XHCS V2.0

8-bit-Code- und Unicode-Unterstützung im BS2000/OSD

Ausgabe März 2007

Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@fujitsu-siemens.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2000

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2000 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Inhalt

1 Einleitung

XHCS (E**x**tended **H**ost **C**ode **S**upport) ist ein Softwareprodukt zur Zeichenbehandlung für das Betriebssystem BS2000/OSD. Damit kann BS2000/OSD alle Landessprachen darstellen, die in den internationalen Code-Tabellen nach ISO 8859 definiert sind. Das sind alle west- und osteuropäischen Sprachen inklusive Kyrillisch, Griechisch und den baltischen Sprachen, sowie Arabisch, Persisch (Farsi) und Maltesisch. Darüber hinaus unterstützt XHCS den Unicode-Zeichensatz.

1.1 Zielgruppe

Das vorliegende Handbuch richtet sich an Anwender der Zugriffsmethoden DCAM, TIAM und openUTM, die in ihren Programmen die Vorteile eines erweiterten Standard-Zeichensatzes oder/und des Unicode-Zeichensatzes nutzen wollen, und an Systembetreuer, die vorgegebene Code-Tabellen gemäß ihren Anforderungen modifizieren wollen.

Vorausgesetzt werden Kenntnisse des BS2000/OSD, der Zugriffsmethoden DCAM, TIAM, openUTM, der Softwarekomponenten VTSU und gegebenenfalls FHS und IFG. Für Systembetreuer werden ferner Kenntnisse in der Systembetreuung für BS2000/OSD-BC vorausgesetzt.

Weiterführende Literatur ist am Ende dieses Buches im Literaturverzeichnis zusammengestellt. Neben den in diesem Handbuch beschriebenen Code-Tabellen werden auch die Code-Tabellen der unterstützten Terminals und Drucker benötigt.

1.2 Wegweiser durch das Handbuch

Dieses Handbuch beschreibt den Einsatz erweiterter Code-Tabellen sowie die Unterstützung des Unicode-Zeichensatzes und die Erstellung und Modifikation von Code-Tabellen.

Das Kapitel „Einführung in XHCS“
gibt eine kurzen Überblick, erklärt erweiterte Zeichensätze, die Besonderheiten der Unicode-Unterstützung in XHCS, und zeigt, wie XHCS den internationalen Einsatz für Anwendungen ermöglicht.

Das Kapitel „XHCS“
beschreibt Konzept und Vorteile von XHCS, die Komponenten und Programmschnittstellen und erklärt die praktische Anwendung von XHCS.

Das Kapitel „XHCS-Makros“
enthält eine detaillierte Beschreibung der XHCS-Makros.

Das Kapitel „Definition von Code-Tabellen“
erläutert dem Systembetreuer, wie er eigene Code-Tabellen erstellen und vorhandene Code-Tabellen modifizieren kann.

Das Kapitel „Einsatzvorbereitung“
gibt Hinweise für die Einsatzvorbereitung.

Das Kapitel „Tabellen“
enthält alle wichtigen Tabellen wie z.B. unterstützte Leitungscodes und BS2000-EBCDI-Codes.

Daran anschließend finden Sie die Verzeichnisse für Fachwörter, Abkürzungen, Literatur und Stichwörter.

1.3 Änderungen gegenüber dem Vorgängerhandbuch V1.3

Dieses Handbuch enthält folgende Funktionserweiterungen und Änderungen von XHCS V2.0 gegenüber V1.3:

Kapitel		ab Version
2.3	XHCS unterstützt den Unicode-Zeichensatz:	V2.0
3.3	• Erweiterung des Moduls GNLMTAB wegen Unicode.	
3.3.1	• Unterstützung eines 8-bit-Terminals oder der Terminal-Emulation MT9750 im Unicode-Modus	
4.1	• Die Programmschnittstelle NLSCOD kann Umsetztabellen zwischen 8-bit-Codes und Unicode ausgeben.	
4.1	• Die Programmschnittstelle NLSCOD zur Ausgabe von Klein-/Großschreibungs-, Eigenschafts- und Sortiertabellen wurde um Unicode-Tabellen erweitert.	
4.1	• Die Programmschnittstelle NLSCOD kann eine Tabelle mit den Byte-Eigenschaften für UTFE ausgeben.	
4.2	• Die Programmschnittstelle NLSCMP liefert Informationen zum Ausgabecode.	
4.2 / 4.3	• An den Schnittstellen NLSCMP und NLSCNV können Sie Ein- und Ausgabezeichenketten auch über eine Adresse und eine Länge angeben und empfangen.	
4.3	• Die Programmschnittstelle NLSCNV bietet für die Umsetzung von Zeichenstrings die Möglichkeit, als Quell- oder Zielcode eine Unicode-Variante anzugeben.	
4.3	• Die Programmschnittstelle NLSCNV kann über eine Call-Schnittstelle aus TU SVC-frei aufgerufen werden.	
4.3	• Alle 7-/8-bit-Zeichensätze können eindeutig umkehrbar konvertiert werden.	
4.3	• Bei NLSCNV wird als Ersatzzeichen nicht mehr das NIL-Zeichen sondern C'.' (U+002E) eingesetzt.	
5.6	• XHCS unterstützt die Definition zusätzlicher Unicode-Zeichen.	
5.7	Mit dem BS2000-Kommando /ADD-CODE-TABLES können Sie Code-Tabellen dynamisch im laufenden Betrieb ändern und erweitern.	V1.4
7.3.1.1	XHCS unterstützt die ISO-Code-Variante 8859-4.	V2.0
7.3.1.1	XHCS unterstützt die ISO-Code-Variante 8859-3.	V2.0

1.4 Verwendete Metasprache

Um eine möglichst einfache Handhabung zu bieten, sind in dieser Beschreibung Zeichen als so genannte Metasymbole verwendet, die bereits weitgehend aus anderen Handbüchern des BS2000/OSD bekannt sind. Sie sind in der folgenden Tabelle erläutert.

