FUJITSU

White Paper
Converting from 7-bit to 8-bit character set in BS2000

When processing data, the BS2000 system uses a 7-bit EBCDIC character set as standard, making 95 different characters available. With XHCS, BS2000 additionally supports 8-bit character sets as well as 7-bit and Unicode variants. With it, BS2000 can represent amongst other all the national languages defined in the international code tables conforming to ISO 8859.
Use of a specific 8-bit EBCDIC code can be set for the entire BS2000 system or for individual user IDs and pubsets. The present white paper describes what a customer must do today and in the future in order to migrate from a 7-bit environment to an 8-bit environment.

Inhalt

Character sets

In order to map characters (glyphs), each character is assigned a hexadecimal number. This assignment takes place, for example, when the character is input via the keyboard or in the keyboard driver. The code used is referred to as a 7- or 8-bit code, according to the number of bits used to encode the characters. To be perfectly accurate, this definition only applies to the ISO codes, for in BS2000 the EBCDIC codes EDF03IRV and EDF03DRV are also designated as 7-bit codes, even though they use 8 bits for encoding. This convention has nonetheless become established because these codes contain only 95 printable characters and in the case of EDF03IRV these correspond to the ASCII 7-bit character set.

A character set is characterized on the one hand by its character repertoire and on the other by the manner in which a character is mapped to a hexadecimal value. Two character sets are described as compatible if they contain identical characters and only the hexadecimal encoding of the characters differs from one set to the other.

The rule for compatible character sets is: All characters can be translated into one another without loss.

Thus, for example, the ISO8859-1 and EBCDIC.DF.04-01 (EDF041) character sets are compatible. The character sets EDF03IRV and EDF03DRV, by contrast, are not compatible! EDF03IRV recognizes no German umlauts, while EDF03DRV does not recognize braces or square brackets, or indeed the tilde character or backslash. Although the character repertoire of both character sets is a true subset of the EDF041 character set, they are not compatible with the latter.

Setting character sets in BS2000

Provided only one character set is used and all participating entities (terminal, monitor, printer, BS2000, file transfer partner, etc.) proceed on the basis of the same character set, the specification of the character set is irrelevant. But as soon as different character sets are involved, even if they are compatible with one another, it must be clear which character set is meant at a given moment. Knowledge of the character set is essential to the correct interpretation of a text field.

BS2000 command processor

Command texts and control characters are stored in binary encoded form in the BS2000 command processor and in the utility routines. Only commands that have the same binary encoding are recognized.

The characters required for this form what is referred to as the "EBCDIC core". The hexadecimal value of each character of the EBCDIC core is fixed and must not be changed. For this reason, every BS2000 EBCDIC table contains the EBCDIC core.

The DEFAULT (i.e. standard value) for the BS2000 commands /ADD-USER and /MODIFY-USER-ATTRIBUTES is specified using the CLASS2 parameter HOSTCODE=<ccsn>. This value is also interpreted by the XHCS component if the value *SYSDEF is specified for the CCS name at the XHCS interface. The parameter is set to the value EDF03IRV by default.

Terminal emulation <-> VTSU protocol

Connection setup time

During the connection setup the station characteristics of the terminal (emulation) are communicated to VTSU by default, and in the case of the 8-bit-capable terminal types (9759, 9763) VTSU subsequently interrogates the terminal or the emulation for all further attributes, e.g. supported code variants. In this process, available 8-bit ISO8859 variants as well as 7-bit ASCII etc. are also reported.

VTSU notes these attributes in its tables. The system and customer applications can in turn request this information from VTSU with the aid of the /SHOW-TERMINAL-ATTRIBUTES *ALL command or using the TSTAT macro.

Normal dialog time

From the emulation viewpoint, the initiative always lies with VTSU. If VTSU sends a message to the emulation in a particular character set, the emulation must respond using the same character set. More specifically: VTSU communicates with the emulation in the ISO variant, the message is encoded. The transport layer then converts the EBCDIC message from VTSU to ISO8859 or the ISO8859 message of the emulation to EBCDIC. The variant can be changed after each communication step.

