DB2

IBM

DB2 Version 9
for Linux, UNIX, and Windows

National Language Support Guide and Reference

SC10-4380-00

DB2®

IBM

DB2 Version 9
for Linux, UNIX, and Windows

National Language Support Guide and Reference

SC10-4380-00

Before using this information and the product it supports, be sure to read the general information under *Notices*.

Edition Notice

This document contains proprietary information of IBM. It is provided under a license agreement and is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

To order DB2 publications from DB2 Marketing and Sales in the United States or Canada, call 1-800-IBM-4YOU (426-4968).

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

Contents

Chapter 1. National language support (NLS)

National language versions

DB2 Database for Linux, UNIX, and Windows Version 9.1 is available in Simplified Chinese, Traditional Chinese, Czech, Danish, English, Finnish, French, German, Italian, Japanese, Korean, Norwegian, Polish, Brazilian Portuguese, Russian, Spanish, and Swedish.

The DB2 Runtime Client is available in these additional languages: Arabic, Bulgarian, Croatian, Dutch, Greek, Hebrew, Hungarian, Portuguese, Romanian, Slovak, Slovenian, and Turkish.

Related reference:

• "Supported territory codes and code pages" on page 1

Supported territory codes and code pages

The following tables show the languages and code sets supported by the database servers, and how these values are mapped to territory code and code page values that are used by the database manager.

Note: When creating a database, you can use any supported code page on any supported platform.

The following is an explanation of the columns in the tables:

- **Code page** shows the IBM-defined code page as mapped from the operating system code set.
- **Group** shows whether a code page is single-byte ("S"), double-byte ("D"), or neutral ("N"). The "-n" is a number used to create a letter-number combination. Matching combinations show where connection and conversion is allowed by DB2® database systems. For example, all "S-1" groups can work together. However, if the group is neutral, then connection and conversion with any other code page listed is allowed.
- **Code set** shows the code set associated with the supported language. The code set is mapped to the DB2 code page.
- **Territory code** shows the code that is used by the database manager internally to provide region-specific support.
- **Locale** shows the locale values supported by the database manager.
- **Operating system** shows the operating system that supports the languages and code sets. When used in this column, the word "host" refers to an operating system such as z/OS® that supports the EBCDIC code pages natively. Note that Linux™ on z/OS is not a host platform. You cannot use DB2 database manager to create a database in a host code page, but you can use DB2 database manager to connect to a host database in a supported host code page.

Table 1. Unicode

Code page	Group	Code set	Territory code	Locale	Operating system
1200	N-1	16-bit Unicode	Any	Any	Any
1208	N-1	UTF-8 encoding of Unicode	Any	Any	Any

Table 2. Albania, territory identifier: AL

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	355	sq_AL	AIX®
850	S-1	IBM-850	355	-	AIX
923	S-1	ISO8859-15	355	sq_AL.8859-15	AIX
1208	N-1	UTF-8	355	SQ_AL	AIX
37	S-1	IBM-37	355	-	Host
1140	S-1	IBM-1140	355	-	Host
819	S-1	iso88591	355	-	HP-UX
923	S-1	iso885915	355	-	HP-UX
1051	S-1	roman8	355	-	HP-UX
437	S-1	IBM-437	355	-	OS/2®
850	S-1	IBM-850	355	-	OS/2
819	S-1	ISO8859-1	355	-	Solaris
923	S-1	ISO8859-15	355	-	Solaris
1252	S-1	1252	355	-	Windows®

Table 3. Arabic countries/regions, territory identifier: AA

Code page	Group	Code set	Territory code	Locale	Operating system
1046	S-6	IBM-1046	785	Ar_AA	AIX
1089	S-6	ISO8859-6	785	ar_AA	AIX
1208	N-1	UTF-8	785	AR_AA	AIX
420	S-6	IBM-420	785	-	Host
425	S-6	IBM-425	785	-	Host
1089	S-6	iso88596	785	ar_SA.iso88596	HP-UX
864	S-6	IBM-864	785	-	OS/2
1256	S-6	1256	785	-	Windows

Table 4. Australia, territory identifier: AU

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	61	en_AU	AIX
850	S-1	IBM-850	61	-	AIX
923	S-1	ISO8859-15	61	en_AU.8859-15	AIX
1208	N-1	UTF-8	61	EN_AU	AIX
37	S-1	IBM-37	61	-	Host
1140	S-1	IBM-1140	61	-	Host
819	S-1	iso88591	61	-	HP-UX
923	S-1	iso885915	61	-	HP-UX
1051	S-1	roman8	61	-	HP-UX
437	S-1	IBM-437	61	-	OS/2

Table 4. Australia, territory identifier: AU (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
850	S-1	IBM-850	61	-	OS/2
819	S-1	ISO8859-1	61	en_AU	SCO
819	S-1	ISO8859-1	61	en_AU	Solaris
923	S-1	ISO8859-15	61	-	Solaris
1252	S-1	1252	61	-	Windows

Table 5. Austria, territory identifier: AT

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	43	-	AIX
850	S-1	IBM-850	43	-	AIX
923	S-1	ISO8859-15	43	-	AIX
1208	N-1	UTF-8	43	-	AIX
37	S-1	IBM-37	43	-	Host
1140	S-1	IBM-1140	43	-	Host
819	S-1	iso88591	43	-	HP-UX
923	S-1	iso885915	43	-	HP-UX
1051	S-1	roman8	43	-	HP-UX
819	S-1	ISO-8859-1	43	de_AT	Linux
923	S-1	ISO-8859-15	43	de_AT@euro	Linux
437	S-1	IBM-437	43	-	OS/2
850	S-1	IBM-850	43	-	OS/2
819	S-1	ISO8859-1	43	de_AT	SCO
819	S-1	ISO8859-1	43	de_AT	Solaris
923	S-1	ISO8859-15	43	de_AT.ISO8859-15	Solaris
1252	S-1	1252	43	-	Windows

Table 6. Belarus, territory identifier: BY

Code page	Group	Code set	Territory code	Locale	Operating system
1167	S-5	KOI8-RU	375	–	–
915	S-5	ISO8859-5	375	be_BY	AIX
1208	N-1	UTF-8	375	BE_BY	AIX
1025	S-5	IBM-1025	375	-	Host
1154	S-5	IBM-1154	375	-	Host
915	S-5	ISO8859-5	375	-	OS/2
1131	S-5	IBM-1131	375	-	OS/2
1251	S-5	1251	375	-	Windows

Table 7. Belgium, territory identifier: BE

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	32	fr_BE	AIX
819	S-1	ISO8859-1	32	nl_BE	AIX
850	S-1	IBM-850	32	Fr_BE	AIX
850	S-1	IBM-850	32	Nl_BE	AIX
923	S-1	ISO8859-15	32	fr_BE.8859-15	AIX
923	S-1	ISO8859-15	32	nl_BE.8859-15	AIX
1208	N-1	UTF-8	32	FR_BE	AIX

Table 7. Belgium, territory identifier: BE (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
1208	N-1	UTF-8	32	NL_BE	AIX
274	S-1	IBM-274	32	-	Host
500	S-1	IBM-500	32	-	Host
1148	S-1	IBM-1148	32	-	Host
819	S-1	iso88591	32	-	HP-UX
923	S-1	iso885915	32	-	HP-UX
819	S-1	ISO-8859-1	32	fr_BE	Linux
819	S-1	ISO-8859-1	32	nl_BE	Linux
923	S-1	ISO-8859-15	32	fr_BE@euro	Linux
923	S-1	ISO-8859-15	32	nl_BE@euro	Linux
437	S-1	IBM-437	32	-	OS/2
850	S-1	IBM-850	32	-	OS/2
819	S-1	ISO8859-1	32	fr_BE	SCO
819	S-1	ISO8859-1	32	nl_BE	SCO
819	S-1	ISO8859-1	32	fr_BE	Solaris
819	S-1	ISO8859-1	32	nl_BE	Solaris
923	S-1	ISO8859-15	32	fr_BE.ISO8859-15	Solaris
923	S-1	ISO8859-15	32	nl_BE.ISO8859-15	Solaris
1252	S-1	1252	32	-	Windows

Table 8. Bulgaria, territory identifier: BG

Code page	Group	Code set	Territory code	Locale	Operating system
915	S-5	ISO8859-5	359	bg_BG	AIX
1208	N-1	UTF-8	359	BG_BG	AIX
1025	S-5	IBM-1025	359	-	Host
1154	S-5	IBM-1154	359	-	Host
915	S-5	iso88595	359	bg_BG.iso88595	HP-UX
855	S-5	IBM-855	359	-	OS/2
915	S-5	ISO8859-5	359	-	OS/2
1251	S-5	1251	359	-	Windows

Table 9. Brazil, territory identifier: BR

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	55	pt_BR	AIX
850	S-1	IBM-850	55	-	AIX
923	S-1	ISO8859-15	55	pt_BR.8859-15	AIX
1208	N-1	UTF-8	55	PT_BR	AIX
37	S-1	IBM-37	55	-	Host
1140	S-1	IBM-1140	55	-	Host
819	S-1	ISO8859-1	55	-	HP-UX
923	S-1	ISO8859-15	55	-	HP-UX
819	S-1	ISO-8859-1	55	pt_BR	Linux
923	S-1	ISO-8859-15	55	-	Linux
850	S-1	IBM-850	55	-	OS/2
819	S-1	ISO8859-1	55	pt_BR	SCO
819	S-1	ISO8859-1	55	pt_BR	Solaris
923	S-1	ISO8859-15	55	-	Solaris
1252	S-1	1252	55	-	Windows

Table 10. Canada, territory identifier: CA

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	1	fr_CA	AIX
850	S-1	IBM-850	1	Fr_CA	AIX
923	S-1	ISO8859-15	1	fr_CA.8859-15	AIX
1208	N-1	UTF-8	1	FR_CA	AIX
37	S-1	IBM-37	1	-	Host
1140	S-1	IBM-1140	1	-	Host
819	S-1	iso88591	1	fr_CA.iso88591	HP-UX
923	S-1	iso885915	1	-	HP-UX
1051	S-1	roman8	1	fr_CA.roman8	HP-UX
819	S-1	ISO-8859-1	1	en_CA	Linux
923	S-1	ISO-8859-15	1	-	Linux
850	S-1	IBM-850	1	-	OS/2
819	S-1	ISO8859-1	1	en_CA	SCO
819	S-1	ISO8859-1	1	fr_CA	SCO
819	S-1	ISO8859-1	1	en_CA	Solaris
923	S-1	ISO8859-15	1	-	Solaris
1252	S-1	1252	1	-	Windows

Table 11. Canada (French), territory identifier: CA

Code page	Group	Code set	Territory code	Locale	Operating system
863	S-1	IBM-863	2	-	OS/2

Table 12. China (PRC), territory identifier: CN

Code page	Group	Code set	Territory code	Locale	Operating system
1383	D-4	IBM-eucCN	86	zh_CN	AIX
1386	D-4	GBK	86	Zh_CN.GBK	AIX
1208	N-1	UTF-8	86	ZH_CN	AIX
935	D-4	IBM-935	86	-	Host
1388	D-4	IBM-1388	86	-	Host
1383	D-4	hp15CN	86	zh_CN.hp15CN	HP-UX
1386	D-4	GBK	86	zh_CN.GBK	Linux
1381	D-4	IBM-1381	86	-	OS/2
1386	D-4	GBK	86	-	OS/2
1383	D-4	eucCN	86	zh_CN	SCO
1383	D-4	eucCN	86	zh_CN.eucCN	SCO
1383	D-4	gb2312	86	zh	Solaris
1208	N-1	UTF-8	86	zh.UTF-8	Solaris
1381	D-4	IBM-1381	86	-	Windows
1386	D-4	GBK	86	-	Windows
1392/5488	D-4		86	-	

See note 1 on page 20.

Table 13. Croatia, territory identifier: HR

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	385	hr_HR	AIX
1208	N-1	UTF-8	385	HR_HR	AIX
870	S-2	IBM-870	385	-	Host
1153	S-2	IBM-1153	385	-	Host
912	S-2	iso88592	385	hr_HR.iso88592	HP-UX
912	S-2	ISO-8859-2	385	hr_HR	Linux
852	S-2	IBM-852	385	-	OS/2
912	S-2	ISO8859-2	385	hr_HR.ISO8859-2	SCO
1250	S-2	1250	385	-	Windows

Table 14. Czech Republic, territory identifier: CZ

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	421	cs_CZ	AIX
1208	N-1	UTF-8	421	CS_CZ	AIX
870	S-2	IBM-870	421	-	Host
1153	S-2	IBM-1153	421	-	Host
912	S-2	iso88592	421	cs_CZ.iso88592	HP-UX
912	S-2	ISO-8859-2	421	cs_CZ	Linux
852	S-2	IBM-852	421	-	OS/2
912	S-2	ISO8859-2	421	cs_CZ.ISO8859-2	SCO
1250	S-2	1250	421	-	Windows

Table 15. Denmark, territory identifier: DK

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	45	da_DK	AIX
850	S-1	IBM-850	45	Da_DK	AIX
923	S-1	ISO8859-15	45	da_DK.8859-15	AIX
1208	N-1	UTF-8	45	DA_DK	AIX
277	S-1	IBM-277	45	-	Host
1142	S-1	IBM-1142	45	-	Host
819	S-1	iso88591	45	da_DK.iso88591	HP-UX
923	S-1	iso885915	45	_	HP-UX
1051	S-1	roman8	45	da_DK.roman8	HP-UX
819	S-1	ISO-8859-1	45	da_DK	Linux
923	S-1	ISO-8859-15	45	-	Linux
850	S-1	IBM-850	45	-	OS/2
819	S-1	ISO8859-1	45	da	SCO
819	S-1	ISO8859-1	45	da_DA	SCO
819	S-1	ISO8859-1	45	da_DK	SCO
819	S-1	ISO8859-1	45	da	Solaris
923	S-1	ISO8859-15	45	da.ISO8859-15	Solaris
1252	S-1	1252	45	-	Windows

Table 16. Estonia, territory identifier: EE

Code page	Group	Code set	Territory code	Locale	Operating system
922	S-10	IBM-922	372	Et_EE	AIX

Table 16. Estonia, territory identifier: EE (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
1208	N-1	UTF-8	372	ET_EE	AIX
1122	S-10	IBM-1122	372	-	Host
1157	S-10	IBM-1157	372	-	Host
922	S-10	IBM-922	372	-	OS/2
1257	S-10	1257	372	-	Windows

Table 17. Finland, territory identifier: FI

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	358	fi_FI	AIX
850	S-1	IBM-850	358	Fi_FI	AIX
923	S-1	ISO8859-15	358	fi_FI.8859-15	AIX
1208	N-1	UTF-8	358	FI_FI	AIX
278	S-1	IBM-278	358	-	Host
1143	S-1	IBM-1143	358	-	Host
819	S-1	iso88591	358	fi_FI.iso88591	HP-UX
923	S-1	iso885915	358	-	HP-UX
1051	S-1	roman8	358	fi-FI.roman8	HP-UX
819	S-1	ISO-8859-1	358	fi_FI	Linux
923	S-1	ISO-8859-15	358	fi_FI@euro	Linux
437	S-1	IBM-437	358	-	OS/2
850	S-1	IBM-850	358	-	OS/2
819	S-1	ISO8859-1	358		SCO
819	S-1	ISO8859-1	358	fi_FI	SCO
819	S-1	ISO8859-1	358	sv_FI	SCO
819	S-1	ISO8859-1	358	-	Solaris
923	S-1	ISO8859-15	358	fi.ISO8859-15	Solaris
1252	S-1	1252	358	-	Windows

Table 18. FYR Macedonia, territory identifier: MK

Code page	Group	Code set	Territory code	Locale	Operating system
915	S-5	ISO8859-5	389	mk_MK	AIX
1208	N-1	UTF-8	389	MK_MK	AIX
1025	S-5	IBM-1025	389	-	Host
1154	S-5	IBM-1154	389	-	Host
915	S-5	iso88595	389	-	HP-UX
855	S-5	IBM-855	389	-	OS/2
915	S-5	ISO8859-5	389	-	OS/2
1251	S-5	1251	389	-	Windows

Table 19. France, territory identifier: FR

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	33	fr_FR	AIX
850	S-1	IBM-850	33	Fr_FR	AIX
923	S-1	ISO8859-15	33	fr_FR.8859-15	AIX
1208	N-1	UTF-8	33	FR_FR	AIX
297	S-1	IBM-297	33	-	Host

Table 19. France, territory identifier: FR (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
1147	S-1	IBM-1147	33	-	Host
819	S-1	iso88591	33	fr_FR.iso88591	HP-UX
923	S-1	iso885915	33	-	HP-UX
1051	S-1	roman8	33	fr_FR.roman8	HP-UX
819	S-1	ISO-8859-1	33	fr_FR	Linux
923	S-1	ISO-8859-15	33	fr_FR@euro	Linux
437	S-1	IBM-437	33	-	OS/2
850	S-1	IBM-850	33	-	OS/2
819	S-1	ISO8859-1	33		SCO
819	S-1	ISO8859-1	33	fr_FR	SCO
819	S-1	ISO8859-1	33		Solaris
923	S-1	ISO8859-15	33	fr.ISO8859-15	Solaris
1208	N-1	UTF-8	33	fr.UTF-8	Solaris
1252	S-1	1252	33	-	Windows

Table 20. Germany, territory identifier: DE

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	49	de_DE	AIX
850	S-1	IBM-850	49	De_DE	AIX
923	S-1	ISO8859-15	49	de_DE.8859-15	AIX
1208	N-1	UTF-8	49	DE_DE	AIX
273	S-1	IBM-273	49	-	Host
1141	S-1	IBM-1141	49	-	Host
819	S-1	iso88591	49	de_DE.iso88591	HP-UX
923	S-1	iso885915	49	_	HP-UX
1051	S-1	roman8	49	de_DE.roman8	HP-UX
819	S-1	ISO-8859-1	49	de_DE	Linux
923	S-1	ISO-8859-15	49	de_DE@euro	Linux
437	S-1	IBM-437	49	-	OS/2
850	S-1	IBM-850	49	-	OS/2
819	S-1	ISO8859-1	49		SCO
819	S-1	ISO8859-1	49	de_DE	SCO
819	S-1	ISO8859-1	49		Solaris
923	S-1	ISO8859-15	49	de.ISO8859-15	Solaris
1208	N-1	UTF-8	49	de.UTF-8	Solaris
1252	S-1	1252	49	-	Windows