Formale Darstellung	**Erläuterung**	**Beispiel**
GROSSBUCH-STABEN	bezeichnen Konstanten, die der Anwender in dieser Form angeben muss.	„YES“
kleinbuchstaben	bezeichnen Variablen, deren Inhalt von Fall zu Fall verschieden sein kann. Der Anwender muss sie bei der Eingabe durch aktuelle Werte ersetzen.	partnername
/	Der Schrägstrich trennt alternative Angaben.	MF=L / S
[ ]	Eckige Klammern schließen Angaben ein, die weggelassen werden können. Steht bei Wahlangaben das Komma innerhalb der Klammer, so muss es nur bei Verwendung dieser Wahlangabe geschrieben werden. Steht es hingegen außerhalb der Klammer, so muss es stets geschrieben werden, auch wenn die Wahlangabe nicht gemacht wird.	[kennwort4] dateiname[,ERASE]
unterstrichen	Standardwerte sind unterstrichen dargestellt. Das sind die Werte, die das System einsetzt, wenn der Anwender keine Angaben macht.	INFO=*LIST-OF-CCS
...	Punkte bedeuten eine Wiederholung. Sie zeigen an, dass die davor stehende Einheit mehrmals wiederholt werden kann.	(archivnr,...)
()	Ein Ausdruck, der zur Darstellung von Variablen benutzt wird, steht in runden Klammern. Diese Darstellung soll auf einen Blick den Wertebereich zeigen. Da dazu mehrere Zeichen notwendig sind, ist auch diese formale Eingrenzung erforderlich.	(0 < länge < 9)
	für Hinweise auf besonders wichtige Informationen	

1.5 Readme-Datei

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte gegebenenfalls der produktspezifischen Readme-Datei. Sie finden die Readme-Datei auf Ihrem BS2000-Rechner unter dem Dateinamen SYSRME.XHCS-SYS.020.D.

Die Benutzerkennung, unter der sich die Readme-Datei gegebenenfalls befindet, erfragen Sie bitte bei Ihrer zuständigen Systembetreuung. Die Readme-Datei können Sie mit dem Kommando /SHOW-FILE oder mit einem Editor ansehen oder auf einem Standarddrucker mit folgendem Kommando ausdrucken:

```
/PRINT-DOCUMENT dateiname, LINE-SPACING=*BY-EBCDIC-CONTROL
```