By default, the 7-bit variant is active after LOGON, i.e. EDF03IRV is set on the BS2000 side.

The assignment of this default value can be changed with the aid of the VTSU-B parameter file ($TSOS.SYSPAR.VTSU-B.<vtsu-version>). The parameter UTM-PERM8=Y, DCAM-PERM8=Y or TIAM-PERM8=Y must be set, depending on whether the partner application is a UTM, DCAM or TIAM application ($DIALOG). Once the parameters have been set, the character set of the HOME pubset of the user ID determines the ISO variant.

To obtain the same effect for a process, the command /MODIFY-TERMINAL-OPTIONS CODE-CHARACTER-SET=*8BIT-DEFAULT is issued.
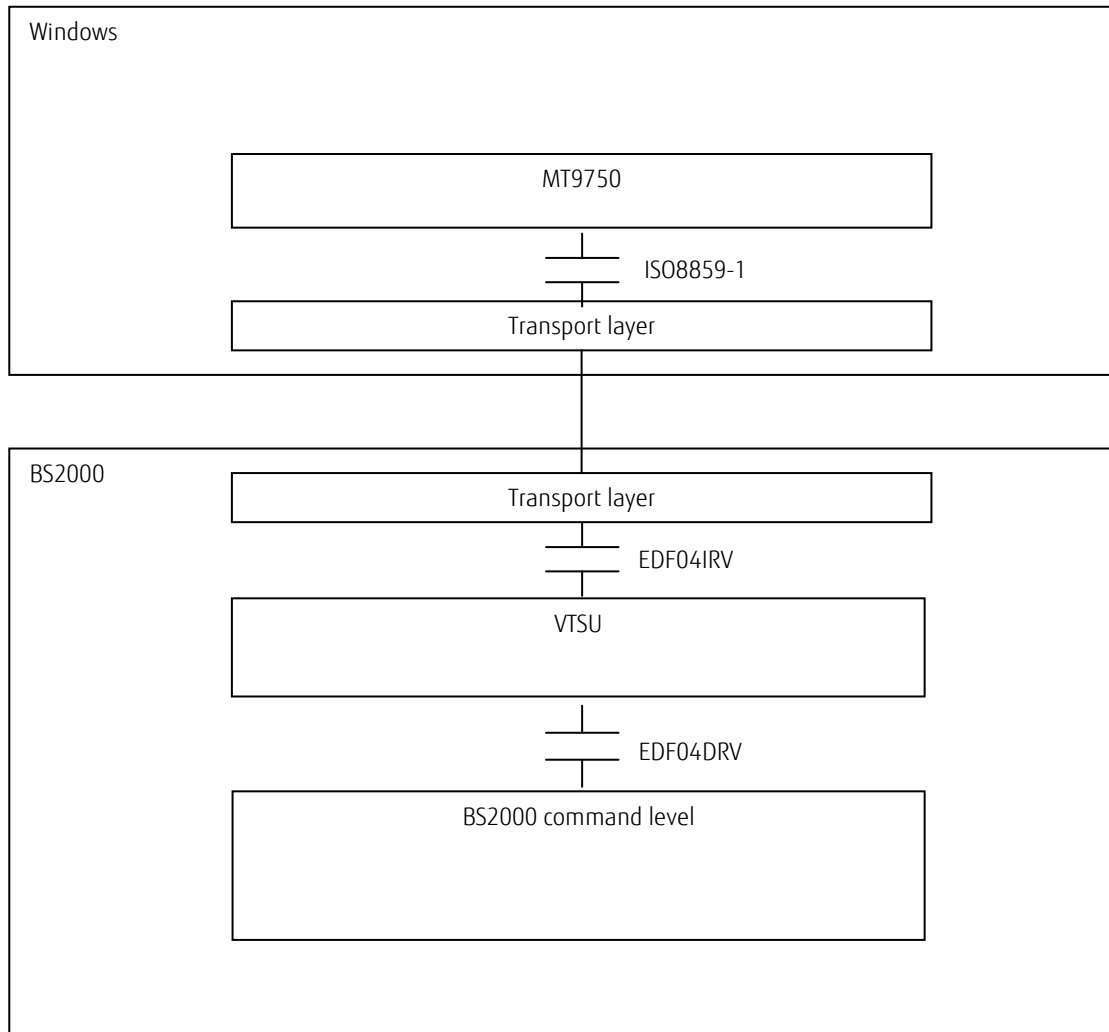
The /MODIFY-TERMINAL-OPTIONS command also permits any other character set to be set provided it is supported by the terminal.