Table 21. Greece, territory identifier: GR

Code page	Group	Code set	Territory code	Locale	Operating system
813	S-7	ISO8859-7	30	el_GR	AIX
1208	N-1	UTF-8	30	EL_GR	AIX
423	S-7	IBM-423	30	-	Host
875	S-7	IBM-875	30	-	Host
813	S-7	iso88597	30	el_GR.iso88597	HP-UX
813	S-7	ISO-8859-7	30	el_GR	Linux
813	S-7	ISO8859-7	30	-	OS/2
869	S-7	IBM-869	30	-	OS/2
813	S-7	ISO8859-7	30	el_GR.ISO8859-7	SCO

Table 21. Greece, territory identifier: GR (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
737	S-7	737	30	-	Windows
1253	S-7	1253	30	-	Windows

Table 22. Hungary, territory identifier: HU

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	36	hu_HU	AIX
1208	N-1	UTF-8	36	HU_HU	AIX
870	S-2	IBM-870	36	-	Host
1153	S-2	IBM-1153	36	-	Host
912	S-2	iso88592	36	hu_HU.iso88592	HP-UX
912	S-2	ISO-8859-2	36	hu_HU	Linux
852	S-2	IBM-852	36	-	OS/2
912	S-2	ISO8859-2	36	hu_HU.ISO8859-2	SCO
1250	S-2	1250	36	-	Windows

Table 23. Iceland, territory identifier: IS

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	354	is_IS	AIX
850	S-1	IBM-850	354	Is_IS	AIX
923	S-1	ISO8859-15	354	is_IS.8859-15	AIX
1208	N-1	UTF-8	354	IS_IS	AIX
871	S-1	IBM-871	354	-	Host
1149	S-1	IBM-1149	354	-	Host
819	S-1	iso88591	354	is_IS.iso88591	HP-UX
923	S-1	iso885915	354	-	HP-UX
1051	S-1	roman8	354	is_IS.roman8	HP-UX
819	S-1	ISO-8859-1	354	is_IS	Linux
923	S-1	ISO-8859-15	354	-	Linux
850	S-1	IBM-850	354	-	OS/2
819	S-1	ISO8859-1	354		SCO
819	S-1	ISO8859-1	354	is_IS	SCO
819	S-1	ISO8859-1	354	-	Solaris
923	S-1	ISO8859-15	354	-	Solaris
1252	S-1	1252	354	-	Windows

Table 24. India, territory identifier: IN

Code page	Group	Code set	Territory code	Locale	Operating system
806	S-13	IBM-806	91	hi_IN	-
1137	S-13	IBM-1137	91	-	Host

Table 25. Indonesia, territory identifier: ID

Code page	Group	Code set	Territory code	Locale	Operating system
1252	S-1	1252	62	-	Windows

Table 26. Ireland, territory identifier: IE

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	353	-	AIX
850	S-1	IBM-850	353	-	AIX
923	S-1	ISO8859-15	353	-	AIX
1208	N-1	UTF-8	353	-	AIX
285	S-1	IBM-285	353	-	Host
1146	S-1	IBM-1146	353	-	Host
819	S-1	iso88591	353	-	HP-UX
923	S-1	iso885915	353	-	HP-UX
1051	S-1	roman8	353	-	HP-UX
819	S-1	ISO-8859-1	353	en_IE	Linux
923	S-1	ISO-8859-15	353	en_IE@euro	Linux
437	S-1	IBM-437	353	-	OS/2
850	S-1	IBM-850	353	-	OS/2
819	S-1	ISO8859-1	353	en_IE.ISO8859-1	SCO
819	S-1	ISO8859-1	353	en_IE	Solaris
923	S-1	ISO8859-15	353	en_IE.ISO8859-15	Solaris
1252	S-1	1252	353	-	Windows

Table 27. Israel, territory identifier: IL

Code page	Group	Code set	Territory code	Locale	Operating system
856	S-8	IBM-856	972	Iw_IL	AIX
916	S-8	ISO8859-8	972	iw_IL	AIX
1208	N-1	UTF-8	972	HE-IL	AIX
916	S-8	ISO-8859-8	972	iw_IL	Linux
424	S-8	IBM-424	972	-	Host
862	S-8	IBM-862	972	-	OS/2
1255	S-8	1255	972	-	Windows

Table 28. Italy, territory identifier: IT

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	39	it_IT	AIX
850	S-1	IBM-850	39	It_IT	AIX
923	S-1	ISO8859-15	39	it_IT.8859-15	AIX
1208	N-1	UTF-8	39	It_IT	AIX
280	S-1	IBM-280	39	-	Host
1144	S-1	IBM-1144	39	-	Host
819	S-1	iso88591	39	it_IT.iso88591	HP-UX
923	S-1	iso885915	39	_	HP-UX
1051	S-1	roman8	39	it_IT.roman8	HP-UX
819	S-1	ISO-8859-1	39	it_IT	Linux
923	S-1	ISO-8859-15	39	it_IT@euro	Linux
437	S-1	IBM-437	39	-	OS/2
850	S-1	IBM-850	39	-	OS/2
819	S-1	ISO8859-1	39		SCO
819	S-1	ISO8859-1	39	it_IT	SCO
819	S-1	ISO8859-1	39		Solaris
923	S-1	ISO8859-15	39	it.ISO8859-15	Solaris
1208	N-1	UTF-8	39	it.UTF-8	Solaris

Table 28. Italy, territory identifier: IT (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
1252	S-1	1252	39	-	Windows

Table 29. Japan, territory identifier: JP

Code page	Group	Code set	Territory code	Locale	Operating system
932	D-1	IBM-932	81	Ja_JP	AIX
943	D-1	IBM-943	81	Ja_JP	AIX
See note 2 on page 20.					
954	D-1	IBM-eucJP	81	ja_JP	AIX
1208	N-1	UTF-8	81	JA_JP	AIX
930	D-1	IBM-930	81	-	Host
939	D-1	IBM-939	81	-	Host
5026	D-1	IBM-5026	81	-	Host
5035	D-1	IBM-5035	81	-	Host
1390	D-1		81	-	Host
1399	D-1		81	-	Host
954	D-1	eucJP	81	ja_JP.eucJP	HP-UX
5039	D-1	SJIS	81	ja_JP.SJIS	HP-UX
954	D-1	EUC-JP	81	ja_JP	Linux
932	D-1	IBM-932	81	-	OS/2
942	D-1	IBM-942	81	-	OS/2
943	D-1	IBM-943	81	-	OS/2
954	D-1	eucJP	81	ja	SCO
954	D-1	eucJP	81	ja_JP	SCO
954	D-1	eucJP	81	ja_JP.EUC	SCO
954	D-1	eucJP	81	ja_JP.eucJP	SCO
943	D-1	IBM-943	81	ja_JP.PCK	Solaris
954	D-1	eucJP	81	ja	Solaris
1208	N-1	UTF-8	81	ja_JP.UTF-8	Solaris
943	D-1	IBM-943	81	-	Windows
1394	D-1		81	-	
See note 3 on page 20.					

Table 30. Kazakhstan, territory identifier: KZ

Code page	Group	Code set	Territory code	Locale	Operating system
1251	S-5	1251	7	-	Windows

Table 31. Korea, South, territory identifier: KR

Code page	Group	Code set	Territory code	Locale	Operating system
970	D-3	IBM-eucKR	82	ko_KR	AIX
1208	N-1	UTF-8	82	KO_KR	AIX
933	D-3	IBM-933	82	-	Host
1364	D-3	IBM-1364	82	-	Host
970	D-3	eucKR	82	ko_KR.eucKR	HP-UX
970	D-3	EUC-KR	82	ko_KR	Linux
949	D-3	IBM-949	82	-	OS/2
970	D-3	eucKR	82	ko_KR.eucKR	SGI

Table 31. Korea, South, territory identifier: KR (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
970	D-3	5601	82	ko	Solaris
1208	N-1	UTF-8	82	ko.UTF-8	Solaris
1363	D-3	1363	82	-	Windows

Table 32. Latin America, territory identifier: Lat

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	3	-	AIX
850	S-1	IBM-850	3	-	AIX
923	S-1	ISO8859-15	3	-	AIX
1208	N-1	UTF-8	3	-	AIX
284	S-1	IBM-284	3	-	Host
1145	S-1	IBM-1145	3	-	Host
819	S-1	iso88591	3	-	HP-UX
923	S-1	iso885915	3	-	HP-UX
1051	S-1	roman8	3	-	HP-UX
819	S-1	ISO-8859-1	3	-	Linux
923	S-1	ISO-8859-15	3	-	Linux
437	S-1	IBM-437	3	-	OS/2
850	S-1	IBM-850	3	-	OS/2
819	S-1	ISO8859-1	3	-	Solaris
923	S-1	ISO8859-15	3	-	Solaris
1252	S-1	1252	3	-	Windows

Table 33. Latvia, territory identifier: LV

Code page	Group	Code set	Territory code	Locale	Operating system
921	S-10	IBM-921	371	Lv_LV	AIX
1208	N-1	UTF-8	371	LV_LV	AIX
1112	S-10	IBM-1112	371	-	Host
1156	S-10	IBM-1156	371	-	Host
921	S-10	IBM-921	371	-	OS/2
1257	S-10	1257	371	-	Windows

Table 34. Lithunia, territory identifier: LT

Code page	Group	Code set	Territory code	Locale	Operating system
921	S-10	IBM-921	370	Lt_LT	AIX
1208	N-1	UTF-8	370	LT_LT	AIX
1112	S-10	IBM-1112	370	-	Host
1156	S-10	IBM-1156	370	-	Host
921	S-10	IBM-921	370	-	OS/2
1257	S-10	1257	370	-	Windows

Table 35. Malaysia, territory identifier: ID

Code page	Group	Code set	Territory code	Locale	Operating system
1252	S-1	1252	60	-	Windows

Table 36. Netherlands, territory identifier: NL

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	31	nl_NL	AIX
850	S-1	IBM-850	31	Nl_NL	AIX
923	S-1	ISO8859-15	31	nl_NL.8859-15	AIX
1208	N-1	UTF-8	31	NL_NL	AIX
37	S-1	IBM-37	31	-	Host
1140	S-1	IBM-1140	31	-	Host
819	S-1	iso88591	31	nl_NL.iso88591	HP-UX
923	S-1	iso885915	31	_	HP-UX
1051	S-1	roman8	31	nl_NL.roman8	HP-UX
819	S-1	ISO-8859-1	31	nl_NL	Linux
923	S-1	ISO-8859-15	31	nl_NL@euro	Linux
437	S-1	IBM-437	31	-	OS/2
850	S-1	IBM-850	31	-	OS/2
819	S-1	ISO8859-1	31	nl	SCO
819	S-1	ISO8859-1	31	nl_NL	SCO
819	S-1	ISO8859-1	31	nl	Solaris
923	S-1	ISO8859-15	31	nl.ISO8859-15	Solaris
1252	S-1	1252	31	-	Windows

Table 37. New Zealand, territory identifier: NZ

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	64	-	AIX
850	S-1	IBM-850	64	-	AIX
923	S-1	ISO8859-15	64	-	AIX
1208	N-1	UTF-8	64	-	AIX
37	S-1	IBM-37	64	-	Host
1140	S-1	IBM-1140	64	-	Host
819	S-1	ISO8859-1	64	-	HP-UX
923	S-1	ISO8859-15	64	-	HP-UX
850	S-1	IBM-850	64	-	OS/2
819	S-1	ISO8859-1	64	en_NZ	SCO
819	S-1	ISO8859-1	64	en_NZ	Solaris
923	S-1	ISO8859-15	64	-	Solaris
1252	S-1	1252	64	-	Windows

Table 38. Norway, territory identifier: NO

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	47	no_NO	AIX
850	S-1	IBM-850	47	No_NO	AIX
923	S-1	ISO8859-15	47	no_NO.8859-15	AIX
1208	N-1	UTF-8	47	NO_NO	AIX
277	S-1	IBM-277	47	-	Host
1142	S-1	IBM-1142	47	-	Host
819	S-1	iso88591	47	no_NO.iso88591	HP-UX
923	S-1	iso885915	47	-	HP-UX
1051	S-1	roman8	47	no_NO.roman8	HP-UX

Table 38. Norway, territory identifier: NO (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO-8859-1	47	no_NO	Linux
923	S-1	ISO-8859-15	47	-	Linux
850	S-1	IBM-850	47	-	OS/2
819	S-1	ISO8859-1	47	no	SCO
819	S-1	ISO8859-1	47	no_NO	SCO
819	S-1	ISO8859-1	47	no	Solaris
923	S-1	ISO8859-15	47	-	Solaris
1252	S-1	1252	47	-	Windows

Table 39. Poland, territory identifier: PL

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	48	pl_PL	AIX
1208	N-1	UTF-8	48	PL_PL	AIX
870	S-2	IBM-870	48	-	Host
1153	S-2	IBM-1153	48	-	Host
912	S-2	iso88592	48	pl_PL.iso88592	HP-UX
912	S-2	ISO-8859-2	48	pl_PL	Linux
852	S-2	IBM-852	48	-	OS/2
912	S-2	ISO8859-2	48	pl_PL.ISO8859-2	SCO
1250	S-2	1250	48	-	Windows

Table 40. Portugal, territory identifier: PT

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	351	pt_PT	AIX
850	S-1	IBM-850	351	Pt_PT	AIX
923	S-1	ISO8859-15	351	pt_PT.8859-15	AIX
1208	N-1	UTF-8	351	PT_PT	AIX
37	S-1	IBM-37	351	-	Host
1140	S-1	IBM-1140	351	-	Host
819	S-1	iso88591	351	pt_PT.iso88591	HP-UX
923	S-1	iso885915	351	-	HP-UX
1051	S-1	roman8	351	pt_PT.roman8	HP-UX
819	S-1	ISO-8859-1	351	pt_PT	Linux
923	S-1	ISO-8859-15	351	pt_PT@euro	Linux
850	S-1	IBM-850	351	-	OS/2
860	S-1	IBM-860	351	-	OS/2
819	S-1	ISO8859-1	351	pt	SCO
819	S-1	ISO8859-1	351	pt_PT	SCO
819	S-1	ISO8859-1	351	pt	Solaris
923	S-1	ISO8859-15	351	pt.ISO8859-15	Solaris
1252	S-1	1252	351	-	Windows

Table 41. Romania, territory identifier: RO

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	40	ro_RO	AIX
1208	N-1	UTF-8	40	RO_RO	AIX

Table 41. Romania, territory identifier: RO (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
870	S-2	IBM-870	40	-	Host
1153	S-2	IBM-1153	40	-	Host
912	S-2	iso88592	40	ro_RO.iso88592	HP-UX
912	S-2	ISO-8859-2	40	ro_RO	Linux
852	S-2	IBM-852	40	-	OS/2
912	S-2	ISO8859-2	40	ro_RO.ISO8859-2	SCO
1250	S-2	1250	40	-	Windows

Table 42. Russia, territory identifier: RU

Code page	Group	Code set	Territory code	Locale	Operating system
915	S-5	ISO8859-5	7	ru_RU	AIX
1208	N-1	UTF-8	7	RU_RU	AIX
1025	S-5	IBM-1025	7	-	Host
1154	S-5	IBM-1154	7	-	Host
915	S-5	iso88595	7	ru_RU.iso88595	HP-UX
878	S-5	KOI8-R	7	ru_RU.koi8-r	Linux, Solaris
915	S-5	ISO-8859-5	7	ru_RU	Linux
866	S-5	IBM-866	7	-	OS/2
915	S-5	ISO8859-5	7	-	OS/2
915	S-5	ISO8859-5	7	ru_RU.ISO8859-5	SCO
1251	S-5	1251	7	-	Windows

Table 43. Serbia/Montenegro, territory identifier: SP

Code page	Group	Code set	Territory code	Locale	Operating system
915	S-5	ISO8859-5	381	sr_SP	AIX
1208	N-1	UTF-8	381	SR_SP	AIX
1025	S-5	IBM-1025	381	-	Host
1154	S-5	IBM-1154	381	-	Host
915	S-5	iso88595	381	-	HP-UX
855	S-5	IBM-855	381	-	OS/2
915	S-5	ISO8859-5	381	-	OS/2
1251	S-5	1251	381	-	Windows

Table 44. Slovakia, territory identifier: SK

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	422	sk_SK	AIX
1208	N-1	UTF-8	422	SK_SK	AIX
870	S-2	IBM-870	422	-	Host
1153	S-2	IBM-1153	422	-	Host
912	S-2	iso88592	422	sk_SK.iso88592	HP-UX
852	S-2	IBM-852	422	-	OS/2
912	S-2	ISO8859-2	422	sk_SK.ISO8859-2	SCO
1250	S-2	1250	422	-	Windows

Table 45. Slovenia, territory identifier: SI

Code page	Group	Code set	Territory code	Locale	Operating system
912	S-2	ISO8859-2	386	sl_SI	AIX
1208	N-1	UTF-8	386	SL_SI	AIX
870	S-2	IBM-870	386	-	Host
1153	S-2	IBM-1153	386	-	Host
912	S-2	iso88592	386	sl_SI.iso88592	HP-UX
912	S-2	ISO-8859-2	386	sl_SI	Linux
852	S-2	IBM-852	386	-	OS/2
912	S-2	ISO8859-2	386	sl_SI.ISO8859-2	SCO
1250	S-2	1250	386	-	Windows

Table 46. South Africa, territory identifier: ZA

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	27	en_ZA	AIX
850	S-1	IBM-850	27	En_ZA	AIX
923	S-1	ISO8859-15	27	en_ZA.8859-15	AIX
1208	N-1	UTF-8	27	EN_ZA	AIX
285	S-1	IBM-285	27	-	Host
1146	S-1	IBM-1146	27	-	Host
819	S-1	iso88591	27	-	HP-UX
923	S-1	iso885915	27	-	HP-UX
1051	S-1	roman8	27	-	HP-UX
437	S-1	IBM-437	27	-	OS/2
850	S-1	IBM-850	27	-	OS/2
819	S-1	ISO8859-1	27	en_ZA.ISO8859-1	SCO
819	S-1	ISO8859-1	27	-	Solaris
923	S-1	ISO8859-15	27	-	Solaris
1252	S-1	1252	27	-	Windows