bei SPOOL -Versionen kleiner 3.0A:

```
/PRINT-FILE FILE-NAME=dateiname,LAYOUT-CONTROL=
    PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```

2 Einführung in XHCS

Das Softwareprodukt XHCS (E**x**tended **H**ost **C**ode **S**upport) ist die zentrale Informationsquelle über die codierten Zeichensätze (CCS **C**oded **C**haracter **S**et), die im BS2000/OSD zur Verfügung stehen. Damit müssen die verschiedenen Programme Informationen über Zeichensätze nicht fest speichern, sie erhalten diese Informationen von XHCS.

XHCS identifiziert die Datencodes, egal woher sie kommen, ob von einer Terminaleingabe, von einer Programmausgabe oder von einem anderen System. Zur Identifikation der übertragenen Datencodes dient der Zeichensatzname (CCSN **C**oded **C**haracter **S**et **N**ame). Die codierten Zeichensätze stellt XHCS in Form von Tabellen zur Verfügung. Je nach anwenderspezifischen Erfordernissen können vorhandene Zeichensätze an die lokalen Erfordernisse angepasst werden und eigene Zeichensätze zu den bereits vorhandenen hinzugefügt werden.

Mit XHCS können Zeichen von erweiterten Zeichensätzen schnell, einfach und konsistent zwischen Rechner und Peripherie übertragen werden. Wegen der Erweiterung der zur Verfügung stehenden Zeichen von 95 (im 7-bit-Code) auf 189 Zeichen (im 8-bit-Code) pro Zeichensatz, spricht man auch von „erweiterten Zeichensätzen“.

Zusätzlich unterstützt XHCS für die unterstützten 8-bit-ISO8859-Codes den Unicode-Zeichensatz und darüber hinaus etwa 70 Zeichen des Unicode-Zeichensatzes, die gemäß der internationalen Richtlinie im Meldungswesen verwendet werden. Außerdem bietet XHCS die Möglichkeit, den Zeichenvorrat für die Unicode-Unterstützung zu erweitern.

2.1 Erweiterte Zeichensätze

Erweiterte Zeichensätze werden auch „8-bit-Codes“ oder erweiterte Codes genannt, weil die Anzahl der zur Verfügung stehenden Codeplätze verdoppelt wurde. Diese Anzahl der Codeplätze erlaubt es, Zeichen für verschiedene Sprachen zu definieren, die aber trotzdem zu dem ursprünglich verwendeten Code, dem ASCII-Code (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange), kompatibel sind. Die verschiedenen 8-bit-Codes sind in der internationalen Norm ISO 8859 definiert. Sie haben alle in der „linken“ (niederwertigen) Hälfte der Codetabelle einen gemeinsamen Teil, analog zu ASCII, in der „rechten“ (höherwertigen) Hälfte unterscheiden sie sich. Einzelne Codes werden zu Gruppen kompatibler Codes zusammengefasst, die über ihre ISO-Code-Variantennummer identifiziert werden. Folgende Codes sind zurzeit in ISO 8859 als Standard definiert:

- 8859-1 Latin-1 (West- und Nord-Europa)
- 8859-2 Latin-2 (Ost-Europa, ausgenommen Türkei und die Baltischen Staaten)
- 8859-3 Latin-3 (Mittelmeerraum und Süd-Afrika)
- 8859-4 Latin-4 (Skandinavien und die Baltischen Staaten)
- 8859-5 Kyrillisch
- 8859-6 Arabisch
- 8859-7 Griechisch
- 8859-8 Hebräisch
- 8859-9 Latin-5 (Türkei, West-Europa inklusive Skandinavien)
- 8859-10 Latin-6 (Nord-Europa und die Baltischen Staaten)
- 8859-15 Latin-9 (West- und Nord-Europa mit EURO-Zeichen €)

Eine Übersicht über alle Codes, die XHCS unterstützt und deren Zuordnung zu den EBCDI-Codes finden Sie im Kapitel „Tabellen“ auf Seite 145.

Auf der BS2000-Seite wird der Zeichensatz ISO 8859-1 durch den EBCDIC.DF.04-1 dargestellt. Der EBCDIC (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchange **C**ode), den BS2000 verwendet, muss so erweitert werden, dass jedes Zeichen ein Gegenstück im entsprechenden ISO 8859-x hat. Da EBCDI-Codes nicht standardisiert sind, existieren unterschiedliche Zuordnungen zwischen EBCDI- und ISO-Codes. Das kyrillische Alphabet zum Beispiel, das im Teil 5 des ISO 8859 Standards (ISO 8859-5) definiert ist, hat zwei verschiedene äquivalente Rechnercodes: EBCDIC.DF.04-5 und EBCDIC.EHC.LC.

Die definierte Zuordnung zwischen ISO 8859-1 und EBCDIC.DF.04-1 spielt eine besondere Rolle. Für jede ISO-Tabelle wird eine EBCDIC-Tabelle definiert, die jeweils den gleichen Zeichenvorrat enthält, aber mit einer anderen Code-Belegung. Diese EBCDIC-Tabellen haben die Bezeichnung EDF041 bis EDF049 und EDF04F. Sie wurden so gewählt, dass mit ein und derselben Umsetztabelle die Konvertierung von ISO 8859-n (n=1..9 oder n=15) nach EDF04n (n=1..9 oder n=F) und umgekehrt möglich ist. Neben der Nutzung vorhandener erweiterter Zeichensätze ermöglicht XHCS dem Systembetreuer, die Codierung erweiterter Zeichensätze selbst festzulegen, einen erweiterten Zeichensatz als Systemstandard auszuwählen und jedem Anwender einen Anwender-Standard-Code zuzuordnen.

2.2 Internationalisierung