If the character set is not what is referred to as a "reference" character set, i.e. a character set which virtually 'automatically' becomes a character set that is supported by the terminal emulation as a result of the conversion in the transport layer, the conversion into the reference character set takes place in VTSU with the aid of XHCS (see Annex: table of EBCDIC codesets supported by XHCS).

Within a program, the character set can be passed to VTSU with each call via the VTSU-B control block. If the character set changes between two calls, this results in the screen being cleared. Explicit specification in the VTSU-B control block has priority over all other settings.

Example:
Data exchange between MT9750 and VTSU after the /MODIFY-TERMINAL-OPTIONS CODE-CHARACTER-SET=EDF04DRV command is issued

```
Windows

                    ┌───────────────────────────┐
                    │           MT9750           │
                    └───────────────────────────┘
                              │   ISO8859-1
                    ┌───────────────────────────┐
                    │      Transport layer       │
                    └───────────────────────────┘
                              │

BS2000
                    ┌───────────────────────────┐
                    │      Transport layer       │
                    └───────────────────────────┘
                              │   EDF04IRV
                    ┌───────────────────────────┐
                    │            VTSU            │
                    └───────────────────────────┘
                              │   EDF04DRV
                    ┌───────────────────────────┐
                    │    BS2000 command level    │
                    │                            │
                    └───────────────────────────┘
```

Behavior of openFT

When files are transferred from one BS2000 computer to another BS2000 computer, the file attribute CCSN is also transferred along with the file contents. The transfer takes place in binary format without conversion. If, on the other hand, a text file is to be sent to a non-BS2000 system (Unix, Windows), a conversion into the reference code must first be performed using XHCS. The conversion from EBCDIC to ISO is then performed during the subsequent transfer. In other words, if a file is present in EDF04DRV, it is first converted to EDF04IRV in order then to be converted to ISO8859-1 during the transfer and stored in the other system.
If the CCS name of the file is *NONE, no reference code is known, i.e. no conversion into the reference code takes place, and the transfer with conversion from EBCDIC to ISO takes place immediately.
If the source code is not compatible, but only a true subset of the reference code, e.g. EDF03IRV of EDF041, the conversion into the reference code leads to data loss for characters or hex values that are outside the scope of EDF03IRV.
If a text file is transferred from a non-BS2000 system into BS2000, it initially arrives in BS2000 in the EBCDIC variant corresponding to the source code. If the CCS name of the user entry on the storing pubset is compatible with this EBCDIC variant, the file is then converted into the corresponding character set, and the CCS name of the file is supplied accordingly. If no conversion can be performed, the file is given the CCS name of the reference character set.
If the CCSN of the user entry is EDF03IRV, the file is assigned *NONE as the CCSN.

Assigning the CCS name to the user ID

When a user ID is created, a CCS name is entered in the user entry of the HOME pubset (/ADD-USER command). If no value is specified explicitly, the value from the CLASS2 option HOSTCODE is used. If further pubsets are assigned to the user ID, a CCS name is likewise entered in the associated user entries on the pubsets according to the same method. The CCS names may all be different.

Evaluating the CCS names in the user entry of the pubset

As described above, the CCS name of the user entry in the HOME pubset is used by VTSU to set the default character set.
The Data Management System uses the CCS name in the user entry of the pubset as the default CCS name when creating new files on this pubset.
This value is also interpreted at the XHCS interface if the value *USRDEF has been specified for the CCS name.