Table 47. Spain, territory identifier: ES

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	34	es_ES	AIX
850	S-1	IBM-850	34	Es_ES	AIX
923	S-1	ISO8859-15	34	es_ES.8859-15	AIX
1208	N-1	UTF-8	34	ES_ES	AIX
284	S-1	IBM-284	34	-	Host
1145	S-1	IBM-1145	34	-	Host
819	S-1	iso88591	34	es_ES.iso88591	HP-UX
923	S-1	iso885915	34	-	HP-UX
1051	S-1	roman8	34	es_ES.roman8	HP-UX
819	S-1	ISO-8859-1	34	es_ES	Linux
923	S-1	ISO-8859-15	34	es_ES@euro	Linux
437	S-1	IBM-437	34	-	OS/2
850	S-1	IBM-850	34	-	OS/2
819	S-1	ISO8859-1	34	es	SCO
819	S-1	ISO8859-1	34	es_ES	SCO
819	S-1	ISO8859-1	34	es	Solaris
923	S-1	ISO8859-15	34	es.ISO8859-15	Solaris
1208	N-1	UTF-8	34	es.UTF-8	Solaris

Table 47. Spain, territory identifier: ES (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
1252	S-1	1252	34	-	Windows

Table 48. Spain (Catalan), territory identifier: ES

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	34	ca_ES	AIX
850	S-1	IBM-850	34	Ca_ES	AIX
923	S-1	ISO8859-15	34	ca_ES.8859-15	AIX
1208	N-1	UTF-8	34	CA_ES	AIX

Table 49. Sweden, territory identifier: SE

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	46	sv_SE	AIX
850	S-1	IBM-850	46	Sv_SE	AIX
923	S-1	ISO8859-15	46	sv_SE.8859-15	AIX
1208	N-1	UTF-8	46	SV_SE	AIX
278	S-1	IBM-278	46	-	Host
1143	S-1	IBM-1143	46	-	Host
819	S-1	iso88591	46	sv_SE.iso88591	HP-UX
923	S-1	iso885915	46	-	HP-UX
1051	S-1	roman8	46	sv_SE.roman8	HP-UX
819	S-1	ISO-8859-1	46	sv_SE	Linux
923	S-1	ISO-8859-15	46	-	Linux
437	S-1	IBM-437	46	-	OS/2
850	S-1	IBM-850	46	-	OS/2
819	S-1	ISO8859-1	46	sv	SCO
819	S-1	ISO8859-1	46	sv_SE	SCO
819	S-1	ISO8859-1	46	sv	Solaris
923	S-1	ISO8859-15	46	sv.ISO8859-15	Solaris
1208	N-1	UTF-8	46	sv.UTF-8	Solaris
1252	S-1	1252	46	-	Windows

Table 50. Switzerland, territory identifier: CH

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	41	de_CH	AIX
850	S-1	IBM-850	41	De_CH	AIX
923	S-1	ISO8859-15	41	de_CH.8859-15	AIX
1208	N-1	UTF-8	41	DE_CH	AIX
500	S-1	IBM-500	41	-	Host
1148	S-1	IBM-1148	41	-	Host
819	S-1	iso88591	41	-	HP-UX
923	S-1	iso885915	41	-	HP-UX
1051	S-1	roman8	41	-	HP-UX
819	S-1	ISO-8859-1	41	de_CH	Linux
923	S-1	ISO-8859-15	41	-	Linux
437	S-1	IBM-437	41	-	OS/2
850	S-1	IBM-850	41	-	OS/2

Table 50. Switzerland, territory identifier: CH (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	41	de_CH	SCO
819	S-1	ISO8859-1	41	fr_CH	SCO
819	S-1	ISO8859-1	41	it_CH	SCO
819	S-1	ISO8859-1	41	de_CH	Solaris
923	S-1	ISO8859-15	41	-	Solaris
1252	S-1	1252	41	-	Windows

Table 51. Taiwan, territory identifier: TW

Code page	Group	Code set	Territory code	Locale	Operating system
950	D-2	big5	88	Zh_TW	AIX
See note 8 on page 21.					
964	D-2	IBM-eucTW	88	zh_TW	AIX
1208	N-1	UTF-8	88	ZH_TW	AIX
937	D-2	IBM-937	88	-	Host
1371	D-2	IBM-1371	88	-	Host
950	D-2	big5	88	zh_TW.big5	HP-UX
964	D-2	eucTW	88	zh_TW.eucTW	HP-UX
950	D-2	BIG5	88	zh_TW	Linux
938	D-2	IBM-938	88	-	OS/2
948	D-2	IBM-948	88	-	OS/2
950	D-2	big5	88	-	OS/2
950	D-2	big5	88	zh_TW.BIG5	Solaris
964	D-2	cns11643	88	zh_TW	Solaris
1208	N-1	UTF-8	88	zh_TW.UTF-8	Solaris
950	D-2	big5	88	-	Windows
See note 8 on page 21.					

Table 52. Thailand, territory identifier: TH

Code page	Group	Code set	Territory code	Locale	Operating system
874	S-20	TIS620-1	66	th_TH	AIX
1208	N-1	UTF-8	66	TH_TH	AIX
838	S-20	IBM-838	66	-	Host
1160	S-20	IBM-1160	66	-	Host
874	S-20	tis620	66	th_TH.tis620	HP-UX
874	S-20	TIS620-1	66	-	OS/2
874	S-20	TIS620-1	66	-	Windows

Table 53. Turkey, territory identifier: TR

Code page	Group	Code set	Territory code	Locale	Operating system
920	S-9	ISO8859-9	90	tr_TR	AIX
1208	N-1	UTF-8	90	TR_TR	AIX
1026	S-9	IBM-1026	90	-	Host
1155	S-9	IBM-1155	90	-	Host
920	S-9	iso88599	90	tr_TR.iso88599	HP-UX
920	S-9	ISO-8859-9	90	tr_TR	Linux
857	S-9	IBM-857	90	-	OS/2

Table 53. Turkey, territory identifier: TR (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
920	S-9	ISO8859-9	90	tr_TR.ISO8859-9	SCO
1254	S-9	1254	90	-	Windows

Table 54. United Kingdom, territory identifier: GB

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	44	en_GB	AIX
850	S-1	IBM-850	44	En_GB	AIX
923	S-1	ISO8859-15	44	en_GB.8859-15	AIX
1208	N-1	UTF-8	44	EN_GB	AIX
285	S-1	IBM-285	44	-	Host
1146	S-1	IBM-1146	44	-	Host
819	S-1	iso88591	44	en_GB.iso88591	HP-UX
923	S-1	iso885915	44	-	HP-UX
1051	S-1	roman8	44	en_GB.roman8	HP-UX
819	S-1	ISO-8859-1	44	en_GB	Linux
923	S-1	ISO-8859-15	44	-	Linux
437	S-1	IBM-437	44	-	OS/2
850	S-1	IBM-850	44	-	OS/2
819	S-1	ISO8859-1	44	en_GB	SCO
819	S-1	ISO8859-1	44	en	SCO
819	S-1	ISO8859-1	44	en_GB	Solaris
923	S-1	ISO8859-15	44	en_GB.ISO8859-15	Solaris
1252	S-1	1252	44	-	Windows

Table 55. Ukraine, territory identifier: UA

Code page	Group	Code set	Territory code	Locale	Operating system
1124	S-12	IBM-1124	380	Uk_UA	AIX
1208	N-1	UTF-8	380	UK_UA	AIX
1123	S-12	IBM-1123	380	-	Host
1158	S-12	IBM-1158	380	-	Host
1168	S-12	KOI8-U	380	uk_UA.koi8u	Linux
1125	S-12	IBM-1125	380	-	OS/2
1251	S-12	1251	380	-	Windows

Table 56. United States of America, territory identifier: US

Code page	Group	Code set	Territory code	Locale	Operating system
819	S-1	ISO8859-1	1	en_US	AIX
850	S-1	IBM-850	1	En_US	AIX
923	S-1	ISO8859-15	1	en_US.8859-15	AIX
1208	N-1	UTF-8	1	EN_US	AIX
37	S-1	IBM-37	1	-	Host
1140	S-1	IBM-1140	1	-	Host
819	S-1	iso88591	1	en_US.iso88591	HP-UX
923	S-1	iso885915	1	-	HP-UX
1051	S-1	roman8	1	en_US.roman8	HP-UX
819	S-1	ISO-8859-1	1	en_US	Linux

Table 56. United States of America, territory identifier: US (continued)

Code page	Group	Code set	Territory code	Locale	Operating system
923	S-1	ISO-8859-15	1	-	Linux
437	S-1	IBM-437	1	-	OS/2
850	S-1	IBM-850	1	-	OS/2
819	S-1	ISO8859-1	1	en_US	SCO
819	S-1	ISO8859-1	1	en_US	SGI
819	S-1	ISO8859-1	1	en_US	Solaris
923	S-1	ISO8859-15	1	en_US.ISO8859-15	Solaris
1208	N-1	UTF-8	1	en_US.UTF-8	Solaris
1252	S-1	1252	1	-	Windows

Table 57. Vietnam, territory identifier: VN

Code page	Group	Code set	Territory code	Locale	Operating system
1129	S-11	IBM-1129	84	Vi_VN	AIX
1208	N-1	UTF-8	84	VI_VN	AIX
1130	S-11	IBM-1130	84	-	Host
1164	S-11	IBM-1164	84	-	Host
1129	S-11	IBM-1129	84	-	OS/2
1258	S-11	1258	84	-	Windows

Notes:

1. CCSIDs 1392 and 5488 (GB 18030) can only be used with the load or import utilities to move data from CCSIDs 1392 and 5488 to a DB2 Unicode database, or to export from a DB2 Unicode database to CCSIDs 1392 or 5488.
2. On AIX 4.3 or later the code page is 943. If you are using AIX 4.2 or earlier, the code page is 932.
3. Code page 1394 (Shift JIS X0213) can only be used with the load or import utilities to move data from code page 1394 to a DB2 Unicode database, or to export from a DB2 Unicode database to code page 1394.
4. The following map to Arabic Countries/Regions (AA):
   - Arabic (Saudi Arabia)
   - Arabic (Iraq)
   - Arabic (Egypt)
   - Arabic (Libya)
   - Arabic (Algeria)
   - Arabic (Morocco)
   - Arabic (Tunisia)
   - Arabic (Oman)
   - Arabic (Yemen)
   - Arabic (Syria)
   - Arabic (Jordan)
   - Arabic (Lebanon)
   - Arabic (Kuwait)
   - Arabic (United Arab Emirates)
   - Arabic (Bahrain)
   - Arabic (Qatar)

5. The following map to English (US):
   - English (Jamaica)
   - English (Caribbean)
6. The following map to Latin America (Lat):
   - Spanish (Mexican)
   - Spanish (Guatemala)
   - Spanish (Costa Rica)
   - Spanish (Panama)
   - Spanish (Dominican Republic)
   - Spanish (Venezuela)
   - Spanish (Colombia)
   - Spanish (Peru)
   - Spanish (Argentina)
   - Spanish (Ecuador)
   - Spanish (Chile)
   - Spanish (Uruguay)
   - Spanish (Paraguay)
   - Spanish (Bolivia)
7. The following Indic scripts are supported through Unicode: Hindi, Gujarati, Kannada, Konkani, Marathi, Punjabi, Sanskrit, Tamil and Telugu.
8. Code page 950 is also known as Big5. Microsoft® code page 950 differs from IBM® code page 950 in the following ways:

Range	Description	IBM	Microsoft	Difference
X'8140' to X'8DFE'	User defined characters	User defined area	User defined area	Same
X'8E40' to X'A0FE'	User defined characters	User defined area	User defined area	Same
X'A140' to X'A3BF'	Special symbols	System characters	System characters	Same
X'A3C0' to X'A3E0'	Control symbols	System characters	Empty	Different
X'A3E1' to X'A3FE'	Reserved	Empty	Empty	Same
X'A440' to X'C67E'	Primary use characters	System characters	System characters	Same
X'C6A1' to X'C878'	Eten added symbols	System characters	User defined area	Different
X'C879' to X'C8CC'	Eten added symbols	Empty	User defined area	Different
X'C8CD' to X'C8D3'	Eten added symbols	System characters	User defined area	Different
X'C8D4' to X'C8FD'	Reserved	System characters	User defined area	Different
X'C8FE'	Invalid/ undefined character	System characters	User defined area	Different
X'C940' to X'F9D5'	Secondary use characters	System characters	System characters	Same
X'F9D6' to X'F9FE'	Eten extension for Big-5	User defined area	System characters	Different

Range	Description	IBM	Microsoft	Difference
X'FA40' to X'FEFE'	User defined characters	User defined area	User defined area	Same
X'8181' to X'8C82'	User defined characters	User defined area	Empty	Different
X'F286' to X'F9A0'	IBM select characters	System characters	Empty	Different
Total characters		14 060	13 502	
Total user defined characters		6 204	6 217	
Total defined code points		20 264	19 719	

Related tasks:

- "Installing the previous tables for converting between code page 1394 and Unicode" on page 57

Date and time formats by territory code

The character string representation of date and time formats is the default format of datetime values associated with the territory code of the application. This default format can be overridden by specifying the DATETIME format option when the program is precompiled or bound to the database.

Following is a description of the input and output formats for date and time:

- Input Time Format
  - There is no default input time format
  - All time formats are allowed as input for all territory codes.
- Output Time Format
  - The default output time format is equal to the local time format.
- Input Date Format
  - There is no default input date format
  - Where the local format for date conflicts with an ISO, JIS, EUR, or USA date format, the local format is recognized for date input. For example, see the UK entry in Table 58.
- Output Date Format
  - The default output date format is shown in Table 58.

    Note: Table 58 also shows a listing of the string formats for the various territory codes.

Table 58. Date and Time Formats by Territory Code

Territory Code	Local Date Format	Local Time Format	Default Output Date Format	Input Date Formats
355 Albania	yyyy-mm-dd	JIS	LOC	LOC, USA, EUR, ISO
785 Arabic	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
001 Australia (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO

Table 58. Date and Time Formats by Territory Code (continued)

Territory Code	Local Date Format	Local Time Format	Default Output Date Format	Input Date Formats
061 Australia	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
032 Belgium	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
055 Brazil	dd.mm.yyyy	JIS	LOC	LOC, EUR, ISO
359 Bulgaria	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
001 Canada	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
002 Canada (French)	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
385 Croatia	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 Czech Republic	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
045 Denmark	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
358 Finland	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
389 FYR Macedonia	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
033 France	dd/mm/yyyy	JIS	EUR	LOC, EUR, ISO
049 Germany	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
030 Greece	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
036 Hungary	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
354 Iceland	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
091 India	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
972 Israel	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
039 Italy	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
081 Japan	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
082 Korea	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
001 Latin America (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
003 Latin America	dd-mm-yyyy	JIS	LOC	LOC, EUR, ISO
031 Netherlands	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
047 Norway	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
048 Poland	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
351 Portugal	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO

Table 58. Date and Time Formats by Territory Code (continued)

Territory Code	Local Date Format	Local Time Format	Default Output Date Format	Input Date Formats
086 China	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
040 Romania	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
007 Russia	dd/mm/yyyy	ISO	LOC	LOC, EUR, ISO
381 Serbia/ Montenegro	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 Slovakia	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
386 Slovenia	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
034 Spain	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
046 Sweden	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
041 Switzerland	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
088 Taiwan	mm-dd-yyyy	JIS	ISO	LOC, USA, EUR, ISO
066 Thailand (2)	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
090 Turkey	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
044 UK	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
001 USA	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
084 Vietnam	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO

Notes:

1. Countries/Regions using the default C locale are assigned territory code 001.
2. yyyy in Buddhist era is equivalent to Gregorian + 543 years (Thailand only).

Related reference:

- "BIND command" in *Command Reference*
- "PRECOMPILE command" in *Command Reference*

Simplified Chinese locale coding set

IBM AIX and some distributions of Linux have changed the code set bound to the simplified Chinese locale from GBK (code page 1386) to GB18030 (code page 5488 or 1392). For example, the Zh_CN locale on AIX is now bound to the GB18030 code set since:

- AIX Version 5.1.0000.0011
- AIX Version 5.1.0 with maintenance level 2

DB2 database manager supports the GBK code set natively and the GB18030 code set only through Unicode. DB2 database manager will default the locale's code set to ISO 8859-1 (code page 819), and in some operations will also default the locale's territory to the United States (US). To work around this limitation, you have two options:

1. You can override the locale's code set from GB18030 to GBK; and the territory from US to China (whose territory identifier is CN and territory code is 86).
2. You can use a different simplified Chinese locale.

If you choose to use the first option, issue the following commands:

```
db2set DB2CODEPAGE=1386
db2set DB2TERRITORY=86
db2 terminate
db2stop
db2start
```

If you choose to use the second option on AIX, issue either of the following commands:

```
export LANG=zh_CN
export LANG=ZH_CN
```

The code set associated with zh_CN is eucCN (code page 1383), and with ZH_CN is UTF-8 (code page 1208).

If you choose to use the second option on Linux, issue one of the following commands:

```
export LANG=zh_CN.gbk
export LANG=zh_CN
export LANG=zh_CN.utf8
```

The code set associated with zh_CN is eucCN (code page 1383), and with zh_CN.utf8 is UTF-8 (code page 1208).

Availability of Asian fonts (Linux)

IBM offers additional font packages for Linux that contain additional double-byte character set (DBCS) support for Asian characters. These font packages are necessary with some versions of Linux which install only the fonts required to display the country-specific or region-specific characters.

If you notice missing characters when you use the DB2 Setup wizard or the DB2 GUI tools (post-installation), install the necessary fonts provided with the DB2 product then re-run the **db2setup** command or restart the DB2 GUI tools you were using. The Asian fonts are found in the `java_fonts` directory on the National Language Pack CD-ROM (NLPACK CD) for your Linux operating system.

In this directory, there are two typefaces available: Times New Roman WorldType and Monotype Sans Duospace WorldType. For each typeface, there is a country- or region-specific font. The following table lists the eight fonts provided in compressed format in the `java_fonts` directory.

Font typeface	Font file name	Country/Region
Times New Roman WT J	tnrwt_j.zip	Japan and other countries/regions
Times New Roman WT K	tnrwt_k.zip	Korea
Times New Roman WT SC	tnrwt_s.zip	China (Simplified Chinese)
Times New Roman WT TC	tnrwt_t.zip	Taiwan (Traditional Chinese)
Monotype Sans Duospace WT J	mtsansdj.zip	Japan and other countries/regions

Font typeface	Font file name	Country/Region
Monotype Sans Duospace WT K	mtsansdk.zip	Korea
Monotype Sans Duospace WT SC	mtsansds.zip	China (Simplified Chinese)
Monotype Sans Duospace WT TC	mtsansdt.zip	Taiwan (Traditional Chinese)

Note: These fonts do not replace the system fonts. These fonts are to be used in conjunction with or for use with DB2. You cannot engage in the general or unrestricted sale or distribution of these fonts.