Heute werden auf der Welt mehr als 2000 Sprachen gesprochen. Viele dieser Sprachen sind nur in einem bestimmten Land gebräuchlich, andere Sprachen wiederum sind in mehreren Ländern verbreitet, und darüber hinaus gibt es nicht wenige Länder, in denen mehr als eine Sprache gesprochen wird.

Um Softwareprodukte international einsetzen zu können, dürfen sie nicht nur die am Einsatzort gesprochene Sprache bzw. die in der dazugehörigen Schrift verwendeten Zeichen berücksichtigen, in mehrsprachigen Ländern müssen sie auch mehrere Sprachen unterstützen. Um die Geschäftsverbindungen mit anderen Ländern zu ermöglichen, müssen sie auch in der Lage sein, die Schriften dieser Länder zu unterstützen.

Mit XHCS ist es möglich, Schriften aller Landessprachen im BS2000 darzustellen, die in den internationalen Code-Tabellen nach ISO 8859 definiert sind. Das sind die Zeichensätze aller west- und osteuropäischen Sprachen einschließlich Kyrillisch, Griechisch und den baltischen Sprachen, sowie zusätzlich Arabisch, Persisch (Farsi) und Maltesisch.

Für die in ISO 8859 definierten Standards genügt die erweiterte 8-bit-Codierung für die Zeichendarstellung der verschiedenen Zeichensätze. Damit ist allerdings nur eine so genannte Einbyte-Code-Verarbeitung für Buchstabenschriften möglich. Für ideographische oder Begriffsschriften, wie z.B. die verschiedenen asiatischen Schriften, genügt eine Einbyte-Code-Verarbeitung nicht mehr. Hier ist eine Mehrbyte-Code-Verarbeitung erforderlich. Dafür gibt es in den jeweiligen Ländern (China, Taiwan, Süd-Korea, Japan) nationale Normen.

2.3 Unicode-Unterstützung

XHCS V2.0 unterstützt folgende Unicode-Varianten:

- UTF-16
  ist weitestgehend identisch mit der Zwei-Bytes-Unicode-Darstellung UCS-2. Die sogenannten Surrogate Pairs (die eine Zwei-Bytes-Darstellung von Zeichen zwischen x'FFFF' und x'10FFFF' ermöglichen) werden nicht unterstützt.
  XHCS-Schreibweise: UTF16 oder 'UNICODE'.
- UTF-8
  ist die am meisten verbreitete Unicode-Variante.
  XHCS-Schreibweise: UTF8.
- UTF-EBCDIC
  ist eine spezielle Realisierung der ursprünglich für IBM entwickelten Unicode-Variante (siehe http://www.unicode.org/reports/tr16/). Anders als diese verwendet XHCS für die Abbildung nach EBCDIC die Tabelle EDF041 (siehe Seite 151).
  XHCS-Schreibweise: UTFE.

Wenn im vorliegenden Handbuch von Unicode die Rede ist, wird darunter immer der Zwei-Bytes-Code UTF-16 verstanden. Unter Unicode-Variante kann sowohl UTF-16 als auch UTF-8 oder UTF-EBCDIC verstanden werden.

Nähere Informationen zu den verschiedenen Unicode-Varianten entnehmen Sie dem Übersichtshandbuch „Unicode im BS2000/OSD“ [32].

XHCS V2.0 umfasst Umsetztabellen der unterstützten 8-bit-ISO-Codes (ISO8859-1/2/3/4/5/7/9/15) nach Unicode und deren Umkehrabbildung.

Über die Zeichen hinaus, die in diesen ISO-Codes enthalten sind, unterstützt XHCS etwa 70 Zeichen des Unicode-Zeichensatzes, die bei unseren Kunden verwendet werden (siehe auch Abschnitt „Zusätzliche Unicode-Zeichen definieren“ auf Seite 132). Das bedeutet, dass nur die diesen Zeichen entsprechenden Unicode-Positionen in den Umsetztabellen definiert sind. Für weitere Zeichen wurde ein Pseudo 8-bit-ISO-Code definiert, siehe auch Abschnitt „Pseudo-8-bit-Code definieren“ auf Seite 132.

Die Klein-/Großschreibungstabelle, die Eigenschaftstabelle, und die Sortiertabelle für Unicode ist nur für Zeichen aus den unterstützten ISO-Codes und zusätzlicher Zeichen definiert.

Zusätzlich haben Sie die Möglichkeit den Zeichenvorrat für die Unicode-Unterstützung zu erweitern: Definieren Sie dazu die entsprechenden Code-Tabellen, die diese neuen Zeichen beinhalten (siehe Seite 132).

Normalisierung

Ein spezielles Feature der Unicode-Variante UTF-16 ist die Normalisierung von Unicode-Zeichenketten (siehe auch http://www.unicode.org/reports/tr15/).

XHCS unterstützt folgenden leicht eingeschränkten Funktionsumfang:

– Komposition
  Ein Grundzeichen wird mit einem oder zwei nachfolgenden diakritischen Zeichen zu einem einzigen Zeichen (falls definiert) zusammengefügt.
  Makroaufruf: NLSCNV ACTION=COMPOSE,….

– Dekomposition
  Ein zusammengesetztes Zeichen wird in ein Grundzeichen und ein oder zwei nachfolgende diakritische Zeichen (falls definiert) zerlegt.
  Makroaufruf: NLSCNV ACTION=DECOMPOSE,….

Grundlage der Normalisierung ist die Normalisierungstabelle, die aus der Unicode-Sortiertabelle abgeleitet wird und für eine performante Informationsbeschaffung sorgt.

Weitere Details der Normalisierung entnehmen Sie dem Übersichtshandbuch „Unicode im BS2000/OSD“ [32].

3 XHCS

XHCS realisiert das Konzept der zentralen Zeichenbehandlung in BS2000/OSD. Es lässt unterschiedliche Zeichensätze zu, und es stellt allen zeichenverarbeitenden Komponenten Mechanismen zur Verfügung, um aktuelle Zeichensätze zu erkennen und zu interpretieren.

