000	KI	NAME	
JK300000	KU	NAME	
JK400000	KE	NAME	
JK500000	KO	NAME	
JM100000	MA	NAME	
JM200000	MI	NAME	
JM300000	MU	NAME	
JM400000	ME	NAME	
JM500000	MO	NAME	
JN000000	N	NAME	
JN100000	NA	NAME	
JN200000	NI	NAME	
JN300000	NU	NAME	

GCGID	Description	Token type	Token flag
JN400000	NE	NAME	
JN500000	NO	NAME	
JO000000	O	NAME	
JO010000	o	NAME	
JQ700000	Katakana full stop	NAME	
JQ710000	Katakana left bracket	NAME	
JQ720000	Katakana right bracket	NAME	
JQ730000	Katakana comma	NAME	
JQ740000	Katakana conjunctive symbol	NAME	
JR100000	RA	NAME	
JR200000	RI	NAME	
JR300000	RU	NAME	
JR400000	RE	NAME	
JR500000	RO	NAME	
JS100000	SA	NAME	
JS200000	SI or SHI	NAME	
JS300000	SU	NAME	
JS400000	SE	NAME	
JS500000	SO	NAME	
JT100000	TA	NAME	
JT200000	TI or CHI	NAME	
JT300000	TU or TSU	NAME	
JT310000	tu or tsu	NAME	
JT400000	TE	NAME	
JT500000	TO	NAME	
JU000000	U	NAME	
JU010000	u	NAME	
JW100000	WA	NAME	
JW500000	WO, Katakana participle	NAME	
JX700000	Prolonged sound symbol	NAME	
JX710000	Voiced sound symbol	NAME	
JX720000	Semi-voiced sound symbol	NAME	
JY100000	YA	NAME	
JY110000	ya	NAME	
JY300000	YU	NAME	
JY310000	yu	NAME	
JY500000	YO	NAME	
JY510000	yo	NAME	

REXX/400 extension characters: Kxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with K.

GCGID	Description	Token type	Token flag
KA010000	a small	NAME	
KA020000	A capital	NAME	
KA150000	ya small	NAME	
KA160000	YA capital	NAME	
KB010000	b small	NAME	
KB020000	B capital	NAME	
KC010000	ts small	NAME	
KC020000	TS capital	NAME	
KC110000	c special small	NAME	
KC120000	C special capital	NAME	
KC210000	ch small	NAME	
KC220000	CH capital	NAME	
KD010000	d small	NAME	
KD020000	D capital	NAME	
KD610000	d special small	NAME	
KD620000	D special capital	NAME	
KE010000	e small	NAME	
KE020000	E capital	NAME	
KE130000	e special small	NAME	
KE140000	E special capital	NAME	
KE150000	ye small	NAME	
KE160000	YE capital	NAME	
KE170000	e diaeresis small	NAME	
KE180000	E diaeresis capital	NAME	
KF010000	f small	NAME	
KF020000	F capital	NAME	
KG010000	g small	NAME	
KG020000	G capital	NAME	
KG110000	g special small	NAME	
KG120000	G special capital	NAME	
KG210000	dz special small	NAME	
KG220000	DZ special capital	NAME	
KH010000	kh small	NAME	
KH020000	KH capital	NAME	
KI010000	i small	NAME	
KI020000	I capital	NAME	
KI110000	i special small	NAME	
KI120000	I special capital	NAME	

GCGID	Description	Token type	Token flag
KI170000	i diaeresis small	NAME	
KI180000	I diaeresis capital	NAME	
KJ010000	j small	NAME	
KJ020000	J capital	NAME	
KJ110000	j special small	NAME	
KJ120000	J special capital	NAME	
KK010000	k small	NAME	
KK020000	K capital	NAME	
KK110000	k special small	NAME	
KK120000	K special capital	NAME	
KL010000	l small	NAME	
KL020000	L capital	NAME	
KL410000	lj small	NAME	
KL420000	LJ capital	NAME	
KM010000	m small	NAME	
KM020000	M capital	NAME	
KN010000	n small	NAME	
KN020000	N capital	NAME	
KN110000	nj small	NAME	
KN120000	NJ capital	NAME	
KO010000	o small	NAME	
KO020000	O capital	NAME	
KP010000	p small	NAME	
KP020000	P capital	NAME	
KR010000	r small	NAME	
KR020000	R capital	NAME	
KS010000	s small	NAME	
KS020000	S capital	NAME	
KS150000	shch small	NAME	
KS160000	SHCH capital	NAME	
KS210000	sh small	NAME	
KS220000	SH capital	NAME	
KT010000	t small	NAME	
KT020000	T capital	NAME	
KU010000	u small	NAME	
KU020000	U capital	NAME	
KU150000	yu small	NAME	
KU160000	YU capital	NAME	
KU210000	Hard sign small	NAME	
KU220000	Hard sign capital	NAME	
KU230000	u Breve small	NAME	