CCS name assignment for BS2000 files

The BS2000 file attribute CCSN (Coded Character Set Name) describes the binary encoding of characters within a file. It conveys no information about other binary patterns. In other words, a file consisting not just of text and control characters contained in the character set cannot be converted without knowledge of the structure.
Up to OSD V6.0B the user was unable to influence the default value assignment of the CCS name of a file. The CCS name *NONE was always assigned by the BS2000 file system. To avoid the need to adapt all the programs when a conversion is planned, there is an optional rep (A0296403) which, when a new file is being created on a pubset, uses the CCS name of the user entry of the pubset for the file.
If a file is given the CCS name EDF03IRV on account of this rep, when it is printed or transferred using openFT this can lead to data loss in the case of hex values that lie outside the EDF03IRV character set (see section on openFT).
Starting with the second OSD V6.0B correction package 2006, it is possible to affect the default value of the CCS name of a file as follows:
For a newly created file, the CCS name of the user entry of the receiving pubset will be adopted as the CCS name of the file if this is not EDF03IRV. If it is EDF03IRV, the file receives the CCS name *NONE.
This arrangement permits the default value to be defined on a user- and pubset-specific basis. The optional rep is no longer required.
However, the basic rule remains that an explicit specification of a CCS name always takes precedence.
If a file is copied, backed up or restored, the CCSN file attribute is always transferred as well.

Please note: The command /MODIFY-FILE-ATTRIBUTES <filename>, C-C-S=<ccs-name> usually modifies the interpretation of the contents of the file, since the hex values may be assigned to other characters (glyphs) on account of the new CCS name. Consequently, this command should only be issued if the above-described behavior is desired. The operating system can provide no support in this case.

Please note: Private disks do not support the file attribute, so they can only be used for accommodating files with *NONE or EDF03IRV.

CCS name assignment in LMS libraries

When a BS2000 file is entered in an LMS library, the element receives the CCS name of the source file. When an element is extracted from an LMS library, the extracted file is given the CCSN of the element. If a new element is created, e.g. by writing the output from SYSLST into a library, the new element is given the CCS name of the library file (with PLAM correction A0536536), if no CCS name was specified explicitly in the /ASSIGN-SYSLST command. Whether the CCS name corresponds to the character set actually used cannot be decided by the system. It is the user's responsibility to ensure consistency.

EDT (V16.6B)

EDT does not convert any data, but rather makes sure that the CCSN of the terminal setting and that of the files (when there are a number of files in different windows) are identical, i.e. that no undefined mixed files are created inadvertently. Input in hex mode is always permitted, however.
If a hexadecimal input of this kind is subsequently interpreted as characters, this is done in accordance with the current CCS name of the file. When EDT is loaded, the so-called "actual" CCS name is determined from the currently set VTSU parameter. Provided no data is yet present in any EDT window, the CCS name can be changed using the @CODENAME statement. The rule in this case is that the newly set codeset must be compatible with the terminal. If a new file is now created, it is given the actual CCS name.
If no data is present in any EDT window and an existing file is opened, the actual CCSN becomes identical to that of the file, once again on condition that this is compatible with the terminal. If the input is performed by means of a procedure file, the CCS name of the procedure file determines the actual CCS name.
If a file has the CCS name *NONE, EDT handles it like a file with the CCS name EDF03IRV.
This strict check by EDT makes it difficult to handle files with different character sets simultaneously, with one character set being a true subset of the other character set. For this purpose, before EDT is called, the user must establish the identity of the CCS names using a /MODIFY-FILE-ATTRIBUTES command.

EDT (V17.0)

With EDT V17.0, the option of a true conversion of the data is supported. In the EDT V17.0 Unicode mode, a different character set can be set for each work file. This makes it possible to keep files with different coding open in parallel in different windows and to process them.

The character sets for the individual work files are either implicitly set when reading in an appropriately coded file or explicitly via the @CODENAME command.
In addition, EDT possesses its own character set – the communication character set – which it uses to communicate with the terminal. This can be different from the character set used in any work file in which data is stored.

