Unicode Collation Algorithm based collations

Version 9.5.0

The CREATE DATABASE command and the COLLATION_KEY_BIT scalar function support a new collation keyword, UCA500R1, which implements the UCA (Unicode Collation Algorithm) based on the Unicode Standard version 5.0.0.

The default Unicode Collation Algorithm is implemented by the UCA500R1 keyword without any attributes. Since the default UCA cannot simultaneously encompass the collating sequence of every language supported by Unicode, optional attributes can be specified to customize the UCA ordering. The attributes are separated by the underscore (_) character. The UCA500R1 keyword and any attributes form a UCA collation name.

The following table describes the collation attributes, their values, and typical usage examples.

Table 1. UCA500R1 attributes

Attribute name	Attribute short form	Valid values	Description
Locale: 1. Language 2. Region 3. Script 4. Keyword	Locale: 1. L[ISO 639-1 language code] 2. R[ISO 3166 country/region code] 3. Z[ISO 15924 script code] 4. K[name]	See Table 2 for a list of all the valid Locale names.	The Locale attribute is probably the most important attribute to obtain ordering that conforms to the user expectations in different countries and regions. You need to explicitly specify the Locale attribute to properly collate text for a specific language. The Locale attribute consists of the following parts: language, region/country, script, and keyword. Not all the parts are mandatory. See Table 2 for a complete list of the valid combinations. The specification of a locale automatically presets all the other collation attributes to values that are suitable for that locale. Typically there is no need to specify additional collation attribute. Examples: • UCA500R1 or UCA500R1_LROOT for the default UCA ordering • UCA500R1_LDE for German, where "Köpfe" < "Kypper" • UCA500R1_LSV for Swedish, where "Köpfe" > "Kypper" • UCA500R1_LDE_KPHONEBOOK, which specifies the German telephone ordering
Strength	S	1, 2, 3, 4, or I	The Strength attribute determines whether accent or case is taken into account when collating or comparing text strings. In writing systems without case or accent, the Strength attribute controls similarly important features. The possible values are: primary (1), secondary (2), tertiary (3), quatern and identity (I). To ignor • accent and case • case only, use t • neither accent n Almost all characters c therefore in most locale However if the Alternate quaternary strength lev characters, punctuation The identity strength level is used to distinguish among similar characters, such as the MATHEMATICAL BOLD SMALL A character (U+1D41A) and the MATHEMATICAL ITALIC SMALL A character (U+1D44E). Setting the Strength attribute to higher level will slow down text string comparisons and increase the length of the sort keys. Examples: • UCA500R1_S1 will collate "role" = "Role" = "rôle" • UCA500R1_S2 will collate "role" = "Role" < "rôle" • UCA500R1_S3 will collate "role" < "Role" < "rôle"
Case Level	E	• X (Off) • O (On)	Setting the Case Level attribute to on and the Strength attribute to primary level will ignore accent but not case. The Case Level attribute is set to X by default in most locales. When this attributes is set to O, it will slightly affect text string comparisons performance and lengthen the sort keys. Examples: • UCA500R1_EX_S1 will collate "role" = "Role" = "rôle" • UCA500R1_EO_S1 will collate "role" = "rôle" < "Role"
Case First	C	X, L, or U	The Case First attribute controls whether upper case characters collate before or after lower case characters, in the absence of other differences in the two text strings. The possible values are upper case first (U), lower case first (L), and off (X). There is almost no difference between the lower case first setting and the off setting, therefore typically there is no need to use the lower case first setting. Specifying a Case First attribute of U or L can increase the length of the sort keys. Examples: • UCA500R1_CX or UCA500R1_CL will collate "china" < "China" < "denmark" < "Denmark" • UCA500R1_CU will collate "China" < "china" < "Denmark" < "denmark"