GCGID	Description	Token type	Token flag
KU240000	U Breve capital	NAME	
KV010000	v small	NAME	
KV020000	V capital	NAME	
KX110000	Soft sign small	NAME	
KX120000	Soft sign capital	NAME	
KY010000	y small	NAME	
KY020000	Y capital	NAME	
KZ010000	z small	NAME	
KZ020000	Z capital	NAME	
KZ150000	s special small	NAME	
KZ160000	S special capital	NAME	
KZ210000	zh small	NAME	
KZ220000	zh capital	NAME	

REXX/400 extension characters: Lxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with L.

GCGID	Description	Token type	Token flag
LA010000	a small	NAME	
LA020000	A capital	NAME	
LA110000	a acute small	NAME	
LA120000	A acute capital	NAME	
LA130000	a grave small	NAME	
LA140000	A grave capital	NAME	
LA150000	a circumflex small	NAME	
LA160000	A circumflex capital	NAME	
LA170000	a diaeresis small	NAME	
LA180000	A diaeresis capital	NAME	
LA190000	a tilde Small	NAME	
LA200000	A tilde capital	NAME	
LA230000	a breve small	NAME	
LA240000	A breve capital	NAME	
LA270000	a overcircle small	NAME	
LA280000	A overcircle capital	NAME	
LA430000	a ogonek small	NAME	
LA440000	A ogonek capital	NAME	
LA510000	ae diphthong small	NAME	
LA520000	ae diphthong capital	NAME	
LB010000	b small	NAME	
LB020000	B capital	NAME	
LC010000	c small	NAME	

GCGID	Description	Token type	Token flag
LC020000	C capital	NAME	
LC110000	c acute small	NAME	
LC120000	C acute capital	NAME	
LC150000	c circumflex small	NAME	
LC160000	C circumflex capital	NAME	
LC210000	c caron small	NAME	
LC220000	C caron capital	NAME	
LC290000	c overdot small	NAME	
LC300000	C overdot capital	NAME	
LC410000	c cedilla small	NAME	
LC420000	C cedilla capital	NAME	
LD010000	d small	NAME	
LD020000	D capital	NAME	
LD210000	d caron small	NAME	
LD220000	D caron capital	NAME	
LD610000	d stroke small	NAME	
LD620000	D stroke capital/Eth Icelandic capital	NAME	
LD630000	eth Icelandic small	NAME	
LE010000	e small	NAME	
LE020000	E capital	NAME	
LE110000	e acute small	NAME	
LE120000	E acute capital	NAME	
LE130000	e grave small	NAME	
LE140000	E grave capital	NAME	
LE150000	e circumflex small	NAME	
LE160000	E circumflex capital	NAME	
LE170000	e diaeresis small	NAME	
LE180000	E diaeresis capital	NAME	
LE210000	e caron small	NAME	
LE220000	E caron capital	NAME	
LE430000	e ogonek small	NAME	
LE440000	E ogonek capital	NAME	
LF010000	f small	NAME	
LF020000	F capital	NAME	
LG010000	g small	NAME	
LG020000	G capital	NAME	
LG150000	g circumflex small	NAME	
LG160000	G circumflex capital	NAME	
LG230000	g breve small	NAME	
LG240000	G breve capital	NAME	