Das XHCS-Konzept umfasst

- erweiterte Endgeräteprotokolle

  Die Datenstationen verfügen nicht nur über alle europäischen Schriften; über ein erweitertes Endgeräteprotokoll teilen sie dem System den aktuell eingestellten Zeichensatz mit und sind umgekehrt in der Lage, diesen auf Anforderung dynamisch zu wechseln. Wenn die Systemvoraussetzungen es erfordern, schalten sie sich in den Modus einer herkömmlichen Datenstation zurück.

- erweiterte BS2000/OSD-Systemkomponenten

  Soweit sie zeichenbezogene Operationen durchführen, wie z.B. sortieren oder umcodieren, sind sie in der Lage, die aktuelle Codierung der Daten und die Einstellung der Datenstation zu identifizieren und ihre Verarbeitung darauf abzustimmen. Dabei bedienen sie sich der zentralen Hilfe von XHCS.

- das Subsystem XHCS

  XHCS liefert alle Informationen zu allen Zeichensätzen, die für Vergleichs- und Umwandlungsoperationen erforderlich sind. Damit müssen die übrigen Komponenten entsprechende Tabellen nicht mehr selbst führen. Die XHCS-Schnittstellen stehen auch jedem Anwendungsprogramm zur Verfügung.

Das folgende Bild zeigt XHCS in seiner Software-Umgebung.

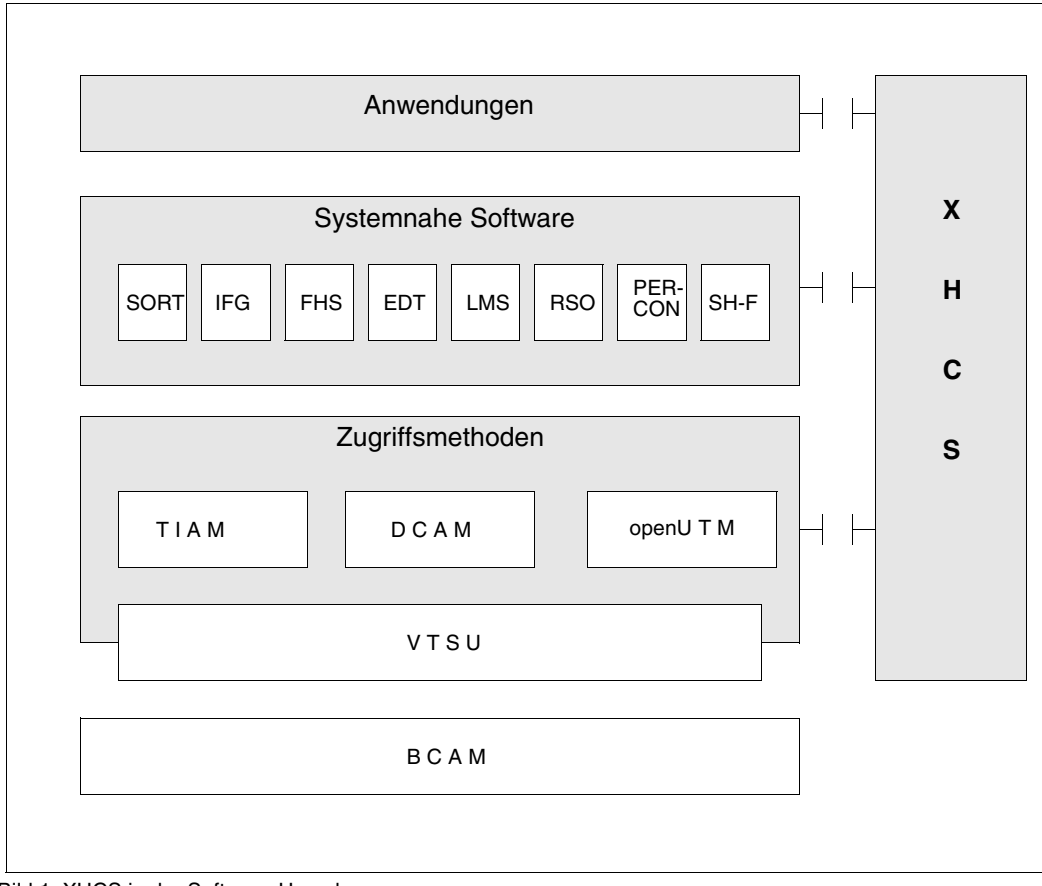


Bild 1: XHCS in der Software-Umgebung

Die Zugriffsmethoden TIAM, DCAM und openUTM sind über die Softwarekomponente VTSU mit XHCS verbunden. Der Anwender kann mithilfe der entsprechenden TIAM-, DCAM- oder openUTM-Anwendungsprogramme auf XHCS zugreifen. openUTM-Anwender können XHCS jedoch nur über FHS nutzen. DCAM- und TIAM-Anwender können XHCS 'direkt' (über den VTSUCB) oder ebenfalls über FHS nutzen.

Das Produkt XHCS ist die Informationsquelle über die codierten Zeichensätze, die im System zur Verfügung stehen. XHCS bietet Programmschnittstellen für die folgenden Funktionen:

– Versorgung und Bereitstellung verschiedener Tabellen eines vorgegebenen Codes (Umsetzung in einen anderen Code, Umsetzung von Klein- in Großbuchstaben, Tabelle der Sortierungsgewichte und verschiedene Zeicheneigenschaften)
– direkte Umwandlung von Zeichenketten
– Informationen liefern über die im System existierenden Codes und die Möglichkeiten der Umsetzung

Mit diesen Schnittstellen können Anwendungen unabhängig vom vorhandenen Code betrieben werden. XHCS bietet einen Mechanismus zur Internationalisierung der Software, soweit es die Zeichensätze betrifft.

Die Codes in XHCS sind definiert mithilfe von angepassten Tabellen. Sie können codierte Zeichensätze zu den vordefinierten hinzufügen und sie an die lokalen Erfordernisse anpassen, z.B. die Sortiersequenz der Zeichen eines Codes ändern.

Die Codes sind logisch in Gruppen kompatibler Codes zusammengefasst, je nach enthaltenem Zeichensatz. Die Code-Gruppen sind gekennzeichnet durch ihre ISO-Variantennummer, diese Nummer bezieht sich auf eine der Varianten der ISO 8859-Norm. Ausgenommen ist die Gruppe der Unicode-Varianten: Für sie wurde eine Pseudo-Variantennummer eingeführt, siehe auch Abschnitt „Code-Information: NLSCOD“ auf Seite 66. Alle 7-/8-bit-Codes gelten als partiell kompatibel zu Unicode.

Alle Codes einer vorhandenen Variante enthalten den gleichen Zeichensatz wie der entsprechende ISO 8859-Zeichensatz oder wie eine Untermenge.

Umsetzungen können nur zwischen Codes der gleichen Gruppe erfolgen, da XHCS entsprechende Zeichen einer anderen Gruppe nicht kennt.

3.1 Vorteile von XHCS

XHCS bietet folgende Vorteile:

- Vollständige und eindeutige Darstellung der Zeichen der Landessprache
  Durch die Verdoppelung der nutzbaren Code-Plätze ermöglicht XHCS die eindeutige Darstellung aller nationaler und internationaler Zeichen, die in ISO 8859 definiert sind.
- Darstellung fremdsprachiger Daten
  Fremdsprachige Texte, Namen und Adressen können eindeutig und korrekt wiedergegeben werden.
- Datenaustausch mit Partnersystemen
  Systemübergreifende Kommunikationsprodukte, wie z.B. Emulation oder File Transfer, nehmen Umcodierungen an den Systemgrenzen vor, um eine korrekte Weiterverarbeitung im Zielsystem zu ermöglichen.
- Unterstützung mehrsprachiger Anwendungen
  Mit XHCS können Anwendungen leicht in andere Sprachen bzw. Ländervarianten portiert werden und es können unterschiedliche Sprachen und Datenbestände simultan bearbeitet werden.
- Individuelles Umstellungskonzept
  XHCS gewährleistet volle Integrität der vorhandenen Hard- und Software-Konfiguration. Es kann festgelegt werden, ob die vorhandenen Daten mit neuen Daten gemischt werden sollen oder ob sie getrennt gehalten werden sollen und ob die vorhandenen Anwendungen zur Verarbeitung der neuen Daten eingesetzt werden sollen bzw. wann sie dafür angepasst werden sollen.

3.2 Voraussetzungen für den Einsatz von XHCS