To install a font:

1. Unzip the font package.
2. Copy the font package to the `/opt/jre/lib/fonts` directory. You need to create the directory if it does not already exist.
3. Enter the following command: **export JAVA_FONTS=/opt/jre/lib/fonts**

   Note: Optionally, you can copy the Asian font package into the `java` directory in the DB2 installation path. For example, `<DB2 installation path>/java/jdk32/jre/lib/fonts`, or `<DB2 installation path>/java/jdk64/jre/lib/fonts`.

As a minimum you need to install one font of each typeface for your country or region. If you are in either the China, Korea, or Taiwan territory, use the territory-specific or region-specific versions; otherwise, use the Japanese version of the fonts. If you have space on your system, it is recommended that you install all eight fonts.

Displaying Indic characters in the DB2 GUI tools

If you have problems displaying Indic characters when using the DB2 GUI tools on Linux or UNIX® operating systems, you might not have the required fonts installed on your system.

DB2 has packaged the following IBM TrueType and OpenType proportional Indic language fonts for your use. You can find these fonts in the `java_fonts` directory on the National Language Pack CD-ROM (NLPACK CD) for the Linux and UNIX operating systems.

These fonts are to be used in conjunction with DB2. You cannot engage in the general or unrestricted sale or distribution of these fonts:

Table 59. Indic fonts packaged with DB2

Typeface	Weight	Font File Name
Devanagari MT for IBM	Medium	devamt.ttf
Devanagari MT for IBM	Bold	devamtb.ttf
Tamil	Medium	TamilMT.ttf
Tamil	Bold	TamilMTB.ttf
Telugu	Medium	TeluguMT.ttf
Telugu	Bold	TeleguMTB.ttf

Detailed instructions on how to install the fonts and modify the `font.properties` file can be found in the Internationalization section of the Java™ documentation.

In addition, some Microsoft products also come with Indic fonts that can be used with our GUI tools.

Choosing a language for your database

When you create a database, you have to decide what language your data will be stored in. When you create a database, you can specify the territory and code set. The territory and code set may be different from the current operating system settings. If you do not explicitly choose a territory and code set at database creation time, the database will be created using the current locale. When you are choosing a code set, make sure it can encode all the characters in the language you will be using.

Another option is to store data in a Unicode database, which means that you do not have to choose a specific language; Unicode encoding includes characters from almost all of the living languages in the world.

Locale setting for the DB2 Administration Server

Ensure that the locale of the DB2 Administration Server instance is compatible with the locale of the DB2 instance. Otherwise, the DB2 instance cannot communicate with the DB2 Administration Server.

If the LANG environment variable is not set in the user profile of the DB2 Administration Server, the DB2 Administration Server will be started with the default system locale. If the default system locale is not defined, the DB2 Administration Server will be started with code page 819. If the DB2 instance uses one of the DBCS locales, and the DB2 Administration Server is started with code page 819, the instance will not be able to communicate with the DB2 Administration Server. The locale of the DB2 Administration Server and the locale of the DB2 instance must be compatible.

For example, on a Simplified Chinese Linux system, `LANG=zh_CN` should be set in the DB2 Administration Server's user profile.

Related tasks:

- "Changing the DB2 interface language (Linux and UNIX)" in *Quick Beginnings for DB2 Servers*
- "Changing the DB2 interface language (Windows)" in *Quick Beginnings for DB2 Servers*

Enabling bidirectional support

Bidirectional layout transformations are implemented in DB2 Database for Linux, UNIX, and Windows using the new Coded Character Set Identifier (CCSID) definitions. For the new bidirectional-specific CCSIDs, layout transformations are performed instead of, or in addition to, code page conversions. To use this support, the DB2BIDI registry variable must be set to YES. By default, this variable is not set. It is used by the server for all conversions, and can only be set when the server is started. Setting DB2BIDI to YES may have some performance impact because of additional checking and layout transformations.

Restrictions:

The following restrictions apply:

- If you select a CCSID that is not appropriate for the code page or string type of your client platform, you may get unexpected results. If you select an incompatible CCSID (for example, the Latin-1 CCSID for connection to an Arabic database), or if DB2BIDI has not been set for the server, you will receive an error message when you try to connect.
- The DB2 Command Line Processor on the Windows operating system does not have bidirectional support.
- CCSID override is not supported for cases where the HOST EBCDIC platform is the client, and DB2 Database is the server.

When converting from one Arabic CCSID to another Arabic CCSID, DB2 employs the following logic to deshape (or expand) the lam-alef ligature. Deshaping will occur when the Text Shaping attribute of the source Arabic CCSID is shaped but the Text Shaping attribute of the target Arabic CCSID is unshaped.

The logic to deshape the lam-alef ligature is:

1. If the *last* character of the data stream is a blank character, then every character after the lam-alef ligature will be shifted to the end of the data stream, therefore making available an empty position for the current lam-alef ligature to be deshaped (expanded) into its two constituent characters: lam and alef.
2. Otherwise, if the *first* character of the data stream is a blank character, then every character before the lam-alef ligature will be shifted to the beginning of the data stream, therefore making available an empty position for the current lam-alef ligature to be deshaped (expanded) into its two constituent characters: lam and alef.
3. Otherwise, there is no blank character at the beginning and end of the data stream, and the lam-alef ligature cannot be deshaped. If the target CCSID does have the lam-alef ligature, then the lam-alef ligature remains as is; otherwise, the lam-alef ligature is replaced by the target CCSID's SUBstitution character.

Conversely when converting from an Arabic CCSID whose Text Shaping attribute is unshaped to an Arabic CCSID whose Text Shaping attribute is shaped, the source lam and alef characters will be contracted to one ligature character, and a blank character is inserted at the end of the target area data stream.

Procedure:

To specify a particular bidirectional CCSID in a non-DRDA environment:

- Ensure the DB2BIDI registry variable is set to YES.
- Select the CCSID that matches the characteristics of your client, and set DB2CODEPAGE to that value.
- If you already have a connection to the database, you must issue a TERMINATE command, and then reconnect to allow the new setting for DB2CODEPAGE to take effect.

For DRDA® environments, if the HOST EBCDIC platform also supports these bidirectional CCSIDs, you only need to set the DB2CODEPAGE value. Note that you must not further specify the same CCSID on the BIDI parameter in the PARMS field of the DCS database directory entry for the server database, otherwise an extra bidi layout conversion would occur, and render any Arabic data to be incorrectly reversed. However, if the HOST platform does not support these

CCSIDs, you must also specify a CCSID override for the HOST database server to which you are connecting. This is accomplished through the use of the BIDI parameter in the PARMS field of the DCS database directory entry for the server database. The override is necessary because, in a DRDA environment, code page conversions and layout transformations are performed by the receiver of data. However, if the HOST server does not support these bidirectional CCSIDs, it does not perform layout transformation on the data that it receives from DB2. If you use a CCSID override, the DB2 client performs layout transformation on the outbound data as well.

Related concepts:

- "Bidirectional support with DB2 Connect" on page 32
- "Handling BiDi data" in *DB2 Connect User's Guide*

Related reference:

- "Bidirectional-specific CCSIDs" on page 29
- "General registry variables" in *Administration Guide: Implementation*

Bidirectional-specific CCSIDs

The following bidirectional attributes are required for correct handling of bidirectional data on different platforms:

- Text type
- Numeric shaping
- Orientation
- Text shaping
- Symmetric swapping

Because default values on different platforms are not the same, problems can occur when DB2 data is moved from one platform to another. For example, the Windows operating system uses LOGICAL UNSHAPED data, while z/OS and OS/390® usually use SHAPED VISUAL data. Therefore, without support for bidirectional attributes, data sent from DB2 Universal Database for z/OS and OS/390 to DB2 on Windows 32-bit operating systems may display incorrectly.

DB2 Database for Linux, UNIX, and Windows supports bidirectional data attributes through special bidirectional Coded Character Set Identifiers (CCSIDs). The following bidirectional CCSIDs have been defined and are implemented with DB2 as shown in Table 60. CDRA string types are defined as shown in Table 61 on page 31.

Table 60. Bidirectional CCSIDs

CCSID	Code Page	String Type
420	420	4
424	424	4
856	856	5
862	862	4
864	864	5
867	862	4
916	916	5

Table 60. Bidirectional CCSIDs (continued)

CCSID	Code Page	String Type
1046	1046	5
1089	1089	5
1200	1200	10
1208	1208	10
1255	1255	5
1256	1256	5
5351	1255	5
5352	1256	5
8612	420	5
8616	424	10
9048	856	5
9238	1046	5
12712	424	4
13488	13488	10
16804	420	4
17248	864	5
62208	856	4
62209	862	10
62210	916	4
62211	424	5
62213	862	5
62215	1255	4
62218	864	4
62220	856	6
62221	862	6
62222	916	6
62223	1255	6
62224	420	6
62225	864	6
62226	1046	6
62227	1089	6
62228	1256	6
62229	424	8
62230	856	8
62231	862	8
62232	916	8
62233	420	8
62234	420	9
62235	424	6
62236	856	10

Table 60. Bidirectional CCSIDs (continued)

CCSID	Code Page	String Type
62237	1255	8
62238	916	10
62239	1255	10
62240	424	11
62241	856	11
62242	862	11
62243	916	11
62244	1255	11
62245	424	10
62246	1046	8
62247	1046	9
62248	1046	4
62249	1046	12
62250	420	12

Table 61. CDRA string types

String type	Text type	Numeric shaping	Orientation	Text shaping	Symmetrical swapping
4	Visual	Passthrough	LTR	Shaped	Off
5	Implicit	Arabic	LTR	Unshaped	On
6	Implicit	Arabic	RTL	Unshaped	On
7*	Visual	Passthrough	Contextual*	Unshaped ligature	Off
8	Visual	Passthrough	RTL	Shaped	Off
9	Visual	Passthrough	RTL	Shaped	On
10	Implicit	Arabic	Contextual LTR	Unshaped	On
11	Implicit	Arabic	Contextual RTL	Unshaped	On
12	Implicit	Arabic	RTL	Shaped	Off

Note: * String orientation is left-to-right (LTR) when the first alphabetic character is a Latin character, and right-to-left (RTL) when it is an Arabic or Hebrew character. Characters are unshaped, but LamAlef ligatures are kept and are not broken into constituents.

Related concepts:

- "Bidirectional support with DB2 Connect" on page 32

Related tasks:

- "Enabling bidirectional support" on page 27

Bidirectional support with DB2 Connect

When data is exchanged between DB2 Connect™ and a database on the server, it is usually the receiver that performs conversion on the incoming data. The same convention would normally apply to bidirectional layout transformations, and is in addition to the usual code page conversion. DB2 Connect has the optional ability to perform bidirectional layout transformation on data it is about to send to the server database, in addition to data received from the server database.

In order for DB2 Connect to perform bidirectional layout transformation on outgoing data for a server database, the bidirectional CCSID of the server database must be overridden. This is accomplished through the use of the BIDI parameter in the PARMS field of the DCS database directory entry for the server database.

Note: If you want DB2 Connect to perform layout transformation on the data it is about to send to the DB2 host or iSeries™ database, even though you do not have to override its CCSID, you must still add the BIDI parameter to the PARMS field of the DCS database directory. In this case, the CCSID that you should provide is the default DB2 host or iSeries database CCSID.

The BIDI parameter is to be specified as the ninth parameter in the PARMS field, along with the bidirectional CCSID with which you want to override the default server database bidirectional CCSID:

```
",,,,,,,,BIDI=xyz"
```

where *xyz* is the CCSID override.

Note: The registry variable DB2BIDI must be set to YES for the BIDI parameter to take effect.

The use of this feature is best described with an example.

Suppose you have a Hebrew DB2 client running CCSID 62213 (bidirectional string type 5), and you want to access a DB2 host or iSeries database running CCSID 00424 (bidirectional string type 4). However, you know that the data contained in the DB2 host or iSeries database is based on CCSID 08616 (bidirectional string type 6).

There are two problems here: The first is that the DB2 host or iSeries database does not know the difference in the bidirectional string types with CCSIDs 00424 and 08616. The second problem is that the DB2 host or iSeries database does not recognize the DB2 client CCSID (62213). It only supports CCSID 00862, which is based on the same code page as CCSID 62213.

You will need to ensure that data sent to the DB2 host or iSeries database is in bidirectional string type 6 format to begin with, and also let DB2 Connect know that it has to perform bidirectional transformation on data it receives from the DB2 host or iSeries database. You will need to use following catalog command for the DB2 host or iSeries database:

```
db2 catalog dcs database nydb1 as telaviv parms ",,,,,,,,BIDI=08616"
```

This command tells DB2 Connect to override the DB2 host or iSeries database CCSID of 00424 with 08616. This override includes the following processing:

1. DB2 Connect connects to the DB2 host or iSeries database using CCSID 00862.

2. DB2 Connect performs bidirectional layout transformation on the data it is about to *send* to the DB2 host or iSeries database. The transformation is from CCSID 62213 (bidirectional string type 5) to CCSID 62221 (bidirectional string type 6).
3. DB2 Connect performs bidirectional layout transformation on data it *receives* from the DB2 host or iSeries database. This transformation is from CCSID 08616 (bidirectional string type 6) to CCSID 62213 (bidirectional string type 5).

Note: In some cases, use of a bidirectional CCSID may cause the SQL query itself to be modified in such a way that it is not recognized by the DB2 server. Specifically, you should avoid using IMPLICIT CONTEXTUAL and IMPLICIT RIGHT-TO-LEFT CCSIDs when a different string type can be used. CONTEXTUAL CCSIDs can produce unpredictable results if the SQL query contains quoted strings. Avoid using quoted strings in SQL statements; use host variables whenever possible.

If a specific bidirectional CCSID is causing problems that cannot be rectified by following these recommendations, set DB2BIDI to NO.

Related concepts:

- "Handling BiDi data" in *DB2 Connect User's Guide*

Related reference:

- "Bidirectional-specific CCSIDs" on page 29

Collating sequences

The database manager compares character data using a *collating sequence*. This is an ordering for a set of characters that determines whether a particular character sorts higher, lower, or the same as another.

Note: Character string data defined with the FOR BIT DATA attribute, and BLOB data, is sorted using the binary sort sequence.

For example, a collating sequence can be used to indicate that lowercase and uppercase versions of a particular character are to be sorted equally.

The database manager allows databases to be created with custom collating sequences. The following sections help you determine and implement a particular collating sequence for a database.

The database manager allows databases to be created with custom collating sequences. For Unicode databases, the various collating sequences supported are described in the "Unicode implementation in the DB2 database" topic. The following sections help you determine and implement a particular collating sequence for a database.

Each single-byte character in a database is represented internally as a unique number between 0 and 255 (in hexadecimal notation, between X'00' and X'FF'). This number is referred to as the *code point* of the character; the assignment of numbers to characters in a set is collectively called a *code page*. A collating sequence is a mapping between the code point and the desired position of each character in a sorted sequence. The numeric value of the position is called the *weight* of the character in the collating sequence. In the simplest collating sequence, the weights are identical to the code points. This is called the *identity sequence*.

For example, suppose the characters B and b have the code points X'42' and X'62', respectively. If (according to the collating sequence table) they both have a sort weight of X'42' (B), they collate the same. If the sort weight for B is X'9E', and the sort weight for b is X'9D', b will be sorted before B. The collation sequence table specifies the weight of each character. The table is different from a code page, which specifies the code point of each character.

Consider the following example. The ASCII characters A through Z are represented by X'41' through X'5A'. To describe a collating sequence in which these characters are sorted consecutively (no intervening characters), you can write: X'41', X'42', ... X'59', X'5A'.

The hexadecimal value of a multi-byte character is also used as the weight. For example, suppose the code points for the double-byte characters A and B are X'8260' and X'8261' respectively, then the collation weights for X'82', X'60', and X'61' are used to sort these two characters according to their code points.

The weights in a collating sequence need not be unique. For example, you could give uppercase letters and their lowercase equivalents the same weight.

Specifying a collating sequence can be simplified if the collating sequence provides weights for all 256 code points. The weight of each character can be determined using the code point of the character.

In all cases, the DB2 database uses the collation table that was specified at database creation time. If you want the multi-byte characters to be sorted the way that they appear in their code point table, you must specify IDENTITY as the collation sequence when you create the database.

Note: For Unicode databases, the various collating sequences supported are described in the "Unicode implementation in the DB2 database" topic.

Once a collating sequence is defined, all future character comparisons for that database will be performed with that collating sequence. Except for character data defined as FOR BIT DATA or BLOB data, the collating sequence will be used for all SQL comparisons and ORDER BY clauses, and also in setting up indexes and statistics.

Potential problems can occur in the following cases:

- An application merges sorted data from a database with application data that was sorted using a different collating sequence.
- An application merges sorted data from one database with sorted data from another, but the databases have different collating sequences.
- An application makes assumptions about sorted data that are not true for the relevant collating sequence. For example, numbers collating lower than alphabetics might or might not be true for a particular collating sequence.

A final point to remember is that the results of any sort based on a direct comparison of character code points will only match query results that are ordered using an identity collating sequence.

Related concepts:

- "Character comparisons based on collating sequences" in *Developing SQL and External Routines*
- "Character conversion" in *SQL Reference, Volume 1*

- “Unicode implementation in DB2 Database for Linux, UNIX, and Windows” on page 38

Collating Thai characters

Thai contains special vowels (″leading vowels″), tonal marks and other special characters that are not sorted sequentially.

Restrictions:

You must either create your database with a Thai locale and code set, or create a Unicode database.

Procedure:

When you create a database using Thai and corresponding code set, use the COLLATE USING NLSCHAR clause on the CREATE DATABASE command. When you create a Unicode database, use the COLLATE USING UCA400_LTH clause on the CREATE DATABASE command.

Related concepts:

- “Collating sequences” on page 33

Related reference:

- “CREATE DATABASE command” in *Command Reference*

Chapter 2. Unicode support

Unicode character encoding

The Unicode character encoding standard is a fixed-length, character encoding scheme that includes characters from almost all of the living languages of the world.