Checklist for converting from a 7-bit character set to an 8-bit character set
The following points should be taken into account for a conversion:

Analysis of the source data
- Is the target character set a superset of the source character set?
- If yes, then renaming is sufficient.
  The /MODIFY-FILE-ATTRIBUTES command is sufficient for all files, or the LMS command MODIFY-ELEMENT-ATTRIBUTES for the LMS library elements
- If no, then a conversion of the data must take place.
  If a conversion is necessary,
  - If just the data is affected:
    Must archived data also be handled?
    > No: if the conversion can be performed when the data is loaded, or the data has to be retained for legal reasons only and programs required for this still 'understand' the data.
    > Yes: if the conversion cannot take place automatically when the data is loaded.
  - If programs or procedures are also affected:
    > Conversion of the archived data if the programs and procedures cannot be modified compatibly.
    > Modification of the loading procedures with possible conversion.

Analysis of the external interfaces
Where is the conversion of incoming and outgoing data carried out; which conversion may need to be adapted?
Possible candidates are:
- Terminal emulation
- Possibly WebTransactions
- Spool
- File transfer
- Tapes, magnetic tape cartridges

Setting the system
Is the entire system to be converted?
- If yes:
  - The CLASS2 parameter HOSTCODE=<desired ccsn> helps avoid user IDs being created in the old character set.
  - Set the parameters UTM-PERM8=Y, TIAM-PERM8=Y, DCAM-PERM8=Y in the VTSU-B parameter file
- If no:
  - Insert the /MODIFY-TERMINAL-OPTIONS CODE-CHARACTER-SET=*8BIT-DEFAULT command in the LOGON procedure of the affected user IDs.
  - Set user ID and each pubset associated with the user ID to the desired CCS name using the /MODIFY-USER-ATTRIBUTES command.

Annex

Overview of the default CCS assignment

	DEFAULT VALUE former behavior	DEFAULT VALUE behavior as of BS2000/OSD-BC V6.0B (CP 2/2006)	Comments
CLASS-2 option HOSTCODE	EDF03IRV	EDF03IRV	
ADD-USER	From CLASS-2 option HOSTCODE	From CLASS-2 option HOSTCODE	
MODFY-USER-ATTR CCS=*STD	From CLASS-2 option HOSTCODE	From CLASS-2 option HOSTCODE	
VTSU	TIAM/UTM/DCAM-PTERM8=N: ■ EDF03IRV TIAM/UTM/DCAM-PTERM8=Y: ■ 8-bit terminal: CCSN of the HOME pubset ■ 7-bit TERMINAL CCSN=EDF03IRV	TIAM/UTM/DCAM-PTERM8=N: ■ EDF03IRV TIAM/UTM/DCAM-PTERM8=Y: ■ 8-bit terminal: CCSN of the HOME pubset ■ 7-bit TERMINAL CCSN=EDF03IRV	The parameters TIAM-PTERM8, UTM-PTERM8 and DCAM-PTERM8 reside in the VTSU parameter file under TSOS. They are valid system-wide
MODIFY-TERM-OPT CCS=*8-bit-DEFAULT	8-bit TERMINAL: ■ CCSN=CCSN of the HOME pubset 7-bit TERMINAL: ■ Error message	8-bit TERMINAL: ■ CCSN=CCSN of the HOME pubset 7-bit TERMINAL: ■ Error message	A CCSN can also be specified explicitly, though it must be supported by the terminal.
EDT	After loading of EDT ■ in DIALOG mode the CCSN matches the CCSN set in the VTSU ■ in BATCH mode the CCSN matches the CCSN of the allocated statement input file	After loading of EDT ■ in DIALOG mode the CCSN matches the CCSN set in the VTSU ■ in BATCH mode the CCSN matches the CCSN of the allocated statement input file	EDT V17.0 in Unicode mode: Each window can have data with a separate CCSN. If data is copied from one window to another, the data is converted. EDT V17.0 in compatibility mode and EDT V16.6: In interactive mode, when a file is imported, the CCSN can change to that of the file, as long as there is no data in either of the EDT windows. This means that all EDT windows can only contain data with the same CCSN.
openFT	CCSN (pubset of the file) = EDF03IRV => CCSN of the file = *NONE CCSN (pubset of the file) !=EDF03IRV => CCSN of the file = CCSN of the pubset	CCSN (pubset of the file) = EDF03IRV => CCSN of the file = *NONE CCSN (pubset of the file) !=EDF03IRV => CCSN of the file = CCSN of the pubset	
BS2000 creation of a file CREATE-FILE CREATE-FILE-GROUP MODIFY-FILE-ATTRIBUTES MODIFY-FILE-GROUPATTRIBUTES	Previously always CCSN=*NONE With optional rep A0296403: ■ always CCSN file = CCSN of the pubset ■ *NONE is no longer assigned by the system	CCSN (pubset of the file) = EDF03IRV => CCSN of the file = *NONE CCSN (pubset of the file) != EDF03IRV => The file receives the CCSN of the user entry of the PVS.	