Alternate	A	N or S	The Alternate attribute controls the handling of variable characters in the UCA: white space, punctuation marks, and symbols. If the Alternate attribute is set to non-ignorable (N), then differences among these variable characters are of the same importance as differences among non-variable characters such as the English alphabet. If the Alternate attribute is set to shifted (S), then these variable characters are of only minor importance. If the Alternate attribute is set to shifted and the Strength attribute is set to the quaternary level, then variable characters are considered in a comparison when all other aspects of the strings — base letters, accents, and case — are identical. The default for most locales is non-ignorable. If shifted is selected, performance will be slower if there are many strings that are identical except for punctuation marks. Sort key length will not be affected unless the strength level is also increased. Examples: • UCA500R1_AN_S3 will collate "di Silva" < "Di Silva" < "diSilva" < "U.S.A." < "USA" • UCA500R1_AS_S3 will collate "di Silva" = "diSilva" < "Di Silva" < "U.S.A." = "USA" • UCA500R1_AS_S4 will collate "di Silva" < "diSilva" < "Di Silva" < "U.S.A." < "USA"
Variable Top	T	[4 or 8 UTF-16BE hexadecimal digits]	The Variable Top attribute controls which characters to ignore, and is only meaningful if the Alternate attribute is set to Shifted. All characters whose primary weight is equal or lower than the specified character are considered ignorable. The character is specified as one or two UTF-16BE code units in hexadecimal notation. A Unicode supplementary character is specified using a surrogate pair. For example, if you want to ignore white space characters and not visible characters, then set the Alternate attribute to Shifted and this attribute to U+0020 (space) or U+3000 (ideographic space). Since all characters having the same primary weight are equivalent, so setting this attribute to U+0020 is equivalent to setting it to U+3000. This attribute alone has little impact on text string comparison performance, but setting it higher makes sort keys longer. Example: • UCA500R1_AS_S3 will collate "di Silva" = "diSilva" < "U.S.A." = "USA" • UCA500R1_AS_S3_T0020 will collate "di Silva" = "diSilva" < "U.S.A." = "USA"
Normalization Checking	N	• X (Off) • O (On)	The Normalization Checking attribute, if set to O, will normalize the input text if necessary. Even if this attribute is set to X, as is the default for many locales, text as represented in common usage will collate correctly. You should, however, set this attribute to O in two cases: • if the text contains accent marks in non-canonical order • if the text is in a script that uses multiple combining characters, such as Arabic, ancient Greek, Hebrew, Hindi, Thai, or Vietnamese There is a medium string comparison performance cost if this attribute is set to on, depending on the frequency of sequences that require normalization. There is no significant effect on length of the sort keys. If the text is already in normalized form NFD or NFKD, then you can set this attribute off to improve performance. Examples: • UCA500R1_NX will collate ä = a + ◌̈ < ä + ◌̣ < ạ + ◌̈ • UCA500R1_NO will collate ä = a + ◌̈ < ä + ◌̣ = ạ + ◌̈
French	F	• X (Off) • O (On)	The French sorts strings by examining the accents starting from the end of the string. This attribute is automatically set to on for the French locales, and has a minor performance cost for text string comparisons, but no change in the length of the sort keys. Examples: • UCA500R1_LFR_FX will collate "cote" < "coté" < "côte" < "côté" • UCA500R1_LFR will collate "cote" < "côte" < "coté" < "côté"
Hiragana	H	• X (Off) • O (On)	The Hiragana attribute determines whether to distinguish between Japanese Hiragana and Katakana characters. To conform with the Japanese JIS X 4061 standard, you need to set this attribute to O and the Strength attribute to the quaternary level. This will, however, slow down text string comparisons and increase the length of the sort keys. Examples: • UCA500R1_LJA_HX_S4 will collate "きゅう" = "キュウ" < "きゆう" = " "キユウ" • UCA500R1_LJA_HO_S4 will collate "きゅう" < "キュウ" < "きゆう" < "キユウ"