Information on Unicode can be found in the latest edition of The Unicode Standard book, and from The Unicode Consortium web site (www.unicode.org).

Unicode uses two encoding forms: 8-bit and 16-bit. The default encoding form is 16-bit, that is, each character is 16 bits (two bytes) wide, and is usually shown as U+hhhh, where hhhh is the hexadecimal code point of the character. While the resulting 65 000+ code elements are sufficient for encoding most of the characters of the major languages of the world, the Unicode standard also provides an extension mechanism that allows the encoding of as many as one million more characters. The extension mechanism uses a pair of high and low surrogate characters to encode one extended or supplementary character. The first (or high) surrogate character has a code value between U+D800 and U+DBFF, and the second (or low) surrogate character has a code value between U+DC00 and U+DFFF.

UCS-2

The International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) standard 10646 (ISO/IEC 10646) specifies the Universal Multiple-Octet Coded Character Set (UCS) that has a 16-bit (two-byte) version (UCS-2) and a 32-bit (four-byte) version (UCS-4). UCS-2 is identical to the Unicode 16-bit form without surrogates. UCS-2 can encode all the (16-bit) characters defined in the Unicode version 3.0 repertoire. Two UCS-2 characters — a high followed by a low surrogate — are required to encode each of the new supplementary characters introduced starting in Unicode version 3.1. These supplementary characters are defined outside the original 16-bit Basic Multilingual Plane (BMP or Plane 0).

UTF-8

Sixteen-bit Unicode characters pose a major problem for byte-oriented ASCII-based applications and file systems. For example, non-Unicode aware applications may misinterpret the leading 8 zero bits of the uppercase character 'A' (U+0041) as the single-byte ASCII NULL character.

UTF-8 (UCS Transformation Format 8) is an algorithmic transformation that transforms fixed-length Unicode characters into variable-length ASCII-safe byte strings. In UTF-8, ASCII and control characters are represented by their usual single-byte codes, and other characters become two or more bytes long. UTF-8 can encode both non-supplementary and supplementary characters.

UTF-16

ISO/IEC 10646 also defines an extension technique for encoding some UCS-4 characters using two UCS-2 characters. This extension, called UTF-16, is identical to the Unicode 16-bit encoding form with surrogates. In summary, the UTF-16

character repertoire consists of all the UCS-2 characters plus the additional one million characters accessible via the surrogate pairs.

When serializing 16-bit Unicode characters into bytes, some processors place the most significant byte in the initial position (known as big-endian order), while others place the least significant byte first (known as little-endian order). The default byte ordering for Unicode is big-endian.

The number of bytes for each UTF-16 character in UTF-8format can be determined from Table 62.

Table 62. UTF-8 Bit Distribution

Code Value (binary)	UTF-16 (binary)	1st byte (binary)	2nd byte (binary)	3rd byte (binary)	4th byte (binary)
00000000 0xxxxxxx	00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzzyyyy yyxxxxxx	zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
uuuuu zzzzyyyy yyxxxxxx	110110ww wwzzzzyy 110111yy yyxxxxxx	11110uuu (where uuuuu = wwww+1)	10uuzzzz	10yyyyyy	10xxxxxx

In each of the above, the series of u's, w's, x's, y's, and z's is the bit representation of the character. For example, U+0080 transforms into 11000010 10000000 in binary, and the surrogate character pair U+D800 U+DC00 becomes 11110000 10010000 10000000 10000000 in binary.

Related concepts:

- "Unicode handling of data types" on page 42
- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38
- "Unicode literals" on page 45

Related tasks:

- "Creating a Unicode database" on page 43

Unicode implementation in DB2 Database for Linux, UNIX, and Windows

DB2 Database for Linux, UNIX, and Windows supports UTF-8 and UCS-2.

When a Unicode database is created, CHAR, VARCHAR, LONG VARCHAR, and CLOB data are stored in UTF-8 form, and GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, and DBCLOB data are stored in UCS-2 big-endian form.

In versions of DB2 products prior to Version 7.2 FixPak 4, DB2 treats the two characters in a surrogate pair as two independent Unicode characters. Therefore transforming the pair from UTF-16/UCS-2 to UTF-8 results in two three-byte sequences. Starting in DB2 Universal Database™ Version 7.2 FixPak 4, DB2 recognizes surrogate pairs when transforming between UTF-16/UCS-2 and UTF-8, thus a pair of UTF-16 surrogates will become one UTF-8 four-byte sequence. In other usages, DB2 continues to treat a surrogate pair as two independent UCS-2 characters. You can safely store supplementary characters in DB2 Unicode databases, provided you know how to distinguish them from the non-supplementary characters.

DB2 treats each Unicode character, including those (non-spacing) characters such as the COMBINING ACUTE ACCENT character (U+0301), as an individual character. Therefore DB2 would not recognize that the character LATIN SMALL LETTER A WITH ACUTE (U+00E1) is canonically equivalent to the character LATIN SMALL LETTER A (U+0061) followed by the character COMBINING ACUTE ACCENT (U+0301).

The default collating sequence for a UCS-2 Unicode database is IDENTITY, which orders the characters by their code points. Therefore, by default, all Unicode characters are ordered and compared according to their code points. For non-supplementary Unicode characters, their binary collation orders when encoded in UTF-8 and UCS-2 are the same. But if you have any supplementary character that requires a pair of surrogate characters to encode, then in UTF-8 encoding the character will be collated towards the end, but in UCS-2 encoding the same character will be collated somewhere in the middle, and its two surrogate characters can be separated. The reason is the extended character, when encoded in UTF-8, has a four-byte binary code value of 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx, which is greater than the UTF-8 encoding of U+FFFF, namely X'EFBFBF'. But in UCS-2, the same supplementary character is encoded as a pair of UCS-2 high and low surrogate characters, and has the binary form of 1101 1000 xxxx xxxx 1101 1100 xxxx xxxx, which is less than the UCS-2 encoding of U+FFFF.

A Unicode database can also be created with the IDENTITY_16BIT collation option. The IDENTITY_16BIT collator implements the CESU-8 *Compatibility Encoding Scheme for UTF-16: 8-Bit* algorithm as specified in the Unicode Technical Report #26 available at the Unicode Technical Consortium web site (www.unicode.org). CESU-8 is binary identical to UTF-8 except for the Unicode supplementary characters, that is, those characters that are defined outside the 16-bit Basic Multilingual Plane (BMP or Plane 0). In UTF-8 encoding, a supplementary character is represented by one four-byte sequence, but the same character in CESU-8 requires two three-byte sequences. Using the IDENTITY_16BIT collation option will yield the same collation order for both character and graphic data.

DB2 UDB Version 8.2 supports three new collation sequence keywords for Unicode databases: UCA400_NO, UCA400_LSK, and UCA400_LTH. The UCA400_NO collators implements the UCA (Unicode Collation Algorithm) based on the Unicode Standard version 4.00 with normalization implicitly set to on. The UCA400_LSK and UCA400_LTH collator also implement the UCA version 4.00. UCA400_LSK will sort all Slovakian characters in the appropriate order, and UCA400_LTH will sort all Thai characters as per the Royal Thai Dictionary order. Details of the UCA can be found in the Unicode Technical Standard #10 available at the Unicode Consortium web site (www.unicode.org).

All culturally sensitive parameters, such as date or time format, decimal separator, and others, are based on the current territory of the client.

A Unicode database allows connection from every code page supported by DB2. The database manager automatically performs code page conversion for character and graphic strings between the client's code page and Unicode.

Every client is limited by the character repertoire, the input method, and the fonts supported by its environment, but the UCS-2 database itself accepts and stores all UCS-2 characters. Therefore, every client usually works with a subset of UCS-2 characters, but the database manager allows the entire repertoire of UCS-2 characters.

When characters are converted from a local code page to Unicode, there may be expansion in the number of bytes. Prior to Version 8, based on the semantics of SQL statements, character data may have been marked as being encoded in the client's code page, and the database server would have manipulated the entire statement in the client's code page. This manipulation could have resulted in potential expansion of the data. Starting in Version 8, once an SQL statement enters the database server, it operates only on the database server's code page. In this case there is no size change. However, specifying string units for some string functions might result in internal codepage conversions. If this occurs, the size of the data string might change.

AIX, UNIX, and Linux distributions and code pages

Newer versions of AIX, some UNIX platforms, and many Linux distributions use Unicode (UTF-8) as the default code page instead of traditional non-Unicode code pages. If the operating system is upgraded on a system and the upgrade includes this change in the default code page, then:

- Applications that used to run may fail because the default active code page is modified.
- Any new database created after the operating system upgrade is created using the UTF-8 Unicode code page unless a code page is explicitly specified when creating a new database. All existing datbases retain their original code page settings; that is, the setting established during database creation.

To determine the active code page the system is running on Linux, run:

```
locale
```

Not all of the information displayed from running this command is important or relevant, however the DB2 database manager uses the following items in the order presented to determine the active code page:

- LC_ALL
- LC_CTYPE
- LANG

To determine which code page a database is using, run:

```
db2 get db cfg for <database name>
```

and check the value for the "Database code page" parameter.

Code Page/CCSID Numbers

Within IBM, the UCS-2 code page has been registered as code page 1200, with a growing character set; that is, when new characters are added to a code page, the code page number does not change. Code page 1200 always refers to the current version of Unicode.

A specific version of the UCS standard, as defined by Unicode 2.0 and ISO/IEC 10646-1, has also been registered within IBM as CCSID 13488. This CCSID has been used internally by DB2 for storing graphic string data in IBM eucJP (Japan) and IBM eucTW (Taiwan) databases. CCSID 13488 and code page 1200 both refer to UCS-2, and are handled the same way, except for the value of their "double-byte" (DBCS) space:

CP/CCSID	Single-byte (SBCS) space	Double-byte (DBCS) space
1200	N/A	U+0020
13488	N/A	U+3000

Note: In a UCS-2 database, U+3000 has no special meaning.

Regarding the conversion tables, since code page 1200 is a superset of CCSID 13488, the same (superset) tables are used for both.

Within IBM, UTF-8 has been registered as CCSID 1208 with growing character set (sometimes also referred to as code page 1208). As new characters are added to the standard, this number (1208) will not change.

The MBCS code page number is 1208, which is the database code page number, and the code page of character string data within the database. The double-byte code page number for UCS-2 is 1200, which is the code page of graphic string data within the database.

Thai and Unicode collation algorithm differences

The collation algorithm used in a Thai Industrial Standard (TIS) TIS620-1 (code page 874) Thai database with the NLSCHAR collation option is similar, but not identical to, the collation algorithm used in a Unicode database with the UCA400_LTH collation option. The differences are as follows:

- When sorting TIS620-1 data, each character only has one weight, and that weight is used to compare with another character's weight during collation. When sorting Unicode data, each character has several weights, and all the weights of that character can be used during collation.
- When sorting TIS620-1 data, the space character X'20', hyphen character X'2D', and full stop character X'2E' all have smaller weights than all the Thai characters. When sorting Unicode data, however, those three characters are considered as punctuation marks; and are used for comparison only when all other characters in the two strings being compared are equal.
- The Paiyannoi character X'CF' and the Maiyamok character X'E6' in a TIS620-1 database are treated as punctuation marks when they follow other Thai characters, and as normal characters, with their own weights, when they appear at the beginning of a string. The same two characters in a Unicode database (U+0E2F and U+0E46 respectively) are always treated as punctuation marks, and will be used for comparison when all other characters in the two strings being compared are equal.

More information on Thai characters can be found in chapter 10.1 Thai of the Unicode Standard book, version 4.0, ISBN 0-321-18578-1.

Related concepts:

- "Unicode character encoding" on page 37
- "Unicode handling of data types" on page 42

- "Unicode literals" on page 45

Related tasks:

- "Creating a Unicode database" on page 43

Related reference:

- "Character strings" in *SQL Reference, Volume 1*

Unicode handling of data types

All data types supported by DB2 Database for Linux, UNIX, and Windows are also supported in a UCS-2 database. In particular, graphic string data is supported for a UCS-2 database, and is stored in UCS-2/Unicode. Every client, including SBCS clients, can work with graphic string data types in UCS-2/Unicode when connected to a UCS-2 database.

A UCS-2 database is like any MBCS database where character string data is measured in number of bytes. When working with character string data in UTF-8, one should not assume that each character is one byte. In multibyte UTF-8 encoding, each ASCII character is one byte, but non-ASCII characters take two to four bytes each. This should be taken into account when defining CHAR fields. Depending on the ratio of ASCII to non-ASCII characters, a CHAR field of size *n* bytes can contain anywhere from *n*/4 to *n* characters.

Using character string UTF-8 encoding versus the graphic string UCS-2 data type also has an impact on the total storage requirements. In a situation where the majority of characters are ASCII, with some non-ASCII characters in between, storing UTF-8 data may be a better alternative, because the storage requirements are closer to one byte per character. On the other hand, in situations where the majority of characters are non-ASCII characters that expand to three- or four-byte UTF-8 sequences (for example ideographic characters), the UCS-2 graphic-string format may be a better alternative, because every three-byte UTF-8 sequence becomes a 16-bit UCS-2 character, while each four-byte UTF-8 sequence becomes two 16-bit UCS-2 characters.

In MBCS environments, SQL functions that operate on character strings, such as SUBSTR, POSSTR, MAX, MIN, and the like, operate on the number of "bytes" rather than number of "characters". The behavior is the same in a UCS-2 database, but you should take extra care when specifying offsets and lengths for a UCS-2 database, because these values are always defined in the context of the database code page. That is, in the case of a UCS-2 database, these offsets should be defined in UTF-8. Since some single-byte characters require more than one byte in UTF-8, SUBSTR indexes that are valid for a single-byte database may not be valid for a UCS-2 database. If you specify incorrect indexes, SQLCODE -191 (SQLSTATE 22504) is returned.

Note: Not all SQL functions that operate on character strings are limited to processing "bytes". The CHARACTER_LENGTH, LENGTH, LOCATE, POSITION, and SUBSTRING functions include a parameter that allows you to specify a predefined set of string units. This means that the functions can process strings using the specified units instead of bytes or double bytes.

SQL CHAR data types are supported (in the C language) by the `char` data type in user programs. SQL GRAPHIC data types are supported by `sqldbchar` in user programs. Note that, for a UCS-2 database, `sqldbchar` data is always in big-endian

(high byte first) format. When an application program is connected to a UCS-2 database, character string data is converted between the application code page and UTF-8, and graphic string data is converted between the application graphic code page and UCS-2 by DB2.

When retrieving data from a Unicode database to an application that does not use an SBCS, EUC, or Unicode code page, the defined substitution character is returned for each blank padded to a graphic column. DB2 pads fixed-length Unicode graphic columns with ASCII blanks (U+0200), a character that has no equivalent in pure DBCS code pages. As a result, each ASCII blank used in the padding of the graphic column is converted to the substitution character on retrieval. Similarly, in a DATE, TIME or TIMESTAMP string, any SBCS character that does not have a pure DBCS equivalent is also converted to the substitution character when retrieved from a Unicode database to an application that does not use an SBCS, EUC, or Unicode code page.

Note: Prior to Version 8, graphic string data was always assumed to be in UCS-2. To provide backward compatibility to applications that depend on the previous behavior of DB2, the registry variable DB2GRAPHICUNICODESERVER has been introduced. Its default value is OFF. Changing the value of this variable to ON will cause DB2 to use its earlier behavior and assume that graphic string data is always in UCS-2. Additionally, the DB2 server will check the version of DB2 running on the client, and will simulate DB2 Universal Database Version 7 behavior if the client is running DB2 UDB Version 7.

Related concepts:

- "Unicode character encoding" on page 37
- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38

Creating a Unicode database

By default, databases are created in the code page of the application creating them. Therefore, if you create your database from a Unicode (UTF-8) client, your database will be created as a Unicode database. Alternatively, you can explicitly specify "UTF-8" as the CODESET name, and use any valid TERRITORY code supported by DB2 Database for Linux, UNIX, and Windows.

In a future release of the DB2 database manager, the default code set will be changed to UTF-8 when creating a database, regardless of the application code page.

Procedure:

To create a Unicode database with the territory code for the United States of America:

```
DB2 CREATE DATABASE dbname USING CODESET UTF-8 TERRITORY US
```

To create a Unicode database using the **sqlecrea** API, you should set the values in *sqledbterritoryinfo* accordingly. For example, set SQLDBCODESET to `UTF-8`, and SQLDBLOCALE to any valid territory code (for example, `US`).

Related concepts:

- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38

Related tasks:

- "Converting non-Unicode databases to Unicode" on page 44

Related reference:

- "sqlecrea API - Create database" in *Administrative API Reference*
- "CREATE DATABASE command" in *Command Reference*
- "Supported territory codes and code pages" on page 1

Converting non-Unicode databases to Unicode

There are some cases where you might need to convert an existing non-Unicode database to a Unicode database. For example, because XML columns are only supported in Unicode databases, if you want to add an XML column to an existing non-Unicode database, you will need to convert the database to a Unicode database before you can add the XML column.

Prerequisites:

You must have enough free disk space to export the data from the non-Unicode database. Also, if you are not reusing the existing table spaces, you will need enough free disk space to create new table spaces for the data.

Restrictions:

XML data can only be stored in single-partition databases defined with the UTF-8 code set.

Procedure:

The following steps illustrate how to convert an existing non-Unicode database to a Unicode database:

1. Export your data using the **db2move** command:

   ```
   cd <export-dir>
   db2move sample export
   ```

   where `<export-dir>` is the directory to which you want to export your data and SAMPLE is the existing database name.
2. Generate a DDL script for your existing database using the **db2look** command:

   ```
   db2look -d sample -e -o unidb.ddl -l -x -f
   ```

   where SAMPLE is the existing database name and `unidb.ddl` is the file name for the generated DDL script. The `-l` option generates DDL for user defined table spaces, database partition groups and buffer pools, the `-x` option generates authorization DDL, and the `-f` option generates an update command for database configuration parameters.
3. Create the Unicode database:

   ```
   CREATE DATABASE UNIDB USING CODESET UTF-8 TERRITORY US
   ```

   where UNIDB is the name of the Unicode database.