	DEFAULT VALUE former behavior	DEFAULT VALUE **behavior as of BS2000/OSD-BC V6.0B (CP 2/2006)**	Comments
LMS EDIT-ELEMENT ELEMENT=*NONE TO-ELEMENT= <output>	After LMS-internal call of EDT, LMS sends an empty @CODENAME statement. EDF03IRV is always set by the sending of an empty @CODENAME statement.	The behavior of the empty @CODENAME statement changes with the 2nd correction package 2006 (OSD V6.0B): CCSN-VTSU is set (same behavior as after new start of EDT)	
PLAM	CCSN=*NONE	With correction A0536536: CCSN= CCSN of the PLAM library	

Tables: Converting from ISO8859 to EBCDIC (BS2000) and vice versa

ISO8859-n and EBCDIC.DF.04.n have the same character set. The conversion is performed using the following conversion table:

Table:	Converting from ISO8859 to EBCDIC (BS2000)														
	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	00	01	02	03	37	2D	2E	2F	16	05	25	0B	0C	0D	0E	0F
1-	10	11	12	13	3C	3D	32	26	18	19	3F	27	1C	1D	1E	1F
2-	40	5A	7F	7B	5B	6C	50	7D	4D	5D	5C	4E	6B	60	4B	61
3-	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	7A	5E	4C	7E	6E	6F
4-	7C	C1	C2	C3	C4	C5	C6	C7	C8	C9	D1	D2	D3	D4	D5	D6
5-	D7	D8	D9	E2	E3	E4	E5	E6	E7	E8	E9	BB	BC	BD	6A	6D
6-	4A	81	82	83	84	85	86	87	88	89	91	92	93	94	95	96
7-	97	98	99	A2	A3	A4	A5	A6	A7	A8	A9	FB	4F	FD	FF	07
8-	20	21	22	23	24	04	06	08	28	29	2A	2B	2C	09	0A	14
9-	30	31	15	33	34	35	36	17	38	38	3A	3B	1A	1B	3E	5F
A-	41	AA	B0	B1	9F	B2	D0	B5	79	B4	9A	8A	BA	CA	AF	A1
B-	90	8F	EA	FA	BE	A0	B6	B3	9D	DA	9B	8B	B7	B8	B9	AB
C-	64	65	62	66	63	67	9E	68	74	71	72	73	78	75	76	77
D-	AC	69	ED	EE	EB	EF	EC	BF	80	E0	FE	DD	FC	AD	AE	59
E-	44	45	42	46	43	47	9C	48	54	51	52	53	58	55	56	57
F-	8C	49	CD	CE	CB	CF	CC	E1	70	C0	DE	DB	DC	8D	8E	DF