Contact Us

Contact IBM Knowledge Center

Email IBM Knowledge Center
ibmkc@us.ibm.com

Feedback

Valid locale names for the collations are shown in Table 2. The Default collation attributes column shows the full name of the UCA500R1 collation for the specific locale. For example, UCA500R1_LAR is equivalent to UCA500R1_LAR_AN_CX_EX_FX_HX_NX_S3.

All the UCA500R1 collations conform to version 1.5.1 of the Common Locale Data Repository (CLDR), as published by the Unicode Consortium at http://www.unicode.org/cldr.

Tip: If a locale name is not listed below, try the LROOT locale instead. While the LROOT locale does not always yield the correct collation for all unlisted locales, it may result in the expected order for some locales.

Table 2. Valid collation locale names

Locale name	Language (Region)	Default collation attributes	Remarks
LAR	Arabic	UCA500R1_LAR_AN_CX_EX_FX_HX_NX_S3	
LAS	Assamese	UCA500R1_LAS_AN_CX_EX_FX_HX_NO_S3	
LBE	Belarusian	UCA500R1_LBE_AN_CX_EX_FX_HX_NX_S3	
LBG	Bulgarian	UCA500R1_LBG_AN_CX_EX_FX_HX_NX_S3	
LCA	Catalan	UCA500R1_LCA_AN_CX_EX_FO_HX_NX_S3	
LCS	Czech	UCA500R1_LCS_AN_CX_EX_FX_HX_NX_S3	
LDA	Danish	UCA500R1_LDA_AN_CU_EX_FX_HX_NX_S3	
LDE	German	UCA500R1_LDE_AN_CX_EX_FX_HX_NX_S3	
LDE_KPHONEBOOK	German	UCA500R1_LDE_KPHONEBOOK_AN_CX_EX_FX_HX_NX_S3	
LEL	Greek	UCA500R1_LEL_AN_CX_EX_FX_HX_NO_S3	
LEN	English	UCA500R1_LEN_AN_CX_EX_FX_HX_NX_S3	
LEN_RBE	English (Belgium)	UCA500R1_LEN_RBE_AN_CX_EX_FO_HX_NX_S3	
LEO	Esperanto	UCA500R1_LEO_AN_CX_EX_FX_HX_NX_S3	
LES	Spanish	UCA500R1_LES_AN_CX_EX_FX_HX_NX_S3	
LES_KTRADITIONAL	Spanish	UCA500R1_LES_KTRADITIONAL_AN_CX_EX_FX_HX_NX_S3	
LET	Estonian	UCA500R1_LET_AN_CX_EX_FX_HX_NX_S3	
LFA	Persian	UCA500R1_LFA_AN_CX_EX_FX_HX_NO_S3	
LFA_RAF	Persian (Afghanistan)	UCA500R1_LFA_RAF_AN_CX_EX_FX_HX_NO_S3	
LFI	Finnish	UCA500R1_LFI_AN_CX_EX_FX_HX_NX_S3	
LFO	Faroese	UCA500R1_LFO_AN_CX_EX_FX_HX_NX_S3	
LFR	French	UCA500R1_LFR_AN_CX_EX_FO_HX_NX_S3	
LGU	Gujarati	UCA500R1_LGU_AN_CX_EX_FX_HX_NO_S3	
LHAW	Hawaiian	UCA500R1_LHAW_AN_CX_EX_FX_HX_NX_S3	
LHE	Hebrew	UCA500R1_LHE_AN_CX_EX_FX_HX_NO_S3	
LHI	Hindi	UCA500R1_LHI_AN_CX_EX_FX_HX_NO_S3	
LHI_KDIRECT	Hindi	UCA500R1_LHI_KDIRECT_AN_CX_EX_FX_HX_NX_S3	
LHR	Croatian	UCA500R1_LHR_AN_CX_EX_FX_HX_NX_S3	
LHU	Hungarian	UCA500R1_LHU_AN_CX_EX_FX_HX_NX_S3	
LIS	Icelandic	UCA500R1_LIS_AN_CX_EX_FX_HX_NX_S3	
LIT	Italian	UCA500R1_LIT_AN_CX_EX_FX_HX_NX_S3	
LJA	Japanese	UCA500R1_LJA_AN_CX_EX_FX_HO_NX_S3	Treat Hiragana as equal to their Katakana equivalents. To sort Hiragana before Katakana, set the strength level to 4.
LJA_KUNIHAN	Japanese	UCA500R1_LJA_KUNIHAN_AN_CX_EX_FX_HX_NX_S3	
LKK	Kazakh	UCA500R1_LKK_AN_CX_EX_FX_HX_NO_S3	
LKL	Kalaallisut	UCA500R1_LKL_AN_CX_EX_FX_HX_NX_S3	
LKM	Khmer	UCA500R1_LKM_AN_CX_EX_FX_HX_NO_S3	
LKN	Kannada	UCA500R1_LKN_AN_CX_EX_FX_HX_NO_S3	
LKO	Korean	UCA500R1_LKO_AN_CX_EX_FX_HX_NX_S3	
LKO_KUNIHAN	Korean	UCA500R1_LKO_KUNIHAN_AN_CX_EX_FX_HX_NX_S3	
LLT	Lithuanian	UCA500R1_LLT_AN_CX_EX_FX_HX_NX_S3	
LLV	Latvian	UCA500R1_LLV_AN_CX_EX_FX_HX_NX_S3	
LMK	Macedonian	UCA500R1_LMK_AN_CX_EX_FX_HX_NX_S3	