GCGID	Description	Token type	Token flag
LG290000	g overdot small	NAME	
LG300000	G overdot capital	NAME	
LH010000	h small	NAME	
LH020000	H capital	NAME	
LH150000	h circumflex small	NAME	
LH160000	H circumflex capital	NAME	
LH610000	h stroke small	NAME	
LH620000	H stroke capital	NAME	
LI010000	i small	NAME	
LI020000	I capital	NAME	
LI110000	i acute small	NAME	
LI120000	I acute capital	NAME	
LI130000	i grave small	NAME	
LI140000	I grave capital	NAME	
LI150000	i circumflex small	NAME	
LI160000	I circumflex capital	NAME	
LI170000	i diaeresis small	NAME	
LI180000	I diaeresis capital	NAME	
LI300000	I overdot capital	NAME	
LI610000	i dotless small	NAME	
LJ010000	j small	NAME	
LJ020000	J capital	NAME	
LJ150000	j circumflex small	NAME	
LJ160000	J circumflex capital	NAME	
LK010000	k small	NAME	
LK020000	K capital	NAME	
LL010000	l small	NAME	
LL020000	L capital	NAME	
LL110000	l acute small	NAME	
LL120000	L acute capital	NAME	
LL210000	l caron small	NAME	
LL220000	L caron capital	NAME	
LL610000	l stroke small	NAME	
LL620000	L stroke capital	NAME	
LM010000	m small	NAME	
LM020000	M capital	NAME	
LN010000	n small	NAME	
LN020000	N capital	NAME	
LN110000	n acute small	NAME	
LN120000	N acute capital	NAME	
LN190000	n tilde small	NAME	

GCGID	Description	Token type	Token flag
LN200000	N tilde capital	NAME	
LN210000	n caron small	NAME	
LN220000	N caron capital	NAME	
LO010000	o small	NAME	
LO020000	O capital	NAME	
LO110000	o acute small	NAME	
LO120000	O acute capital	NAME	
LO130000	o grave small	NAME	
LO140000	O grave capital	NAME	
LO150000	o circumflex small	NAME	
LO160000	O circumflex capital	NAME	
LO170000	o diaeresis small	NAME	
LO180000	O diaeresis capital	NAME	
LO190000	o tilde small	NAME	
LO200000	O tilde capital	NAME	
LO250000	o double acute small	NAME	
LO260000	O double acute capital	NAME	
LO610000	o slash small	NAME	
LO620000	O slash capital	NAME	
LP010000	p small	NAME	
LP020000	P capital	NAME	
LQ010000	q small	NAME	
LQ020000	Q capital	NAME	
LR010000	r small	NAME	
LR020000	R capital	NAME	
LR110000	r acute small	NAME	
LR120000	R acute capital	NAME	
LR210000	r caron small	NAME	
LR220000	R caron capital	NAME	
LS010000	s small	NAME	
LS020000	S capital	NAME	
LS110000	s acute small	NAME	
LS120000	S acute capital	NAME	
LS150000	s circumflex small	NAME	
LS160000	S circumflex capital	NAME	
LS210000	s caron small	NAME	
LS220000	S caron capital	NAME	
LS410000	s cedilla small	NAME	
LS420000	S cedilla capital	NAME	
LS610000	Sharp s small	NAME	
LT010000	t small	NAME	

GCGID	Description	Token type	Token flag
LT020000	T capital	NAME	
LT210000	t caron small	NAME	
LT220000	T caron capital	NAME	
LT410000	t cedilla small	NAME	
LT420000	T cedilla capital	NAME	
LT630000	Thorn Icelandic small	NAME	
LT640000	Thorn Icelandic capital	NAME	
LU010000	u small	NAME	
LU020000	U capital	NAME	
LU110000	u acute small	NAME	
LU120000	U acute capital	NAME	
LU130000	u grave small	NAME	
LU140000	U grave capital	NAME	
LU150000	u circumflex small	NAME	
LU160000	U circumflex capital	NAME	
LU170000	u diaeresis small	NAME	
LU180000	U diaeresis capital	NAME	
LU230000	u breve small	NAME	
LU240000	U breve capital	NAME	
LU250000	u double acute small	NAME	
LU260000	U double acute capital	NAME	
LU270000	u overcircle small	NAME	
LU280000	u overcircle capital	NAME	
LV010000	v small	NAME	
LV020000	V capital	NAME	
LW010000	w small	NAME	
LW020000	W capital	NAME	
LX010000	x small	NAME	
LX020000	X capital	NAME	
LY010000	y small	NAME	
LY020000	Y capital	NAME	
LY110000	y acute small	NAME	
LY120000	Y acute capital	NAME	
LY170000	y diaeresis small	NAME	
LZ010000	z small	NAME	
LZ020000	Z capital	NAME	
LZ110000	z acute small	NAME	
LZ120000	Z acute capital	NAME	
LZ210000	z caron small	NAME	
LZ220000	Z caron capital	NAME	
LZ290000	z overdot small	NAME	

GCGID	Description	Token type	Token flag
LZ300000	Z overdot capital	NAME	

REXX/400 extension characters: Nxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with N.