Dieser Abschnitt beschreibt die Hard- und Software-Voraussetzungen für den Einsatz von XHCS, Hinweise zur Installation sowie die Unterstützung spezieller Terminals.

3.2.1 Hardware-Umgebung

XHCS unterstützt folgende 8-bit-fähigen Datenstationen:

Terminals

- 9756-National (Europäisch, arabisch, persisch)
- 9758-M486
- 9759-M2/M4
- 9763-M/C/G
- EMDS V5.1 Emulation (SINIX)
- MT9750 V7.0C Emulation (MS-Windows)
- DOORS-Emulation Winsock V3.1A
- WT9750 V2.0A

Diese Terminals können den verwendeten Zeichensatz mithilfe eines erweiterten Terminalprotokolls zum System übertragen und den Zeichensatz dynamisch ändern.

Um im 8-bit-Modus zu arbeiten, muss im PDN das Terminal 9756-National mit dem Typ DSS-9755 eingetragen werden, die anderen Terminals mit dem Typ DSS-9763.

Für EMDS müssen die entsprechenden Umgebungsvariablen richtig gesetzt werden (siehe Handbuch „EMDS (SINIX)“ [10]).

Drucker

- 9001-32
- 9011-28/29
- 9012
- 9013-31x
- 9014
- 9097
- 9021
- 4819/20

Alle Drucker müssen im PDN mit ihrem realen Namen eingetragen werden.

3.2.2 Software-Umgebung

XHCS

XHCS wird ausgeliefert mit vordefinierten Codes (Objekt-Module), die direkt verwendet werden können. Diese Codes und verschiedene andere stehen auch als Source-Code (Assembler-Makros) zur Verfügung. Damit kann der Systembetreuer eigene XHCS-Tabellenmodule erstellen.

BS2000/OSD

XHCS unterstützt folgende Funktionen:

– allen Dateien und Bibliothekselementen ist ein CCSN fest zugeordnet

– jedem Anwender kann ein Standard-Benutzer-CCSN zugeordnet werden. Dadurch erhält der CCS die Information, den 8-bit-Modus für Ein-/Ausgaben zu verwenden.

– VTSU-Sonderroutinen

VTSU

Die Systemsoftware für die Unterstützung virtueller Terminals (VTSU) ist Bestandteil von *open*Net Server (früher DCAM). Die Angaben in dieser Beschreibung gelten für VTSU ab V13.2A.

Die Unterstützung erweiterter Codes ist eine Standardfunktion von VTSU. Für folgende spezielle Fälle ist der Einsatz von VTSU-Sonderroutinen (siehe Seite 36) erforderlich:

– Unterstützung von 7-bit-Codes für nationale 7-bit-Datenstationen

– Unterstützung der 8-bit-Schriften Arabisch, Nord-Afrikanisch (Arabisch/Französisch) und Farsi (Persisch)

– Unterstützung von Unicode

– Unterstützung der ESC-Drucker 9001, 9013 und 9022

TIAM

Ab TIAM V13.1C wird Unicode unterstützt.
Mit dem BS2000-Kommando SHOW-TERMINAL-ATTRIBUTES INF=*ALL wird die Information, die VTSU von einer Datenstation oder einer Emulation erhalten hat, ausgegeben. Anhand dieser Information kann man feststellen, ob die Datenstation Unicode-fähig ist.

PDN

Für die volle Unterstützung von VTSU ab V11.0 ist PDN ab V11.0A erforderlich.

Generell hat PDN an Bedeutung verloren. Die Beschreibungen und Darstellungen zu PDN wurden in diesem Handbuch jedoch noch zum besseren allgemeinen Verständnis beibehalten.

3.2.3 Hinweise zur Installation

Das optionale Produkt XHCS ist ein dynamisches Subsystem, das über DSSM (Dynamic Subsystem Manager) geladen wird und nicht entladen werden kann. Deshalb muss das Subsystem XHCS-SYS im Subsystemkatalog deklariert werden (siehe Handbuch „BS2000/OSD-BC Systeminstallation“ [3]). Die Modul-Bibliothek SYSLNK.XHCS-SYS.*xxx* (*xxx*=Versionsnummer) und die REP-Dateien SYSREP.XHCS-SYS.*xxx* (*xxx*=Versionsnummer) sind standardmäßig unter TSOS zu katalogisieren oder mit IMON zu installieren.

Die Datei SYSSSC.XHCS-SYS.*xxx* (*xxx*=Versionsnummer) enthält die Subsystemdeklarationen (SSD-OBJ Datei). Die Subsystemdeklarationen erwarten standardmäßig die Dateien unter TSOS oder dort, wo sie mit IMON installiert wurden.

3.3 Komponenten und Programmschnittstellen

XHCS ist als dynamisches Subsystem definiert (Subsystemname ist XHCS-SYS). Es wird von DSSM (Dynamic Subsystem Manager) verwaltet und beim Systemstart geladen. XHCS ist für privilegierte und für nicht privilegierte Programme verfügbar. Es basiert auf Tabellen, wobei alle Tabellen Codes beschreiben. Jedem zu beschreibenden Code sind mehrere Tabellen zugeordnet. Diese Tabellen werden durch Makros aufgebaut und modifiziert. Sowohl Betriebssystem wie auch Anwendungsprogramme können auf diese Tabellen zugreifen. Die Code-Tabellen sind in dem Modul GNLMTAB enthalten. Sollen die mitgelieferten Standardcodes des Moduls GNLMTAB nicht verwendet werden, so muss das entsprechend modifizierte Modul vor dem Laden des Subsystems einmal assembliert und in die Modulbibliothek SYSLNK.XHCS-SYS.020 aufgenommen werden. Ein gebrauchsfertiges Standard-Modul GNLMTAB wird mit XHCS ausgeliefert und erlaubt es Ihnen, mit folgenden ISO 8859-Codes zu arbeiten:

- Norm 1: EDF041, ISO88591, EDF04DRV, EDF03IRV, ISO646
- Norm 2: EDF042, EEHCL2, ISO88592
- Norm 3: EDF043, ISO88593
- Norm 4: EDF044, ISO88594
- Norm 5: EDF045, EEHCLC, ISO88595, EEHCLC1