Table:	Converting from EBCDIC (BS2000) to ISO8859														
	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	00	01	02	03	85	09	86	7F	87	8D	8E	0B	0C	0D	0E	0F
1-	10	11	12	13	8F	92	08	97	18	19	9C	9D	1C	1D	1E	1F
2-	80	81	82	83	84	0A	17	1B	88	89	8A	8B	8C	05	06	07
3-	90	91	16	93	94	95	96	04	98	99	9A	9B	14	15	9E	1A
4-	20	A0	E2	E4	E0	E1	E3	E5	E7	F1	60	2E	3C	28	2B	7C
5-	26	E9	EA	EB	E8	ED	EE	EF	EC	DF	21	24	2A	29	3B	9F
6-	2D	2F	C2	C4	C0	C1	C3	C5	C7	D1	5E	2C	25	5F	3E	3F
7-	F8	C9	CA	CB	C8	CD	CE	CF	CC	A8	3A	23	40	27	3D	22
8-	D8	61	62	63	64	65	66	67	68	69	AB	BB	F0	FD	FE	B1
9-	B0	6A	6B	6C	6D	6E	6F	70	71	72	AA	BA	E6	B8	C6	A4
A-	B5	AF	73	74	75	76	77	78	79	7A	A1	BF	D0	DD	DE	AE
B-	A2	A3	A5	B7	A9	A7	B6	BC	BD	BE	AC	5B	5C	5D	B4	D7
C-	F9	41	42	43	44	45	46	47	48	49	AD	F4	F6	F2	F3	F5
D-	A6	4A	4B	4C	4D	4E	4F	50	51	52	B9	FB	FC	DB	FA	FF
E-	D9	F7	53	54	55	56	57	58	59	5A	B2	D4	D6	D2	D3	D5
F-	30	31	32	33	34	35	36	37	38	39	B3	7B	DC	7D	DA	7E

Table of all EBCDIC codes supported by XHCS

Table of all EBCDIC codes supported by XHCS				
CCS of the table	Reference table	ISO variant	Terminal setting	Full name
EDF041	EDF041	1	ISO8859-1	
EDF04DRV				EBCDIC.DF.04-DRV
EDF04IRV				EBCDIC.DF.04-IRV
EDF03IRV		1 (7-bit international)		EBCDIC.DF.03-IRV
EDF03DRV		1 (7-bit German)		EBCDIC.DF.03-DRV
EDF042	EDF042	2 EAST-LATIN	ISO8859-2	EBCDIC.DF.04-2
EEHCL2				EBCDIC.EHC.L2
EDF043	EDF043	3 ESPERANTO, GALICIAN, MALTESE, TURKISH	ISO8859-3	EBCDIC.DF.04-3
EDF044	EDF044	4 ESTONIAN, LATVIAN, LITHUANIAN	ISO8859-4	EBCDIC.DF.04-4
EDF045	EDF045	5 LATIN/CYRILLIC	ISO8859-5	EBCDIC.DF.04-5
EEHCLC				EBCDIC.EHC.LC
EEHCLC1				EBCDIC.EHC.LC.1
EDF047	EDF047	7 LATIN/GREEK	ISO8859-7	EBCDIC.DF.04-7
EEHCLG				EBCDIC.EHC.LG
EDF049	EDF049	9 LATIN ALPHABET #5	ISO8859-9	EBCDIC.DF.04-9 = EBCDIC.DF.04.L5
EDF04F	EDF04F	15 (LATIN WEST – EURO)	ISO8859-15	EBCDIC.DF.04-F

XHCS tables which are not generated by default and which require special terminals or character sets				
CCS of the table	Reference table	ISO variant	Terminal setting	Full name
EEHCSB	EEHCSB	1 MODIFY FEDERAL STATISTICS OFFICE	Private codeset	EBCDIC.EHC.SB
EDF041SB	EDF041SB	1 FEDERAL STATISTICS OFFICE		EBCDIC.DF.04-1 MODIF
EDF049BE	EDF049BE	9 DIN 31628/2		EBCDIC.DF.04.BIB
EDF04B	EDF04B	B		EBCDIC.DF.04.BIB.9756
EDF046	EDF046	6 (ARAB NUMERIC)		EBCDIC.DF.04-6
EEHCLA		A (INDIAN NUMERIC)		EBCDIC.EHC.LA
EDF04E	EDF04E	E (ARAB NUMERIC)		EBCDIC.DF.04-NAF
EEHCNA		F (INDIAN NUMERIC)		EBCDIC.EHC.NA
EDF04C	EDF04C	C (INTERNATIONAL NUMERIC)		EBCDIC.DF.04-FAR
EEHCLF		D (FARSI NUMERIC)		EBCDIC.EHC.LF

Contact:
Fujitsu
Barbara Stadler
Mies-van-der-Rohe-Str. 8, 80807 Munich
Germany
Telephone: +49 (0) 89 62060-1978
Email: Barbara.stadler@ts.fujitsu.com
Website: de.fujitsu.com
September, 2 2015 EM EN