4. Edit the `unidb.ddl` script and change all occurrences of the database name to the new Unicode database name:

   ```
   CONNECT TO UNIDB
   ```

   To keep the existing database, you must also change the file name specification for table spaces in the `unidb.ddl` file. Otherwise, you can drop the existing database and use the same table space files:

   ```
   DROP DATABASE SAMPLE
   ```

5. Recreate your database structure by running the DDL script that you edited:

   ```
   db2 -tvf unidb.ddl
   ```

6. Import your data into the new Unicode database using the **db2move** command:

   ```
   cd <export-dir>
   db2move unidb import
   ```

   where `<export-dir>` is the directory where you exported your data and UNIDB is the Unicode database name.

Related concepts:

- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38
- "Native XML data store overview" in *XML Guide*
- "XML data type" in *XML Guide*

Related tasks:

- "Creating a Unicode database" on page 43

Related reference:

- "db2look - DB2 statistics and DDL extraction tool command" in *Command Reference*
- "db2move - Database movement tool command" in *Command Reference*
- "DROP DATABASE command" in *Command Reference*
- "CONNECT (Type 1) statement" in *SQL Reference, Volume 2*
- "CONNECT (Type 2) statement" in *SQL Reference, Volume 2*

Unicode literals

Unicode literals can be specified in two ways:

- As a graphic string constant, using the G'...' or N'....' format. Any literal specified in this way will be converted by the database manager from the application code page to 16-bit Unicode.
- As a Unicode hexadecimal string, using the UX'....' or GX'....' format. The constant specified between the quotation marks after UX or GX must be a multiple of four hexadecimal digits in big-endian order. Each four-digit group represents one 16-bit Unicode code point. Note that surrogate characters always appear in pairs, therefore you need two four-digit groups to represent the high and low surrogate characters.

When using the command line processor (CLP), the first method is easier if the UCS-2 character exists in the local application code page (for example, when entering any code page 850 character from a terminal that is using code page 850). The second method should be used for characters that are outside of the

application code page repertoire (for example, when specifying Japanese characters from a terminal that is using code page 850).

Related concepts:

- "Unicode character encoding" on page 37
- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38

Related reference:

- "Constants" in *SQL Reference, Volume 1*

String comparisons in a Unicode database

Pattern matching is one area where the behavior of existing MBCS databases is slightly different from the behavior of a UCS-2 database.

For MBCS databases in DB2 Database for Linux, UNIX, and Windows, the current behavior is as follows: If the match-expression contains MBCS data, the pattern can include both SBCS and non-SBCS characters. The special characters in the pattern are interpreted as follows:

- An SBCS halfwidth underscore refers to one SBCS character.
- A non-SBCS fullwidth underscore refers to one non-SBCS character.
- A percent (either SBCS halfwidth or non-SBCS fullwidth) refers to zero or more SBCS or non-SBCS characters.

In a Unicode database, there is really no distinction between ″single-byte″ and ″non-single-byte″ characters. Although the UTF-8 format is a ″mixed-byte″ encoding of Unicode characters, there is no real distinction between SBCS and non-SBCS characters in UTF-8. Every character is a Unicode character, regardless of the number of bytes in UTF-8 format. In a Unicode graphic column, every non-supplementary character, including the halfwidth underscore (U+005F) and halfwidth percent (U+0025), is two bytes in width. For Unicode databases, the special characters in the pattern are interpreted as follows:

- For character strings, a halfwidth underscore (X'5F') or a fullwidth underscore (X'EFBCBF') refers to one Unicode character. A halfwidth percent (X'25') or a fullwidth percent (X'EFBC85') refers to zero or more Unicode characters.
- For graphic strings, a halfwidth underscore (U+005F) or a fullwidth underscore (U+FF3F) refers to one Unicode character. A halfwidth percent (U+0025) or a fullwidth percent (U+FF05) refers to zero or more Unicode characters.

Note: You need two underscores to match a Unicode supplementary graphic character because such a character is represented by two UCS-2 characters in a GRAPHIC column. Only one underscore is needed to match a Unicode supplementary character in a CHAR column.

For the optional ″escape expression″, which specifies a character to be used to modify the special meaning of the underscore and percent sign characters, the expression can be specified by any one of:

- A constant
- A special register
- A host variable
- A scalar function whose operands are any of the above

- An expression concatenating any of the above

with the restrictions that:

- No element in the expression can be of type LONG VARCHAR, CLOB, LONG VARGRAPHIC, or DBCLOB. In addition, it cannot be a BLOB file reference variable.
- For CHAR columns, the result of the expression must be one character or a binary string containing exactly one (1) byte (SQLSTATE 22019). For GRAPHIC columns, the result of the expression must be one character (SQLSTATE 22019).

Related concepts:

- "Unicode character encoding" on page 37
- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38

Related reference:

- "Character strings" in *SQL Reference, Volume 1*
- "Graphic strings" in *SQL Reference, Volume 1*

Chapter 3. Code page conversions

Character-conversion guidelines

Data conversion might be required to map data between application and database code pages when your application and database do not use the same code page. Because mapping and data conversion require additional overhead application performance improves if the application and database use the same code page or the identity collating sequence.

Character conversion occurs in the following circumstances:

- When a client or application runs in a code page that is different from the code page of the database that it accesses.

  The conversion occurs on the database server machine that receives the data. If the database server receives the data, character conversion is from the application code page to the database code page. If the application machine receives the data, conversion is from the database code page to the application code page.
- When a client or application that imports or loads a file runs in a code page different from the file being imported or loaded.

Character conversion does not occur for the following objects:

- File names.
- Data targeted for or coming from a column for which the FOR BIT DATA attribute is assigned, or data that is used in an SQL operation whose result is FOR BIT or BLOB data.
- A DB2 product or platform for which no supported conversion function to or from EUC or UCS-2 is installed. Your application receives an SQLCODE -332 (SQLSTATE 57017) error in this case.

The conversion function and conversion tables or DBCS conversion APIs that the database manager uses when it converts multi-byte code pages depends on the operating system environment.

Note: Character string conversions between multi-byte code pages, such as DBCS with EUC, might increase or decrease length of a string. In addition, code points assigned to different characters in the PC DBCS, EUC, and UCS-2 code sets might produce different results when same characters are sorted.

Extended UNIX Code (EUC) Code Page Support

Host variables that use graphic data in C or C++ applications require special considerations that include special precompiler, application performance, and application design issues.

Many characters in both the Japanese and Traditional Chinese EUC code pages require special methods of managing database and client application support for graphic data, which require double byte characters. Graphic data from these EUC code pages is stored and manipulated using the UCS-2 code set.

Related concepts:

- "Guidelines for analyzing where a federated query is evaluated" in *Performance Guide*

Related reference:

- "Conversion table files for euro-enabled code pages" on page 51
- "Conversion tables for code pages 923 and 924" on page 56

Enabling and disabling euro symbol support

DB2 Database for Linux, UNIX, and Windows provides support for the euro currency symbol. The euro symbol has been added to numerous code pages.

Microsoft ANSI code pages have been modified to include the euro currency symbol in position X'80'. Code page 850 has been modified to replace the character DOTLESS I (found at position X'D5') with the euro currency symbol. DB2 internal code page conversion routines use these revised code page definitions as the default to provide euro symbol support.

However, if you want to use the non-euro definitions of the code page conversion tables, follow the procedure below after installation is complete.

Prerequisites:

For replacing existing external code page conversion table files, you may want to back up the current files before copying the non-euro versions over them.

The files are located in the directory `sqllib/conv/`. On UNIX, `sqllib/conv/` is linked to the install path of the DB2 database system.

Procedure:

To disable euro-symbol support:

1. Stop the DB2 instance.
2. Download the appropriate conversion table files, in binary:
   - For big-endian platforms from ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/conv/BigEndian/. This ftp server is anonymous, so if you are connecting via the command line, log in as user "anonymous" and use your e-mail address as your password. After logging in, change to the conversion tables directory: cd ps/products/db2/info/vr8/conv/BigEndian/
   - For little-endian platforms from ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/conv/LittleEndian/. This ftp server is anonymous, so if you are connecting via the command line, log in as user "anonymous" and use your e-mail address as your password. After logging in, change to the conversion tables directory: cd ps/products/db2/info/vr8/conv/LittleEndian
3. Copy the files to your `sqllib/conv/` directory.
4. Restart the DB2 instance.

Code pages 819 and 1047:

For code pages 819 (ISO 8859-1 Latin 1 ASCII) and 1047 (Latin 1 Open System EBCDIC), the euro replacement code pages, 923 (ISO 8859-15 Latin 9 ASCII) and 924 (Latin 9 Open System EBCDIC) respectively, contain not just the euro symbol

but also several new characters. DB2 Database for Linux, UNIX, and Windows continues to use the old (non-euro) definitions of these two code pages and conversion tables, namely 819 and 1047, by default. There are two ways to activate the new 923/924 code page and the associated conversion tables:

- Create a new database that uses the new code page. For example,
  ```
  DB2 CREATE DATABASE dbname USING CODESET ISO8859-15 TERRITORY US
  ```
- Copy the 923 or 924 conversion table files from the `sqllib/conv/alt/` directory to the `sqllib/conv/` directory and rename them to 819 or 1047, respectively.

Related concepts:

- "Character conversion" in *SQL Reference, Volume 1*

Related reference:

- "Conversion table files for euro-enabled code pages" on page 51
- "Conversion tables for code pages 923 and 924" on page 56

Conversion table files for euro-enabled code pages

The following tables list the conversion tables that have been enhanced to support the euro currency symbol. If you want to disable euro symbol support, download the conversion table file indicated in the column titled ″Conversion table file″.

Arabic:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
864, 17248	1046, 9238	08641046.cnv, 10460864.cnv, IBM00864.ucs
864, 17248	1256, 5352	08641256.cnv, 12560864.cnv, IBM00864.ucs
864, 17248	1200, 1208, 13488, 17584	IBM00864.ucs
1046, 9238	864, 17248	10460864.cnv, 08641046.cnv, IBM01046.ucs
1046, 9238	1089	10461089.cnv, 10891046.cnv, IBM01046.ucs
1046, 9238	1256, 5352	10461256.cnv, 12561046.cnv, IBM01046.ucs
1046, 9238	1200, 1208, 13488, 17584	IBM01046.ucs
1089	1046, 9238	10891046.cnv, 10461089.cnv
1256, 5352	864, 17248	12560864.cnv, 08641256.cnv, IBM01256.ucs
1256, 5352	1046, 9238	12561046.cnv, 10461256.cnv, IBM01256.ucs
1256, 5352	1200, 1208, 13488, 17584	IBM01256.ucs

Baltic:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
921, 901	1257	09211257.cnv, 12570921.cnv, IBM00921.ucs
921, 901	1200, 1208, 13488, 17584	IBM00921.ucs
1257, 5353	921, 901	12570921.cnv, 09211257.cnv, IBM01257.ucs
1257, 5353	922, 902	12570922.cnv, 09221257.cnv, IBM01257.ucs
1257, 5353	1200, 1208, 13488, 17584	IBM01257.ucs

Belarus:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1131, 849	1251, 5347	11311251.cnv, 12511131.cnv
1131, 849	1283	11311283.cnv

Cyrillic:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
855, 872	866, 808	08550866.cnv, 08660855.cnv
855, 872	1251, 5347	08551251.cnv, 12510855.cnv
866, 808	855, 872	08660855.cnv, 08550866.cnv
866, 808	1251, 5347	08661251.cnv, 12510866.cnv
1251, 5347	855, 872	12510855.cnv, 08551251.cnv, IBM01251.ucs
1251, 5347	866, 808	12510866.cnv, 08661251.cnv, IBM01251.ucs
1251, 5347	1124	12511124.cnv, 11241251.cnv, IBM01251.ucs
1251, 5347	1125, 848	12511125.cnv, 11251251.cnv, IBM01251.ucs
1251, 5347	1131, 849	12511131.cnv, 11311251.cnv, IBM01251.ucs
1251, 5347	1200, 1208, 13488, 17584	IBM01251.ucs

Estonia:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
922, 902	1257	09221257.cnv, 12570922.cnv, IBM00922.ucs
922, 902	1200, 1208, 13488, 17584	IBM00922.ucs
1122, 1157	1257, 5353	11221257.cnv

Greek:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
813, 4909	869, 9061	08130869.cnv, 08690813.cnv, IBM00813.ucs
813, 4909	1253, 5349	08131253.cnv, 12530813.cnv, IBM00813.ucs
813, 4909	1200, 1208, 13488, 17584	IBM00813.ucs
869, 9061	813, 4909	08690813.cnv, 08130869.cnv
869, 9061	1253, 5349	08691253.cnv, 12530869.cnv
1253, 5349	813, 4909	12530813.cnv, 08131253.cnv, IBM01253.ucs
1253, 5349	869, 9061	12530869.cnv, 08691253.cnv, IBM01253.ucs
1253, 5349	1200, 1208, 13488, 17584	IBM01253.ucs

Hebrew:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
856, 9048	862, 867	08560862.cnv, 08620856.cnv, IBM0856.ucs
856, 9048	916	08560916.cnv, 09160856.cnv, IBM0856.ucs
856, 9048	1255, 5351	08561255.cnv, 12550856.cnv, IBM0856.ucs
856, 9048	1200, 1208, 13488, 17584	IBM0856.ucs
862, 867	856, 9048	08620856.cnv, 08560862.cnv, IBM00862.ucs
862, 867	916	08620916.cnv, 09160862.cnv, IBM00862.ucs
862, 867	1255, 5351	08621255.cnv, 12550862.cnv, IBM00862.ucs
862, 867	1200, 1208, 13488, 17584	IBM00862.ucs
916	856, 9048	09160856.cnv, 08560916.cnv
916	862, 867	09160862.cnv, 08620916.cnv
1255, 5351	856, 9048	12550856.cnv, 08561255.cnv, IBM01255.ucs
1255, 5351	862, 867	12550862.cnv, 08621255.cnv, IBM01255.ucs
1255, 5351	1200, 1208, 13488, 17584	IBM01255.ucs

Latin-1:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
437	850, 858	04370850.cnv, 08500437.cnv

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
500, 1148	437	05000437.cnv, IBM00500.ucs
850, 858	437	08500437.cnv, 04370850.cnv
850, 858	860	08500860.cnv, 08600850.cnv
850, 858	1114, 5210	08501114.cnv, 11140850.cnv
850, 858	1275	08501275.cnv, 12750850.cnv
860	850, 858	08600850.cnv, 08500860.cnv
1275	850, 858	12750850.cnv, 08501275.cnv

Latin-2:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
852, 9044	1250, 5346	08521250.cnv, 12500852.cnv
1250, 5346	852, 9044	12500852.cnv, 08521250.cnv, IBM01250.ucs
1250, 5346	1200, 1208, 13488, 17584	IBM01250.ucs

Simplified Chinese:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
837, 935, 1388	1200, 1208, 13488, 17584	1388ucs2.cnv
1386	1200, 1208, 13488, 17584	1386ucs2.cnv, ucs21386.cnv

Traditional Chinese:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
937, 835, 1371	950, 1370	09370950.cnv, 0937ucs2.cnv
937, 835, 1371	1200, 1208, 13488, 17584	0937ucs2.cnv
1114, 5210	850, 858	11140850.cnv, 08501114.cnv

Thailand:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
874, 1161	1200, 1208, 13488, 17584	IBM00874.ucs

Turkish:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
857, 9049	1254, 5350	08571254.cnv, 12540857.cnv
1254, 5350	857, 9049	12540857.cnv, 08571254.cnv, IBM01254.ucs

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1254, 5350	1200, 1208, 13488, 17584	IBM01254.ucs

Ukraine:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1124	1251, 5347	11241251.cnv, 12511124.cnv
1125, 848	1251, 5347	11251251.cnv, 12511125.cnv

Unicode:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1200, 1208, 13488, 17584	813, 4909	IBM00813.ucs
1200, 1208, 13488, 17584	862, 867	IBM00862.ucs
1200, 1208, 13488, 17584	864, 17248	IBM00864.ucs
1200, 1208, 13488, 17584	874, 1161	IBM00874.ucs
1200, 1208, 13488, 17584	921, 901	IBM00921.ucs
1200, 1208, 13488, 17584	922, 902	IBM00922.ucs
1200, 1208, 13488, 17584	1046, 9238	IBM01046.ucs
1200, 1208, 13488, 17584	1250, 5346	IBM01250.ucs
1200, 1208, 13488, 17584	1251, 5347	IBM01251.ucs
1200, 1208, 13488, 17584	1253, 5349	IBM01253.ucs
1200, 1208, 13488, 17584	1254, 5350	IBM01254.ucs
1200, 1208, 13488, 17584	1255, 5351	IBM01255.ucs
1200, 1208, 13488, 17584	1256, 5352	IBM01256.ucs
1200, 1208, 13488, 17584	1386	ucs21386.cnv, 1386ucs2.cnv

Vietnamese:

Database server CCSIDs/CPGIDs	Database client CCSIDs/CPGIDs	Conversion table files
1258, 5354	1129, 1163	12581129.cnv

Related concepts:

- "Character conversion" in *SQL Reference, Volume 1*

Related tasks:

- "Enabling and disabling euro symbol support" on page 50

Conversion tables for code pages 923 and 924

The following is a list of all the code page conversion table files that are associated with code pages 923 and 924. Each file is of the form XXXXYYYY.cnv or ibmZZZZZ.ucs, where XXXXX is the source code page number and YYYY is the target code page number. The file ibmZZZZZ.ucs supports conversion between code page ZZZZZ and Unicode.

To activate a particular code page conversion table, copy the conversion table file from the `sqllib/conv/alt/` directory to the `sqllib/conv/` directory and rename that conversion table file as shown in the second column.

For example, to support the euro symbol when connecting a 8859-1/15 (Latin 1/9) client to a Windows 1252 database, you need to copy and rename the following code page conversion table files:

- sqllib/conv/alt/09231252.cnv to sqllib/conv/08191252.cnv
- sqllib/conv/alt/12520923.cnv to sqllib/conv/12520819.cnv
- sqllib/conv/alt/ibm00923.ucs to sqllib/conv/ibm00819.ucs

923 and 924 conversion table files in the sqlllib/conv/alt/ directory	**New name in the sqllib/conv/ directory**
04370923.cnv	04370819.cnv
08500923.cnv	08500819.cnv
08600923.cnv	08600819.cnv
08630923.cnv	08630819.cnv
09230437.cnv	08190437.cnv
09230850.cnv	08190850.cnv
09230860.cnv	08190860.cnv
09231043.cnv	08191043.cnv
09231051.cnv	08191051.cnv
09231114.cnv	08191114.cnv
09231252.cnv	08191252.cnv
09231275.cnv	08191275.cnv
09241252.cnv	10471252.cnv
10430923.cnv	10430819.cnv
10510923.cnv	10510819.cnv
11140923.cnv	11140819.cnv
12520923.cnv	12520819.cnv
12750923.cnv	12750819.cnv
ibm00923.ucs	ibm00819.ucs

Related concepts:

- "Character conversion" in *SQL Reference, Volume 1*

Related tasks:

- "Enabling and disabling euro symbol support" on page 50

Installing the previous tables for converting between code page 1394 and Unicode

The conversion tables for code page 1394 (also known as Shift JIS X0213) and Unicode have been enhanced. The conversion between Japanese Shift JIS X0213 (1394) and Unicode now conforms to the final ISO/IEC 10646-1:2000 Amendment-1 for JIS X0213 characters. The previous version of the conversion tables is available via FTP from `ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/conv/`.