- Norm 7: EDF047, EEHCLG, ISO88597
- Norm 9: EDF049, ISO88599
- Norm F: EDF04F, ISO8859F, WCP1252P

Der Gruppen-Referenzcode (siehe Abschnitt „Gruppierung kompatibler Codes“ auf Seite 116) steht bei jeder Gruppe an erster Stelle. EDF03IRV ist Systemstandard.

Der Quellcode von GNLMTAB ist in der Bibliothek SYSSRC.XHCS-SYS.020.GNLMTAB enthalten. Er enthält zusätzlich als Kommentare die notwendigen Code-Zeilen, die die restlichen im Kapitel „Tabellen“ auf Seite 145 aufgelisteten Zeichensätze definieren. Wollen Sie einen oder mehrere dieser Zeichensätze nutzen, muss der Systembetreuer die Kommentarzeichen der entsprechenden Code-Zeilen löschen. Danach muss der Quellcode neu assembliert werden. Zusätzlich kann der Systembetreuer das Modul GNLMTAB selbst erstellen. Dazu wird der Makro NLSCTAB oder die Makros NLSHEAD und NLSCCS benötigt (siehe Seite 122ff).

Ab XHCS V2.0 sind im Modul GNLMTAB zusätzlich enthalten:

– die Unicode-Sortierinformationen

– die Unicode-Eigenschaftstabelle

– die Tabelle mit den Zuordnungen von Großbuchstaben zu den entsprechenden Kleinbuchstaben und umgekehrt für Unicode

– die Tabelle zur Umwandlung von Großbuchstaben in die entsprechenden Kleinbuchstaben für die UTFE-Einbyte-Codierungen

– die Tabelle zur Umwandlung von Kleinbuchstaben in die entsprechenden Großbuchstaben für die UTFE-Einbyte-Codierungen

– die Tabelle mit den Byte-Eigenschaften für UTFE

3.3.1 XHCS-Unterstützung durch VTSU

Die Softwarekomponente VTSU realisiert eine logische Zeilen-/Seiten-Datenstation. VTSU erlaubt es Anwendern der Zugriffsmethoden DCAM, TIAM und openUTM, für die Aufbereitung von Nachrichten logische, d.h. symbolische Steuerzeichen zu verwenden, statt der gerätespezifischen realen Steuerzeichen. Alle logischen Steuerzeichen sind zusammen mit ihren symbolischen Namen in einer Datenstruktur enthalten, die Sie bei Bedarf in das Anwendungsprogramm kopieren können. VTSU wandelt diese logischen Steuerzeichen in die entsprechenden realen Steuerzeichen um, die die jeweilige Datenstation verlangt. Auf diese Weise macht VTSU Anwendungsprogramme unabhängig vom Typ der realen Datenstationen.

Zur Unterstützung von XHCS unterscheidet VTSU:

– 8-bit-Datenstationen im 8-bit-Modus, die entweder einen 7-bit-Code mit den Steuerzeichen SI/SO oder einen 8-bit-Code senden

– 8-bit-Datenstationen im 7-bit-Modus, die einen 7-bit-Code senden

– 8-bit-Datenstationen im Unicode-Modus, die einen Unicode sendet

– 7-bit-Datenstationen, die einen 7-bit-Code senden

Der Nachrichtenkopf ist dabei immer in einem ISO-Code codiert.

8-bit-Terminal im 8-bit-Modus

Zur Unterstützung von 8-bit-Terminals benötigt VTSU Informationen über den Typ der Datenstation, welche ISO-Code-Varianten vom System unterstützt werden, ob ein Code-Erweiterungsmechanismus (SI/SO) benutzt wird und ob die benutzten Codes kompatibel zu den von dem Terminal unterstützten Codes sind. Diese Informationen erhält VTSU über eine Statusabfrage und eine Kompatibilitätsprüfung. Der Status wird bei TIAM- und openUTM-Anwendungen zum Zeitpunkt des Verbindungsaufbaus automatisch abgefragt. Bei DCAM-Anwendungen müssen Sie selbst den Status (OPTCD = TERMSTAT beim YOPNCON-Aufruf) abfragen.

Über die gelieferten Informationen der Statusabfrage erkennt VTSU, ob der Rechner mit 7-bit-Terminals oder mit 8-bit-Terminals verbunden ist. Wird der Status nicht abgefragt, wird der erweiterte Standard-Code ignoriert. Als Standard-Code gilt dann EDF03IRV.

Die Kompatibilität wird bei TIAM-Anwendungen bei der LOGON-Verarbeitung, bei DCAM und openUTM-Anwendungen beim Verbindungsaufbau überprüft. Wird der angegebene Code nicht von der Datenstation unterstützt, wird der Anwender-Standard-Code ignoriert und ebenfalls im 7-bit-Code gearbeitet. TIAM-Anwendungen geben in diesem Fall eine Warnung aus.

Erst nachdem VTSU festgestellt hat, welcher Datenstationstyp und welcher Code benutzt wird, wird XHCS aufgerufen. XHCS liefert die benötigten Umwandlungstabellen, um entweder den spezifizierten Code umzuwandeln, oder in den spezifizierten Code umzuwandeln.

– Wird bei einer DCAM-Anwendung der Status nicht abgefragt, wird das Terminal 9758 als Terminal des Typs 9763 angesehen. Dies führt im Format-Modus (FHS, LINE-Modus, Extended-Line-Modus) zu Fehlern. Wird der Status abgefragt, wird das Terminal 9758 als Terminal des Typs 9755 angesehen, auch wenn das Terminal 9758 im PDN als Terminal 9763 generiert ist. Gemäß der PDN-Generierung des Terminals 9758 können folgende Situationen auftreten:

Definition der Datenstation 9758 im PDN	**Reaktion von FHS**
Das Terminal 9758 ist im PDN als Terminal des Typs 9763 generiert. Der Status wird nicht abgefragt.	Formatierung nicht erfolgreich; die FHS-Formatierung erfolgt für eine 9763 anstatt für eine 9755.