GCGID	Description	Token type	Token flag
ND010000	One	NUMBER	
ND010001	One (Arabic, Farsi, Urdu)	INVALID	
ND010002	One, Thai	INVALID	
ND011000	One superscript	INVALID	
ND020000	Two	NUMBER	
ND020001	Two (Arabic, Farsi, Urdu)	INVALID	
ND020002	Two, Thai	INVALID	
ND021000	Two superscript	INVALID	
ND030000	Three	NUMBER	
ND030001	Three (Arabic, Farsi, Urdu)	INVALID	
ND030002	Three, Thai	INVALID	
ND031000	Three superscript	INVALID	
ND040000	Four	NUMBER	
ND040001	Four (Arabic)	INVALID	
ND040002	Four, Thai	INVALID	
ND040003	Four (Farsi)	INVALID	
ND050000	Five	NUMBER	
ND050001	Five (Arabic)	INVALID	
ND050002	Five, Thai	INVALID	
ND050004	Five (Farsi, Urdu)	INVALID	
ND060000	Six	NUMBER	
ND060001	Six (Arabic, Urdu)	INVALID	
ND060002	Six, Thai	INVALID	
ND060003	Six (Farsi)	INVALID	
ND070000	Seven	NUMBER	
ND070001	Seven (Arabic, Farsi)	INVALID	
ND070002	Seven, Thai	INVALID	
ND080000	Eight	NUMBER	
ND080001	Eight (Arabic, Farsi, Urdu)	INVALID	
ND080002	Eight, Thai	INVALID	
ND090000	Nine	NUMBER	
ND090001	Nine (Arabic, Farsi, Urdu)	INVALID	
ND090002	Nine, Thai	INVALID	
ND100000	Zero	NUMBER	
ND100001	Zero (Arabic, Urdu)	INVALID	

GCGID	Description	Token type	Token flag
ND100002	Zero, Thai	INVALID	
ND100003	Zero (Farsi)	INVALID	
NF010000	One half	INVALID	
NF040000	One quarter	INVALID	
NF050000	Three quarters	INVALID	

REXX/400 extension characters: Oxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with O.

GCGID	Description	Token type	Token flag
OA000000	A (basic vowel)	NAME	
OA200000	AE (compound vowel)	NAME	
OB000000	B (basic consonant)	NAME	
OB100000	BB (compound consonant)	NAME	
OB200000	BS (compound consonant)	NAME	
OC200000	CH (basic consonant)	NAME	
OD000000	D (basic consonant)	NAME	
OD100000	DD (compound consonant)	NAME	
OE000000	E (compound vowel)	NAME	
OE200000	EO (basic vowel)	NAME	
OE300000	EU (basic vowel)	NAME	
OE400000	EUI (compound vowel)	NAME	
OG000000	G (basic consonant)	NAME	
OG100000	GG (compound consonant)	NAME	
OG200000	GS (compound consonant)	NAME	
OH000000	H (basic consonant)	NAME	
OI000000	I (basic vowel)	NAME	
OJ000000	J (basic consonant)	NAME	
OJ100000	JJ (compound consonant)	NAME	
OK000000	K (basic consonant)	NAME	
OL000000	L (basic consonant)	NAME	
OL100000	LB (compound consonant)	NAME	
OL200000	LG (compound consonant)	NAME	
OL300000	LH (compound consonant)	NAME	
OL400000	:c 2 .LM (compound consonant)	NAME	
OL500000	LP (compound consonant)	NAME	
OL600000	LS (compound consonant)	NAME	
OL700000	LT (compound consonant)	NAME	
OM000000	M (basic consonant)	NAME	
ON000000	N (basic consonant)	NAME	