LML	Malayalam	UCA500R1_LML_AN_CX_EX_FX_HX_NO_S3	
LMR	Marathi	UCA500R1_LMR_AN_CX_EX_FX_HX_NO_S3	
LMT	Maltese	UCA500R1_LMT_AN_CU_EX_FX_HX_NX_S3	
LNB	Norwegian Bokmål	UCA500R1_LNB_AN_CX_EX_FX_HX_NX_S3	
LNN	Norwegian Nynorsk	UCA500R1_LNN_AN_CX_EX_FX_HX_NX_S3	
LOM	Oromo	UCA500R1_LOM_AN_CX_EX_FX_HX_NX_S3	
LOR	Oriya	UCA500R1_LOR_AN_CX_EX_FX_HX_NO_S3	
LPA	Punjabi	UCA500R1_LPA_AN_CX_EX_FX_HX_NO_S3	
LPL	Polish	UCA500R1_LPL_AN_CX_EX_FX_HX_NX_S3	
LPS	Pashto	UCA500R1_LPS_AN_CX_EX_FX_HX_NO_S3	
LRO	Romanian	UCA500R1_LRO_AN_CX_EX_FX_HX_NX_S3	
LROOT	Root	UCA500R1_LROOT_AN_CX_EX_FX_HX_NX_S3	Default UCA
LRU	Russian	UCA500R1_LRU_AN_CX_EX_FX_HX_NX_S3	
LSK	Slovak	UCA500R1_LSK_AN_CX_EX_FX_HX_NX_S3	
LSL	Slovenian	UCA500R1_LSL_AN_CX_EX_FX_HX_NX_S3	
LSQ	Albanian	UCA500R1_LSQ_AN_CX_EX_FX_HX_NX_S3	
LSR	Serbian	UCA500R1_LSR_AN_CX_EX_FX_HX_NX_S3	
LSR_ZLATN	Serbian	UCA500R1_LSR_ZLATN_AN_CX_EX_FX_HX_NX_S3	
LSV	Swedish	UCA500R1_LSV_AN_CX_EX_FX_HX_NX_S3	
LTA	Tamil	UCA500R1_LTA_AN_CX_EX_FX_HX_NO_S3	
LTE	Telugu	UCA500R1_LTE_AN_CX_EX_FX_HX_NO_S3	
LTH	Thai	UCA500R1_LTH_AN_CX_EX_FX_HX_NO_S3	
LTR	Turkish	UCA500R1_LTR_AN_CX_EX_FX_HX_NX_S3	
LUK	Ukrainian	UCA500R1_LUK_AN_CX_EX_FX_HX_NX_S3	
LVI	Vietnamese	UCA500R1_LVI_AN_CX_EX_FX_HX_NO_S3	
LZH	Chinese	UCA500R1_LZH_AN_CX_EX_FX_HX_NX_S3	Pinyin ordering
LZH_KUNIHAN	Chinese	UCA500R1_LZH_KUNIHAN_AN_CX_EX_FX_HX_NX_S3	Default UCA ordering
LZH_KBIG5HAN	Chinese	UCA500R1_LZH_KBIG5HAN_AN_CX_EX_FX_HX_NX_S3	Big5 ordering
LZH_KGB2312HAN	Chinese	UCA500R1_LZH_KGB2312HAN_AN_CX_EX_FX_HX_NX_S3	GB2312 ordering
LZH_KSTROKE	Chinese	UCA500R1_LZH_KSTROKE_AN_CX_EX_FX_HX_NX_S3	Stroke ordering

The UCA400_NO, UCA400_LSK, and UCA400_LTH collations from DB2® database versions earlier than Version 9.5 are still supported when creating databases. However, these collations are not supported in the COLLATION_KEY_BIT function.

The UCA400R1 collations from DB2 database versions earlier than Version 9.5 Fix Pack 1 are still supported in the COLLATION_KEY_BIT function. However these collations are not supported when creating databases.

In Unicode, most accented characters can be represented in multiple ways. For example, the character Ö can be represented as one code point, X'00D6' (Latin capital letter O with diaeresis) or as two code points, X'004F' X'0308' (Latin capital letter O followed by combining diaeresis). The collations UCA400_NO, UCA400_LSK, and UCA400_LTH always distinguish between different representations of a character.

For example, consider the ordering of O and the two different representations of Ö:

- In UCA400_NO, 'O' < X'004F' X'0308' < X'00D6'.
- In UCA500R1_NO, 'O' < X'004F' X'0308' = X'00D6'.

Details of the Unicode Collation Algorithm can be found in the Unicode Technical Standard #10, available at the Unicode Consortium web site at http://www.unicode.org.

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

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 UCA500R1_LTH collation option.

Related reference:
COLLATION_KEY_BIT scalar function
CREATE DATABASE command

Reference topic