Procedure:

To install the previous definitions for converting between Shift JIS X0213 and Unicode:

1. Stop the DB2 Database for Linux, UNIX, and Windows instance.
2. Point your Web browser to `ftp://ftp.software.ibm.com/ps/products/db2/info/vr8/conv/` or use FTP to connect to the `ftp.software.ibm.com` site. This FTP server is anonymous.
3. If you are connecting via the command line, log in by entering `anonymous` as your user ID and your e-mail address as your password.
4. After logging in, change to the conversion tables directory:
   ```
   cd ps/products/db2/info/vr8/conv
   ```
5. Copy the two files, 1394ucs4.cnv and ucs41394.cnv, in binary form to your `sqllib/conv/` directory.
6. Restart the DB2 instance.

Related concepts:

- "Unicode implementation in DB2 Database for Linux, UNIX, and Windows" on page 38

Related reference:

- "Supported territory codes and code pages" on page 1

Alternative Unicode conversion table for the coded character set identifier (CCSID) 943

There are several IBM coded character set identifiers (CCSIDs) for Japanese code pages. CCSID 943 is registered as the Microsoft Japanese Windows Shift-JIS code page. You might encounter the following two problems when converting characters between CCSID 943 and Unicode. The problems are the result of differences between the IBM code page conversion tables and the Microsoft code page conversion tables.

Problem 1::

For historical reasons, over 300 characters in the CCSID 943 code page are represented by two or three code points each. The use of input method editors (IMEs) and code page conversion tables cause only one of these equivalent code points to be entered. For example, the lower case character for Roman numeral one ("i") has two equivalent code points: X'EEEF' and X'FA40'. Microsoft Windows IMEs always generate X'FA40' when "i" is entered. In general, IBM and Microsoft use the same primary code point to represent the character, except for the following 13 characters:

Table 63. CCSID 943 Shift-JIS code point conversion

Character name (Unicode code point)	IBM primary Shift-JIS code point	Microsoft Shift-JIS primary code point
Roman numeral one (U+2160)	X'FA4A'	X'8754'
Roman numeral two (U+2161)	X'FA4B'	X'8755'
Roman numeral three (U+2162)	X'FA4C'	X'8756'
Roman numeral four (U+2163)	X'FA4D'	X'8757'
Roman numeral five (U+2164)	X'FA4E'	X'8758'
Roman numeral six (U+2165)	X'FA4F'	X'8759'
Roman numeral seven (U+2166)	X'FA50'	X'875A'
Roman numeral eight (U+2167)	X'FA51'	X'875B'
Roman numeral nine (U+2168)	X'FA52'	X'875C'
Roman numeral ten (U+2169)	X'FA53'	X'875D'
Parenthesized ideograph stock (U+3231)	X'FA58'	X'878A'
Numero sign (U+2116)	X'FA59'	X'8782'
Telephone sign (U+2121)	X'FA5A'	X'8784'

IBM products such as DB2 database manager primarily use IBM code points, for example X'FA4A', to present the upper case Roman numeral "I", but Microsoft products use X'8754' to represent the same character. A Microsoft ODBC application can insert the "I" character as X'8754' into a DB2 database of CCSID 943, and the DB2 Control Center can insert the same character as X'FA4A' into the same CCSID 943 database. However, Microsoft ODBC applications can find only those rows that have "I" encoded as X'8754', and the DB2 Control Center can locate only those rows that have encoded "I" as X'FA4A'. To enable the DB2 Control Center to select "I" as X'8754', you need to replace the default IBM conversion table from Unicode to CCSID 943 with the alternate Microsoft conversion table provided by the DB2 database manager.

Problem 2::

The following list of characters, when converted from CCSID 943 to Unicode, will result in different code points depending on whether the IBM conversion table or the Microsoft conversion table is used. For these characters, the IBM conversion table conforms to the character names as specified in the Japanese Industry Standard JISX0208, JISX0212, and JISX0221.

Table 64. CCSID 943 to Unicode code point conversion

Shift-JIS code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'815C' (EM Dash)	U+2014 (EM Dash)	U+2015 (Horizontal Bar)

Table 64. CCSID 943 to Unicode code point conversion (continued)

Shift-JIS code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'8160' (Wave Dash)	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
X'8161' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel To)
X'817C' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
X'FA55' (Broken bar)	U+00A6 (Broken bar)	U+FFE4 (Fullwidth broken bar)

For example, the character EM dash with the CCSID 943 code point of X'815C' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid these potential problems, you need to replace the default IBM conversion table from CCSID 943 to Unicode with the alternate Microsoft conversion table provided by the DB2 database manager.

The use of the alternate Microsoft conversion tables between CCSID 943 and Unicode should be restricted to closed environments, where the DB2 clients and the DB2 databases that are running CCSID 943 and are all using the same alternate Microsoft conversion tables. If you have a DB2 client using the default IBM conversion tables and another client using the alternate Microsoft conversion tables, and both clients are inserting data to the same DB2 database of CCSID 943, the same character may be stored as different code points in the database.

Related concepts:

- "Unicode character encoding" on page 37

Related tasks:

- "Replacing the Unicode conversion tables for coded character set identifier (CCSID) 943 with Microsoft conversion tables" on page 59

Replacing the Unicode conversion tables for coded character set identifier (CCSID) 943 with Microsoft conversion tables

When you convert between coded character set identifier (CCSID) 943 and Unicode, the DB2 Database for Linux, UNIX, and Windows database manager default code page conversion tables are used. If you want to use a different version of the conversion tables, such as the Microsoft version, you must manually override the default conversion tables.

Prerequisites:

If the code page conversion table file you want to override already exists in the `conv` subdirectory of the `sqllib` directory, you should back up that file in case you want to revert to the default table.

Restrictions:

For conversion table replacement to be effective, the conversion table on the database server and all of its clients must be changed.

Procedure:

To replace the DB2 default conversion tables for converting between CCSID 943 and Unicode, follow these steps:

1. When replacing conversion tables on the client, stop all the applications that are using the database. If you have any CLP sessions running, issue the TERMINATE command for each session. When replacing conversion tables on the database server, stop all instances on all nodes by issuing the **db2stop** command.
2. Copy `sqllib/conv/ms/0943ucs2.cnv` to `sqllib/conv/0943ucs2.cnv`.
3. Copy `sqllib/conv/ms/ucs20943.cnv` to `sqllib/conv/ucs20943.cnv`.
4. Restart all the applications.

Related concepts:

- "Unicode character encoding" on page 37
- "Alternative Unicode conversion table for the coded character set identifier (CCSID) 943" on page 57

Alternative Unicode conversion table for the coded character set identifier (CCSID) 954

There are several IBM coded character set identifiers (CCSIDs) for Japanese code pages. CCSID 954 is registered as the Japanese EUC code page. CCSID 954 is a common encoding for Japanese UNIX and Linux platforms. When using Microsoft ODBC applications to connect to a DB2 database using CCSID 954, you might encounter potential problems when converting data in CCSID 954 to Unicode. The problems are the result of differences between IBM's code page conversion table and Microsoft's code page conversion table.

The following list of characters, when converted from CCSID 954 to Unicode, will result in different code points depending on which conversion table (IBM or Microsoft) is used. For these characters, the IBM conversion table conforms to the character names as specified in the Japanese Industry Standard (JIS) JISX0208, JISX0212, and JISX0221.

Table 65. CCSID 954 to Unicode code point conversion

EUC-JP code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'A1BD' (EM Dash)	U+2014 (EM Dash)	U+2015 (Horizontal Bar)
X'A1C1' (Wave Dash)	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
X'A1C2' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel To)
X'A1DD' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
X'8FA2C3' (Broken bar)	U+00A6 (Broken bar)	U+FFE4 (Fullwidth broken bar)

For example, the character EM dash with the CCSID 954 code point of X'A1BD' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. This can create potential problems for Microsoft ODBC applications because they would

treat U+2014 as an invalid code point. To avoid these potential problems, you need to replace the default IBM conversion table from CCSID 954 to Unicode with the alternate Microsoft conversion table provided by the DB2 database manager.

Related concepts:

- "Replacing the Unicode conversion table for coded character set identifier (CCSID) 954 with the Microsoft conversion table" on page 61
- "Unicode character encoding" on page 37

Replacing the Unicode conversion table for coded character set identifier (CCSID) 954 with the Microsoft conversion table

When you convert from coded character set identifier (CCSID) 954 to Unicode, the DB2 database manager default code page conversion table is used. If you want to use a different version of the conversion table such as the Microsoft version, you must manually override the default conversion table.

Prerequisites::

If the code page conversion table file you want to override already exists in the *conv* subdirectory of the *sqllib* directory, you should back up that file in case you want to revert to the default table.

Restrictions::

For conversion table replacement to be effective, every DB2 client that connects to the same database must have its conversion table changed. If your client is Japanese Windows whose ANSI code page is Shift-JIS (CCSID 943), you will also need to change the default conversion tables between CCSID 943 and Unicode to the Microsoft version. Otherwise, the different clients might store the same character using different code points.

Procedure::

To replace the DB2 default conversion table for converting from CCSID 954 to Unicode, follow these steps:

1. When replacing conversion tables on the client, stop all the applications that are using the database. If you have any CLP sessions running, issue the **TERMINATE** command for each session. When replacing conversion tables on the database server, stop all instances on all nodes by issuing the **db2stop** command.
2. Copy *sqllib/conv/ms/0954ucs2.cnv* to *sqllib/conv/0954ucs2.cnv*.
3. Restart all the applications.

To replace the DB2 default conversion tables for converting between CCSID 943 and Unicode, follow these steps:

1. When replacing conversion tables on the client, stop all the applications that are using the database. If you have any CLP sessions running, issue the **TERMINATE** command for each session. When replacing conversion tables on the database server, stop all instances on all nodes by issuing the **db2stop** command.
2. Copy *sqllib/conv/ms/0943ucs2.cnv* to *sqllib/conv/0943ucs2.cnv*.
3. Copy *sqllib/conv/ms/ucs20943.cnv* to *sqllib/conv/ucs20943.cnv*.

4. Restart all the applications.

Related concepts:

- "Alternative Unicode conversion table for the coded character set identifier (CCSID) 954" on page 60
- "Unicode character encoding" on page 37

Alternative Unicode conversion table for the coded character set identifier (CCSID) 5026

There are several IBM coded character set identifiers (CCSIDs) for Japanese code pages. CCSID 5026 is registered as a Japanese EBCDIC code page. When using Microsoft ODBC applications to connect to a DB2 host database of CCSID 5026, you might encounter potential problems when converting data in CCSID 5026 to Unicode. The problems are the result of differences between IBM's code page conversion table and Microsoft's code page conversion table. The following list of characters, when converted from CCSID 5026 to Unicode, will result in different code points depending on which conversion table (IBM or Microsoft) is used. For these characters, the IBM conversion table conforms to the character names as specified in the Japanese Industry Standard (JIS) JISX0208, JISX0212, and JISX0221.

Table 66. CCSID 5026 to Unicode code point conversion

EBCDIC code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'444A' (EM Dash)	U+2014 (EM Dash)	U+2015 (Horizontal Bar)
X'43A1' (Wave Dash)	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
X'447C' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel To)
X'4260' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
X'426A' (Broken bar)	U+00A6 (Broken bar)	U+FFE4 (Fullwidth broken bar)

For example, the character EM dash with the CCSID 5026 code point of X'444A' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid these potential problems, you need to replace the default IBM conversion table from CCSID 5026 to Unicode with the alternate Microsoft conversion table provided by the DB2 database manager.

Related concepts:

- "Unicode character encoding" on page 37

Related tasks:

- "Replacing the Unicode conversion table for coded character set identifier (CCSID) 5026 with the Microsoft conversion table" on page 63

Replacing the Unicode conversion table for coded character set identifier (CCSID) 5026 with the Microsoft conversion table

When you convert from coded character set identifier (CCSID) 5026 to Unicode, the DB2 database manager default code page conversion table is used. If you want to use a different version of the conversion table such as the Microsoft version, you must manually override the default conversion table.

Prerequisites::

If the code page conversion table file you want to override already exists in the *conv* subdirectory of the *sqllib* directory, you should back up that file in case you want to revert to the default table.

Restrictions::

For conversion table replacement to be effective, every DB2 client that connects to the same database must have its conversion table changed.

This Microsoft conversion table is only for data encoded in CCSID 5026 or 930, and cannot be used for data encoded in CCSID 1390. Since the DB2 database manager uses the same conversion table for data encoded in CCSIDs 5026, 930, and 1390, this means that once the default IBM conversion table has been replaced with the Microsoft conversion table, you should not select any data that is encoded in CCSID 1390.

Activating this alternate Microsoft conversion table does not change the code page conversion behavior of graphic data encoded in 5026 to Unicode. To enable graphic data encoded in 5026 conversion to Unicode using the alternate Microsoft conversion table, you must also copy the file *sqllib/conv/ms/0939ucs2.cnv* to *sqllib/conv/1399ucs2.cnv* in addition to the procedure outlined below. Once you complete these steps, the conversion of both character data and graphic data to Unicode from the following CCSIDs will also use the Microsoft conversion table: 5026, 930, 1390, 5035, 939, and 1399.

Procedure::

To replace the DB2 default conversion table for converting from CCSID 5026 to Unicode, follow these steps:

1. When replacing conversion tables on the client, stop all the applications that are using the database. If you have any CLP sessions running, issue the TERMINATE command for each session.
2. Copy *sqllib/conv/ms/0930ucs2.cnv* to *sqllib/conv/1390ucs2.cnv*.
3. Restart all of the applications.

Related concepts:

- "Alternative Unicode conversion table for the coded character set identifier (CCSID) 5026" on page 62
- "Unicode character encoding" on page 37

Alternative Unicode conversion table for the coded character set identifier (CCSID) 5035

There are several IBM coded character set identifiers (CCSIDs) for Japanese code pages. CCSID 5035 is registered as a Japanese EBCDIC code page. When using Microsoft ODBC applications to connect to a DB2 host database of CCSID 5035, you might encounter potential problems when converting data in CCSID 5035 to Unicode. The problems are the result of differences between IBM's code page conversion table and Microsoft's code page conversion table.

The following list of characters, when converted from CCSID 5035 to Unicode, will result in different code points depending on which conversion table (IBM or Microsoft) is used. For these characters, the IBM conversion table conforms to the character names as specified in the Japanese Industry Standard (JIS) JISX0208, JISX0212, and JISX0221.

Table 67. CCSID 5035 to Unicode code point conversion

EBCDIC code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'444A' (EM Dash)	U+2014 (EM Dash)	U+2015 (Horizontal Bar)
X'43A1' (Wave Dash)	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
X'447C' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel To)
X'4260' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
X'426A' (Broken bar)	U+00A6 (Broken bar)	U+FFE4 (Fullwidth broken bar)

For example, the character EM dash with the CCSID 5035 code point of X'444A' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid these potential problems, you need to replace the default IBM conversion table from CCSID 5035 to Unicode with the alternate Microsoft conversion table provided by the DB2 database manager.

Related concepts:

- "Unicode character encoding" on page 37
- "Replacing the Unicode conversion table for coded character set identifier (CCSID) 5035 with the Microsoft conversion table" on page 64

Replacing the Unicode conversion table for coded character set identifier (CCSID) 5035 with the Microsoft conversion table

When you convert from coded character set identifier (CCSID) 5035 to Unicode, the DB2 database manager default code page conversion table is used. If you want to use a different version of the conversion table such as the Microsoft version, you must manually override the default conversion table.

Prerequisites::

If the code page conversion table file you want to override already exists in the *conv* subdirectory of the *sqllib* directory, you should back up that file in case you want to revert to the default table.

Restrictions::

For conversion table replacement to be effective, every DB2 client that connects to the same database must have its conversion table changed.

This Microsoft conversion table is only for data encoded in CCSID 5039 or 939, and cannot be used for data encoded in CCSID 1399. Since the DB2 database manager uses the same conversion table for data encoded in CCSIDs 5035, 939, and 1399, this means that once the default IBM conversion table has been replaced with the Microsoft conversion table, you should not select any data that is encoded in CCSID 1399.

Once you have replaced the default IBM conversion table with the Microsoft conversion table, the conversion of graphic data to Unicode from the following CCSIDs will also use this Microsoft conversion table: 930, 1390, 939, and 1399.

Procedure::

To replace the DB2 default conversion table for converting from CCSID 5035 to Unicode, follow these steps:

1. When replacing conversion tables on the client, stop all the applications that are using the database. If you have any CLP sessions running, issue the **TERMINATE** command for each session.
2. Copy *sqllib/conv/ms/0939ucs2.cnv* to *sqllib/conv/1399ucs2.cnv*.
3. Restart all the applications.

Related concepts:

- "Alternative Unicode conversion table for the coded character set identifier (CCSID) 5035" on page 64
- "Unicode character encoding" on page 37

Alternative Unicode conversion table for the coded character set identifier (CCSID) 5039

There are several IBM coded character set identifiers (CCSIDs) for Japanese code pages. CCSID 943 is registered as the the Microsoft Japanese Windows Shift-JIS code page. However, the Shift-JIS code page on the HP-UX platform is registered as CCSID 5039. CCSID 5039 contains only Japanese Industry Standard (JIS) characters, and does not have any vendor-defined characters. When using Microsoft ODBC applications, you might encounter potential problems when converting data in CCSID 5039 to Unicode. The problems are the result of differences between IBM's code page conversion table and Microsoft's code page conversion table.

The following list of characters, when converted from CCSID 5039 to Unicode, will result in different code points depending on which conversion table (IBM or Microsoft) is used. For these characters, the IBM conversion table conforms to the character names as specified in the Japanese Industry Standard (JIS) JISX0208, and JISX0221.