GCGID	Description	Token type	Token flag
ON100000	NH (compound consonant)	NAME	
ON150000	NJ (compound consonant)	NAME	
ON200000	NG or W (basic consonant)	NAME	
OO000000	O (basic vowel)	NAME	
OO100000	OA (compound vowel)	NAME	
OO200000	OAE (compound vowel)	NAME	
OO300000	OI (compound vowel)	NAME	
OP000000	P (basic consonant)	NAME	
OS000000	S (basic consonant)	NAME	
OS100000	SS (compound consonant)	NAME	
OT000000	T (basic consonant)	NAME	
OU000000	U (basic vowel)	NAME	
OU200000	UE (compound vowel)	NAME	
OU300000	UEO (compound vowel)	NAME	
OU400000	UI (compound vowel)	NAME	
OY200000	YA (basic vowel)	NAME	
OY250000	YAE (compound vowel)	NAME	
OY300000	YE (compound vowel)	NAME	
OY400000	YEO (basic vowel)	NAME	
OY500000	YO (basic vowel)	NAME	
OY600000	YU (basic vowel)	NAME	

REXX/400 extension characters: Sxxxxxxx GCGIDs

The i5/OS operating system supports REXX extension characters. This table lists system-supported REXX extension characters whose GCGID begins with S.

GCGID	Description	Token type	Token flag
SA010000	Plus sign	OPER	PLUS
SA020000	Plus or minus sign	INVALID	
SA030000	Less than sign/greater than sign (Arabic)	OPER	LESS_THAN
SA040000	Equal sign	OPER	EQUAL
SA050000	Greater than sign/less than sign (Arabic)	OPER	GREAT_THAN
SA060000	Divide sign	INVALID	
SA070000	Multiply sign	INVALID	
SC010000	International currency symbol	INVALID	
SC020000	Pound sterling sign	INVALID	
SC030000	Dollar sign	INVALID	
SC040000	Cent sign	INVALID	
SC050000	Yen sign	INVALID	
SC060000	Peseta sign	INVALID	

GCGID	Description	Token type	Token flag
SC070000	Florin sign	INVALID	
SC120000	Yuan sign	INVALID	
SC130000	Currency symbol, Thailand	INVALID	
SC140000	Won sign	INVALID	
SC160000	Rial sign, Iran	INVALID	
SD110000	Acute accent	INVALID	
SD130000	Grave accent	INVALID	
SD150000	Circumflex accent	INVALID	
SD170000	Diaeresis/Umlaut accent	INVALID	
SD190000	Tilde accent	INVALID	
SD210000	Caron accent	INVALID	
SD230000	Breve accent	INVALID	
SD250000	Double acute accent	INVALID	
SD290000	Overdot accent	INVALID	
SD410000	Cedilla or sedila accent	INVALID	
SD430000	Ogonek accent	INVALID	
SD630000	Middle dot	INVALID	
SD730000	Acute and diaeresis accent	INVALID	
SM000000	Numero sign	INVALID	
SM010000	Number sign	INVALID	
SM020000	Percent sign	OPER	PERCENT
SM020007	Percent sign (Arabic)	OPER	PERCENT
SM030000	Ampersand	OPER	AMPERSAND
SM040000	Asterisk	OPER	ASTERISK
SM040007	Asterisk (Arabic preference - 5 points)	OPER	ASTERISK
SM050000	At sign	INVALID	
SM060000	Left bracket	INVALID	
SM070000	Backslash	OPER	BACKSLASH
SM080000	Right bracket	INVALID	
SM100000	Double underscore	INVALID	
SM110000	Left brace	INVALID	
SM120000	Long dash/throughscore	INVALID	
SM130000	Vertical line/logical OR	OPER	VERTI_BAR
SM140000	Right brace	INVALID	
SM150000	Overline	INVALID	
SM170000	Micro symbol	INVALID	
SM190000	Degree symbol	INVALID	
SM200000	Ordinal indicator, Masculine	INVALID	
SM210000	Ordinal indicator, Feminine	INVALID	

GCGID	Description	Token type	Token flag
SM240000	Section symbol (USA)/paragraph symbol (Europe)	INVALID	
SM250000	Paragraph symbol (USA)	INVALID	
SM520000	Copyright symbol	INVALID	
SM530000	Registered trademark symbol	INVALID	
SM570000	Bullet	INVALID	
SM650000	Vertical line, broken	INVALID	
SM660000	Logical NOT/End Of Line symbol	OPER	NOT_SIGN
SM860000	Tatweel (Connector)	INVALID	
SM870000	Kasseh (Tail)	INVALID	
SP010000	Space	WHITE	BLANK
SP020000	Exclamation point	NAME	
SP030000	Exclamation point, inverted	INVALID	
SP040000	Quotation marks	STRING	QUOTE
SP050000	Apostrophe	STRING	APOSTROPH
SP060000	Left parenthesis	PUNCT	L_PAREN
SP070000	Right parenthesis	PUNCT	R_PAREN
SP080000	Comma	PUNCT	COMMA
SP080007	Comma rotated (Arabic)	INVALID	
SP090000	Underline/continuous underscore	NAME	
SP100000	Hyphen/minus sign	OPER	MINUS
SP110000	Period/full stop	NAME	PERIOD
SP120000	Slash	OPER	SLASH
SP130000	Colon	PUNCT	COLON
SP140000	Semicolon	PUNCT	SEMI_COLON
SP140007	Semicolon, rotated (Arabic)	INVALID	
SP150000	Question mark	NAME	QUESTION
SP150007	Question mark, reversed (Arabic)	INVALID	
SP160000	Question mark, inverted	INVALID	
SP170000	Left angle quotation marks	INVALID	
SP180000	Right angle quotation marks	INVALID	
SP190000	Left single quotation mark	INVALID	
SP200000	Right single quotation mark	INVALID	
SP300000	Required space	INVALID	
SP310000	Numeric space	INVALID	
SP320000	Syllable hyphen	INVALID	

GCGID	Description	Token type	Token flag
SP490000	Korean fill (NULL) character	INVALID	

Default character data conversion that can use substitution

The default CCSID conversions use substitution because the character sets within the CCSIDs are different. The table shows which CCSIDs (From CCSID column) can be substituted by other CCSIDs (To CCSID column).

Table 1. Default conversion that can use substitution

From CCSID	To CCSID
00037	00290, 00833, 00836, 00838, 00930, 00933, 00835, 00939, 00948, 01027, 01043, 05026, 05035
00256	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00273	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00277	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 05026, 05035
00278	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00280	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00284	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00285	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00290	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00833, 00836, 00850 00871, 00897, 00933, 00935, 00937, 00942, 01041
00297	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00300	00301
00301	00300, 04396
00420	04960
00437	00870, 00871, 00880, 01025
00500	00290, 00367, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01010, 01011, 01012, 01013, 01014, 01015, 01016, 01017, 01018, 01019, 01027, 05026, 05035
00833	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00836, 00850, 00871, 00930, 00935, 00937, 00939, 00944, 01027, 01040, 05026, 05035
00836	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00833, 00871, 00903, 00930, 00933, 00937, 00939, 00946, 01027, 01042, 01115, 05026, 05035
00837	00928
00838	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00871, 00937

Table 1. Default conversion that can use substitution (continued)

From CCSID	To CCSID
00850	00290, 00833, 00870, 00880, 00930, 00933, 00939, 01025, 01027, 01041, 05026, 05035
00870	00437, 00880, 00915, 01025, 04951
00871	00290, 00437, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 01027, 05026, 05035
00880	00437, 00850, 00870, 00912, 00915, 04948, 04951
00897	01027
00903	00836
00912	00880, 01025
00915	00870, 00880
00930	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00833, 00836, 00850, 00871, 00933, 00935, 00937, 00942, 01041
00933	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00836, 00850, 00871, 00930, 00934, 00935, 00937, 00939, 00944, 01027, 01040, 05026, 05035
00934	00933
00935	00037, 00256, 00277, 00278, 00280, 00284, 00285, 00290, 00297, 00500, 00833, 00871, 00930, 00933, 00936, 00937, 00939, 00946, 01027, 01042, 05026, 05035
00936	00935
00937	00290, 00833, 00836, 00930, 00933, 00935, 00938, 00939, 00948, 01027, 01043, 05026, 05035
00938	00937
00939	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00833, 00836, 00850, 00871, 00933, 00935, 00937, 00942, 01041
00942	00290, 00930, 00939, 01027, 05026, 05035
00944	00833, 00933
00946	00836, 00935
00948	00037, 00937
01010	00500
01011	00500
01012	00500
01013	00500
01014	00500
01015	00500
01016	00500
01017	00500
01018	00500
01019	00500
01025	00437, 00850, 00870, 00912, 04948