Table 68. CCSID 5039 to Unicode code point conversion

Shift-JIS code point (character name)	IBM primary code point (Unicode name)	Microsoft primary code point (Unicode name)
X'815C' (EM Dash)	U+2014 (EM Dash)	U+2015 (Horizontal Bar)
X'8160' (Wave Dash)	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
X'8161' (Double vertical line)	U+2016 (Double vertical line)	U+2225 (Parallel To)
X'817C' (Minus sign)	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)

For example, the character EM dash with the CCSID 5039 code point of X'815C' is converted to the Unicode code point U+2014 when using the IBM conversion table, but is converted to U+2015 when using the Microsoft conversion table. This can create potential problems for Microsoft ODBC applications because they would treat U+2014 as an invalid code point. To avoid these potential problems, you need to replace the default IBM conversion table from CCSID 5039 to Unicode with the alternate Microsoft conversion table provided by the DB2 database manager.

Related concepts:

- "Replacing the Unicode conversion table for coded character set identifier (CCSID) 5039 with the Microsoft conversion table" on page 66
- "Unicode character encoding" on page 37

Replacing the Unicode conversion table for coded character set identifier (CCSID) 5039 with the Microsoft conversion table

When you convert from coded character set identifier (CCSID) 5039 to Unicode, the DB2 database manager default code page conversion table is used. If you want to use a different version of the conversion table such as the Microsoft version, you must manually override the conversion table.

Prerequisites::

If the code page conversion table file you want to override already exists in the *conv* subdirectory of the *sqllib* directory, you should back up that file in case you want to revert to the default table.

Restrictions::

For conversion table replacement to be effective, every DB2 client that connects to the same database must have its conversion table changed.

Procedure::

To replace the DB2 default conversion table for converting from CCSID 5039 to Unicode, follow these steps:

1. When replacing conversion tables on the client, stop all the applications that are using the database. If you have any CLP sessions running, issue the **TERMINATE** command for each session.
2. Copy *sqllib/conv/ms/5039ucs2.cnv* to *sqllib/conv/5039ucs2.cnv*.
3. Restart all the applications.

Related concepts:

- "Alternative Unicode conversion table for the coded character set identifier (CCSID) 5039" on page 65
- "Unicode character encoding" on page 37

Appendix A. DB2 Database technical information

Overview of the DB2 technical information

DB2 technical information is available through the following tools and methods:

- DB2 Information Center
  - Topics
  - Help for DB2 tools
  - Sample programs
  - Tutorials
- DB2 books
  - PDF files (downloadable)
  - PDF files (from the DB2 PDF CD)
  - printed books
- Command line help
  - Command help
  - Message help
- Sample programs

IBM periodically makes documentation updates available. If you access the online version on the DB2 Information Center at ibm.com®, you do not need to install documentation updates because this version is kept up-to-date by IBM. If you have installed the DB2 Information Center, it is recommended that you install the documentation updates. Documentation updates allow you to update the information that you installed from the *DB2 Information Center CD* or downloaded from Passport Advantage as new information becomes available.

Note: The DB2 Information Center topics are updated more frequently than either the PDF or the hard-copy books. To get the most current information, install the documentation updates as they become available, or refer to the DB2 Information Center at ibm.com.

You can access additional DB2 technical information such as technotes, white papers, and Redbooks™ online at ibm.com. Access the DB2 Information Management software library site at http://www.ibm.com/software/data/sw-library/.

Documentation feedback

We value your feedback on the DB2 documentation. If you have suggestions for how we can improve the DB2 documentation, send an e-mail to db2docs@ca.ibm.com. The DB2 documentation team reads all of your feedback, but cannot respond to you directly. Provide specific examples wherever possible so that we can better understand your concerns. If you are providing feedback on a specific topic or help file, include the topic title and URL.

Do not use this e-mail address to contact DB2 Customer Support. If you have a DB2 technical issue that the documentation does not resolve, contact your local IBM service center for assistance.

Related concepts:

- "Features of the DB2 Information Center" in *Online DB2 Information Center*
- "Sample files" in *Samples Topics*

Related tasks:

- "Invoking command help from the command line processor" in *Command Reference*
- "Invoking message help from the command line processor" in *Command Reference*
- "Updating the DB2 Information Center installed on your computer or intranet server" on page 75

Related reference:

- "DB2 technical library in hardcopy or PDF format" on page 70

DB2 technical library in hardcopy or PDF format

The following tables describe the DB2 library available from the IBM Publications Center at www.ibm.com/shop/publications/order. DB2 Version 9 manuals in PDF format can be downloaded from www.ibm.com/software/data/db2/udb/support/manualsv9.html.

Although the tables identify books available in print, the books might not be available in your country or region.

The information in these books is fundamental to all DB2 users; you will find this information useful whether you are a programmer, a database administrator, or someone who works with DB2 Connect or other DB2 products.

Table 69. DB2 technical information

Name	Form Number	Available in print
Administration Guide: Implementation	SC10-4221	Yes
Administration Guide: Planning	SC10-4223	Yes
Administrative API Reference	SC10-4231	Yes
Administrative SQL Routines and Views	SC10-4293	No
Call Level Interface Guide and Reference, Volume 1	SC10-4224	Yes
Call Level Interface Guide and Reference, Volume 2	SC10-4225	Yes
Command Reference	SC10-4226	No
Data Movement Utilities Guide and Reference	SC10-4227	Yes
Data Recovery and High Availability Guide and Reference	SC10-4228	Yes
Developing ADO.NET and OLE DB Applications	SC10-4230	Yes
Developing Embedded SQL Applications	SC10-4232	Yes

Table 69. DB2 technical information (continued)

Name	Form Number	Available in print
Developing SQL and External Routines	SC10-4373	No
Developing Java Applications	SC10-4233	Yes
Developing Perl and PHP Applications	SC10-4234	No
Getting Started with Database Application Development	SC10-4252	Yes
Getting started with DB2 installation and administration on Linux and Windows	GC10-4247	Yes
Message Reference Volume 1	SC10-4238	No
Message Reference Volume 2	SC10-4239	No
Migration Guide	GC10-4237	Yes
Net Search Extender Administration and User's Guide **Note:** HTML for this document is not installed from the HTML documentation CD.	SH12-6842	Yes
Performance Guide	SC10-4222	Yes
Query Patroller Administration and User's Guide	GC10-4241	Yes
Quick Beginnings for DB2 Clients	GC10-4242	No
Quick Beginnings for DB2 Servers	GC10-4246	Yes
Spatial Extender and Geodetic Data Management Feature User's Guide and Reference	SC18-9749	Yes
SQL Guide	SC10-4248	Yes
SQL Reference, Volume 1	SC10-4249	Yes
SQL Reference, Volume 2	SC10-4250	Yes
System Monitor Guide and Reference	SC10-4251	Yes
Troubleshooting Guide	GC10-4240	No
Visual Explain Tutorial	SC10-4319	No
What's New	SC10-4253	Yes
XML Extender Administration and Programming	SC18-9750	Yes
XML Guide	SC10-4254	Yes
XQuery Reference	SC18-9796	Yes

Table 70. DB2 Connect-specific technical information

Name	Form Number	Available in print
DB2 Connect User's Guide	SC10-4229	Yes

Table 70. DB2 Connect-specific technical information (continued)

Name	Form Number	Available in print
Quick Beginnings for DB2 Connect Personal Edition	GC10-4244	Yes
Quick Beginnings for DB2 Connect Servers	GC10-4243	Yes

Table 71. WebSphere® Information Integration technical information

Name	Form Number	Available in print
WebSphere Information Integration: Administration Guide for Federated Systems	SC19-1020	Yes
WebSphere Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018	Yes
WebSphere Information Integration: Configuration Guide for Federated Data Sources	SC19-1034	No
WebSphere Information Integration: SQL Replication Guide and Reference	SC19-1030	Yes

Note: The DB2 Release Notes provide additional information specific to your product's release and fix pack level. For more information, see the related links.

Related concepts:

- "Overview of the DB2 technical information" on page 69
- "About the Release Notes" in *Release Notes*

Related tasks:

- "Ordering printed DB2 books" on page 72

Ordering printed DB2 books

If you require printed DB2 books, you can buy them online in many but not all countries or regions. You can always order printed DB2 books from your local IBM representative. Keep in mind that some softcopy books on the *DB2 PDF Documentation* CD are unavailable in print. For example, neither volume of the *DB2 Message Reference* is available as a printed book.

Printed versions of many of the DB2 books available on the DB2 PDF Documentation CD can be ordered for a fee from IBM. Depending on where you are placing your order from, you may be able to order books online, from the IBM Publications Center. If online ordering is not available in your country or region, you can always order printed DB2 books from your local IBM representative. Note that not all books on the DB2 PDF Documentation CD are available in print.

Note: The most up-to-date and complete DB2 documentation is maintained in the DB2 Information Center at http://publib.boulder.ibm.com/infocenter/db2help/.

Procedure:

To order printed DB2 books:

- To find out whether you can order printed DB2 books online in your country or region, check the IBM Publications Center at http://www.ibm.com/shop/publications/order. You must select a country, region, or language to access publication ordering information and then follow the ordering instructions for your location.
- To order printed DB2 books from your local IBM representative:
  - Locate the contact information for your local representative from one of the following Web sites:
    - The IBM directory of world wide contacts at www.ibm.com/planetwide
    - The IBM Publications Web site at http://www.ibm.com/shop/publications/order. You will need to select your country, region, or language to the access appropriate publications home page for your location. From this page, follow the "About this site" link.
  - When you call, specify that you want to order a DB2 publication.
  - Provide your representative with the titles and form numbers of the books that you want to order.

Related concepts:

- "Overview of the DB2 technical information" on page 69

Related reference:

- "DB2 technical library in hardcopy or PDF format" on page 70

Displaying SQL state help from the command line processor

DB2 returns an SQLSTATE value for conditions that could be the result of an SQL statement. SQLSTATE help explains the meanings of SQL states and SQL state class codes.

Procedure:

To invoke SQL state help, open the command line processor and enter:

```
? sqlstate or ? class code
```

where *sqlstate* represents a valid five-digit SQL state and *class code* represents the first two digits of the SQL state.

For example, `? 08003` displays help for the 08003 SQL state, and `? 08` displays help for the 08 class code.

Related tasks:

- "Invoking command help from the command line processor" in *Command Reference*
- "Invoking message help from the command line processor" in *Command Reference*

Accessing different versions of the DB2 Information Center

For DB2 Version 9 topics, the DB2 Information Center URL is http://publib.boulder.ibm.com/infocenter/db2luw/v9/.

For DB2 Version 8 topics, go to the Version 8 Information Center URL at: http://publib.boulder.ibm.com/infocenter/db2luw/v8/.

Related tasks:

- "Setting up access to DB2 contextual help and documentation" in *Administration Guide: Implementation*

Displaying topics in your preferred language in the DB2 Information Center

The DB2 Information Center attempts to display topics in the language specified in your browser preferences. If a topic has not been translated into your preferred language, the DB2 Information Center displays the topic in English.

Procedure:

To display topics in your preferred language in the Internet Explorer browser:

1. In Internet Explorer, click the **Tools** —> **Internet Options** —> **Languages...** button. The Language Preferences window opens.
2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button.

     Note: Adding a language does not guarantee that the computer has the fonts required to display the topics in the preferred language.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

To display topics in your preferred language in a Firefox or Mozilla browser:

1. Select the **Tools** —> **Options** —> **Languages** button. The Languages panel is displayed in the Preferences window.
2. Ensure your preferred language is specified as the first entry in the list of languages.
   - To add a new language to the list, click the **Add...** button to select a language from the Add Languages window.
   - To move a language to the top of the list, select the language and click the **Move Up** button until the language is first in the list of languages.
3. Clear the browser cache and then refresh the page to display the DB2 Information Center in your preferred language.

On some browser and operating system combinations, you might have to also change the regional settings of your operating system to the locale and language of your choice.

Related concepts:

- "Overview of the DB2 technical information" on page 69

Updating the DB2 Information Center installed on your computer or intranet server

If you have a locally-installed DB2 Information Center, updated topics can be available for download. The 'Last updated' value found at the bottom of most topics indicates the current level for that topic.

To determine if there is an update available for the entire DB2 Information Center, look for the 'Last updated' value on the Information Center home page. Compare the value in your locally installed home page to the date of the most recent downloadable update at http://www.ibm.com/software/data/db2/udb/support/icupdate.html. You can then update your locally-installed Information Center if a more recent downloadable update is available.

Updating your locally-installed DB2 Information Center requires that you:

1. Stop the DB2 Information Center on your computer, and restart the Information Center in stand-alone mode. Running the Information Center in stand-alone mode prevents other users on your network from accessing the Information Center, and allows you to download and apply updates.
2. Use the Update feature to determine if update packages are available from IBM.

   Note: Updates are also available on CD. For details on how to configure your Information Center to install updates from CD, see the related links.

   If update packages are available, use the Update feature to download the packages. (The Update feature is only available in stand-alone mode.)
3. Stop the stand-alone Information Center, and restart the DB2 Information Center service on your computer.

Procedure:

To update the DB2 Information Center installed on your computer or intranet server:

1. Stop the DB2 Information Center service.
   - On Windows, click **Start → Control Panel → Administrative Tools → Services**. Then right-click on **DB2 Information Center** service and select **Stop**.
   - On Linux, enter the following command:

     ```
     /etc/init.d/db2icdv9 stop
     ```
2. Start the Information Center in stand-alone mode.
   - On Windows:
     a. Open a command window.
     b. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `C:\Program Files\IBM\DB2 Information Center\Version 9` directory.
     c. Run the `help_start.bat` file using the fully qualified path for the DB2 Information Center:

        ```
        <DB2 Information Center dir>\doc\bin\help_start.bat
        ```
   - On Linux:

a. Navigate to the path where the Information Center is installed. By default, the DB2 Information Center is installed in the `/opt/ibm/db2ic/V9` directory.
b. Run the `help_start` script using the fully qualified path for the DB2 Information Center:

```
<DB2 Information Center dir>/doc/bin/help_start
```

The systems default Web browser launches to display the stand-alone Information Center.

3. Click the Update button (). On the right hand panel of the Information Center, click **Find Updates**. A list of updates for existing documentation displays.
4. To initiate the download process, check the selections you want to download, then click **Install Updates**.
5. After the download and installation process has completed, click **Finish**.
6. Stop the stand-alone Information Center.
   - On Windows, run the `help_end.bat` file using the fully qualified path for the DB2 Information Center:

     ```
     <DB2 Information Center dir>\doc\bin\help_end.bat
     ```

     Note: The help_end batch file contains the commands required to safely terminate the processes that were started with the help_start batch file. Do not use `Ctrl-C` or any other method to terminate `help_start.bat`.
   - On Linux, run the `help_end` script using the fully qualified path for the DB2 Information Center:

     ```
     <DB2 Information Center dir>/doc/bin/help_end
     ```

     Note: The help_end script contains the commands required to safely terminate the processes that were started with the help_start script. Do not use any other method to terminate the `help_start` script.
7. Restart the DB2 Information Center service.
   - On Windows, click **Start → Control Panel → Administrative Tools → Services**. Then right-click on **DB2 Information Center** service and select **Start**.
   - On Linux, enter the following command:

     ```
     /etc/init.d/db2icdv9 start
     ```

The updated DB2 Information Center displays the new and updated topics.

Related concepts:

- "DB2 Information Center installation options" in *Quick Beginnings for DB2 Servers*

Related tasks:

- "Installing DB2 Information Center updates from CD" in *Online DB2 Information Center*
- "Installing the DB2 Information Center using the DB2 Setup wizard (Linux)" in *Quick Beginnings for DB2 Servers*
- "Installing the DB2 Information Center using the DB2 Setup wizard (Windows)" in *Quick Beginnings for DB2 Servers*

DB2 tutorials

The DB2 tutorials help you learn about various aspects of DB2 products. Lessons provide step-by-step instructions.

Before you begin:

You can view the XHTML version of the tutorial from the Information Center at http://publib.boulder.ibm.com/infocenter/db2help/.

Some lessons use sample data or code. See the tutorial for a description of any prerequisites for its specific tasks.

DB2 tutorials:

To view the tutorial, click on the title.

Native XML data store
Set up a DB2 database to store XML data and to perform basic operations with the native XML data store.

Visual Explain Tutorial
Analyze, optimize, and tune SQL statements for better performance using Visual Explain.

Related concepts:

- "Visual Explain overview" in *Administration Guide: Implementation*

DB2 troubleshooting information

A wide variety of troubleshooting and problem determination information is available to assist you in using DB2 products.

DB2 documentation
Troubleshooting information can be found in the DB2 Troubleshooting Guide or the Support and Troubleshooting section of the DB2 Information Center. There you will find information on how to isolate and identify problems using DB2 diagnostic tools and utilities, solutions to some of the most common problems, and other advice on how to solve problems you might encounter with your DB2 products.

DB2 Technical Support Web site
Refer to the DB2 Technical Support Web site if you are experiencing problems and want help finding possible causes and solutions. The Technical Support site has links to the latest DB2 publications, TechNotes, Authorized Program Analysis Reports (APARs or bug fixes), fix packs, and other resources. You can search through this knowledge base to find possible solutions to your problems.

Access the DB2 Technical Support Web site at http://www.ibm.com/software/data/db2/udb/support.html

Related concepts:

- "Introduction to problem determination" in *Troubleshooting Guide*
- "Overview of the DB2 technical information" on page 69

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, non commercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work 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.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Appendix B. Notices

IBM may not offer the products, services, or features discussed in this document in all 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 give 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/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country/region 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 that has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

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, 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 may contain 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.

COPYRIGHT LICENSE:

This information may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

Trademarks

Company, product, or service names identified in the documents of the DB2 Version 9 documentation library may be trademarks or service marks of International Business Machines Corporation or other companies. Information on the trademarks of IBM Corporation in the United States, other countries, or both is located at http://www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 documentation library:

Microsoft, Windows, Windows NT®, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel®, Itanium®, Pentium®, and Xeon® are trademarks of Intel 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.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Index

Contacting IBM

To contact IBM in your country or region, check the IBM Directory of Worldwide Contacts at http://www.ibm.com/planetwide

To learn more about DB2 products, go to http://www.ibm.com/software/data/db2/.

Printed in USA

SC10-4380-00