Table 1. Default conversion that can use substitution (continued)

From CCSID	To CCSID
01027	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00833, 00836, 00850, 00871, 00933, 00935, 00937, 00942, 01041
01040	00833, 00933
01041	00290, 00850, 00930, 00939, 01027, 05026, 05035
01042	00836, 00935
01043	00037, 00937
01114	28709
01115	00836
04396	00301
04948	00880, 01025
04951	00870, 00880
04960	00420
05026	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00833, 00836, 00850, 00871, 00933, 00935, 00937, 00942, 01041
05035	00037, 00256, 00273, 00277, 00278, 00280, 00284, 00285, 00297, 00500, 00833, 00836, 00850, 00871, 00933, 00935, 00937, 00942, 01041
28709	00290, 00833, 00836, 00838, 00930, 00933, 00935, 00939, 00948, 01027, 01043, 05026, 05035

Related concepts

"Conversion of character data" on page 5
The Character Data Representation Architecture (CDRA) system of tags ensures that you can convert character data in a predictable, repeatable way.

Globalization checklists

The numerous checklists, presented within the i5/OS globalization topic collection, are useful tools for ensuring that you have considered key items that are 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
"Checklist: Globalization planning" on page 31	This comprehensive two-part checklist, which you can find in "Setting up i5/OS with a national language version" on page 29, provides a good summary of items that relate to hardware installation, software installation, and system configuration.
"Checklist: Application design" on page 56	Use this checklist as you begin planning the development of a globalized application. You can find this checklist in "Designing globalized applications" on page 56.
"Checklist: User interface design" on page 65	Use this checklist to ensure that the user interfaces you create for your applications anticipate unique requirements from multiple linguistic and cultural environments. You can find this checklist in "User interfaces" on page 64.

Checklist	Where you can go to read related topics
"Checklist: Bidirectional support guidelines" on page 175	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 "Working with bidirectional data" on page 174.
"Checklist: DBCS application design" on page 177	You need to take numerous items into consideration when you develop applications that use double-byte character set (DBCS) support. You can find this checklist in "Working with DBCS data" on page 176.

Related information for i5/OS globalization

Product manuals, Web sites, and information center topic collections contain information that relates to the i5/OS globalization topic collection. You can view or print any of the PDFs.

Manuals

The following manuals are not included in the V6R1 i5/OS Information Center. However, these manuals might be a useful reference to you. Each of the manuals is available from the IBM Publications Center as a printed hardcopy that you can order, in an online format that you can download at no charge, or both.

- *Character Data Representation Architecture: Reference & Registry*, SC09-2190

  This manual provides a complete discussion of the Character Data Representation Architecture (CDRA) and a complete list of the CDRA CCSIDs.
- *Character Data Representation Architecture: Overview*, GC09-2207

  This manual provides an overview of CDRA.
- *Communications Configuration*, SC41-5401

  This manual describes the objects, commands, and parameters used to configure operating system communications. It includes a general discussion of the objects and methods used to configure communications and detailed descriptions of all parameters that can be specified for the commands used to create the configuration objects. You can find more information about configuration names in this book.
- *National Language Design Guide Volume 2*, SE09-8002

  This manual provides information about linguistic and cultural conventions by country.
- *National Language Design Guide Volume 1*, SE09-8001

  This manual provides a complete list of the guidelines of DBCS application design, as well as a full description of each guideline.

Web sites

- Globalize your On Demand Business (www.ibm.com/software/globalization)
- System i Globalization (www.ibm.com/servers/eserver/iseries/software/globalization)

Other information

- Database file management
- Database programming
- DDS concepts

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This i5/OS globalization publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Presentation
AFP
DB2
Distributed Relational Database Architecture
DRDA
i5/OS
IBM
IBM (logo)
Intelligent Printer Data Stream
IPDS
iSeries
OS/400
REXX
PS/2
SAA
System i
WebSphere
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Printed in USA