Das Terminal 9758 ist im PDN als Terminal des Typs 9763 generiert. Der Status wird abgefragt.	Formatierung erfolgreich; die FHS-Formatierung findet für eine 9755 statt.
Das Terminal 9758 ist im PDN als Terminal des Typs 9755 oder 9758 generiert.	Formatierung erfolgreich; die FHS-Formatierung findet ohne Probleme statt. Das Terminal 9758 ist jedoch nicht im 8-bit-Modus nutzbar.

– Im 8-bit-Modus kann der ICE-Zeichensatz nicht benutzt werden.

8-bit-Terminal im Unicode-Modus

VTSU unterstützt ab VTSUB V13.2A Unicode. Wenn Unicode-Daten empfangen oder gesendet werden sollen, muss dies VTSU mitgeteilt werden. Dies kann in den Anwendungen über den VTSU-B-Kontrollblock oder über das BS2000-Kommando /MODIFY-TERMINAL-OPTIONS erfolgen. Die Einstellung im VTSU-B-Kontrollblock hat Vorrang vor dem mit /MODIFY-TERMINAL-OPTIONS eingestellten Wert. Er gilt jedoch nur für den aktuellen Aufruf.
Um mit der Terminal-Emulation MT9750 V7.0 im Unicode-Modus zu arbeiten, müssen die Sitzungsparameter der Emulation entsprechend konfiguriert werden (siehe auch Abschnitt „Terminal-Emulation MT9750 im Unicode-Modus“ auf Seite 38) .

7-bit-Terminal/8-bit-Terminal im 7-bit-Modus

7-bit-Terminals und 8-bit-Terminals im 7-bit-Modus haben die gleiche Code-Behandlung, das heißt, wenn eines dieser Terminals angeschlossen ist, wandelt ausschließlich die Netzsoftware PDN die Codes um. Die Steuerzeichen SI, SO und die ISO-Code-Varianten werden von VTSU ignoriert.
Anwender und Systembetreuer müssen selbst dafür sorgen, dass der erweiterte Standard-Code zur Tastaturvariante passt.

8-bit-Drucker

7-bit-Drucker und 8-bit-Drucker haben die gleiche PDN-Generierung und VTSU kann auch über eine Statusabfrage nicht erkennen, mit welchem Druckertyp gearbeitet wird. Deswegen wird der Druckertyp (7-bit/8-bit) und der Verbindungstyp (7-bit/8-bit) durch Betriebsparameter (VTSU) oder den Freitextparameter (PDN) festgelegt. Jeder Drucker hat seine eigenen Betriebsparameter, die den Verbindungstyp und den Druckertyp beschreiben.
Eine Beschreibung, wie Sie die Betriebsparameter einstellen, finden Sie im Handbuch „VTSU“ [1].

Zur Code-Umwandlung muss VTSU die ESCAPE-Sequenzen zur Druckersteuerung aus der Nachricht herausnehmen. Damit VTSU diese Sequenzen erkennt, müssen sie im Standard EBCDIC.DF.03 codiert sein. Im physikalischen Modus (MODE=PHYS) können Sie jedoch ’spezielle’ ESCAPE-Sequenzen benutzen, die nicht zum EBCDIC-Kern gehören. VTSU erkennt diese speziellen ESCAPE-Sequenzen nicht und versucht, sie gemäß dem festgelegten Zeichensatz umzuwandeln. Dies kann zu Inkonsistenzen bei der Code-Umwandlung führen. Um in diesem Fall eine Code-Umwandlung durch VTSU zu unterdrücken, wird im VTSUCB der Parameter CODETR=NO angeboten.

Bei der Code-Umwandlung nimmt VTSU automatisch an, dass der eingestellte Zeichensatz des Druckers gleich dem geforderten 8-bit-Zeichensatz ist.

3.3.1.1 VTSU-Control-Block

Mithilfe des VTSUCB (VTSU-Control-Block) können Sie spezielle VTSU-Parameter zur Nachrichtenaufbereitung für die Ein- und Ausgabe einstellen, unabhängig von der Zugriffsmethode, jedoch nicht für openUTM-Anwendungen. Ebenso haben Sie die Möglichkeit die XHCS-Funktionalität zu nutzen. Der VTSUCB wird bei den Ein- und Ausgabeoperationen der entsprechenden Zugriffsmethoden an VTSU übergeben.

Im VTSUCB können Sie mit dem Parameter CCSNAME den Namen des zu verwendenden Zeichensatzes festlegen. Die möglichen Werte von CCSNAME sind entweder ein CCSN oder der Wert *EXTEND, der den erweiterten Standard-Anwenderzeichensatz bezeichnet. Der Code muss ein erweiterter EBCDIC sein. Geben Sie *EXTEND ein, benutzt VTSU für die aktuelle Verarbeitung den erweiterten Standard-Anwenderzeichensatz. Den Namen des Zeichensatzes können Sie über die DCSTA-Schnittstelle ermitteln. Geben Sie weder *EXTEND noch explizit den erweiterten Standard-Anwender-Code-Namen ein, bleibt die Datenstation im 7-bit-Modus, außer Sie haben sie selbst in den 8-bit-Modus gesetzt (z.B. bei MODE=PHYS oder über das TIAM-Kommando MODIFY-TERMINAL-OPTIONS).

Über den Makroaufruf VTSUCB können Sie auch den Unicode-Modus einstellen. Es kann jedoch nur die Unicode-Variante UTFE angegeben werden.

Der spezifizierte Code wird nur für den aktuellen Aufruf benutzt. Das bedeutet, nur für diesen Aufruf werden die Steuerzeichen SI/SO interpretiert (Eingabe) und eingefügt (Ausgabe), wenn die Verbindung eine 7-bit-Verbindung ist. Und nur für diesen Aufruf ist eine Umwandlung aus dem spezifizierten Code oder in den spezifizierten Code vorbereitet.

Diese Verarbeitung ist nur bei MODE=MIXED, LINE, EXTEND, INFO, FORM und PHYS jedoch nicht bei MODE=CHIP oder MODE=TRANS möglich.

Für die Zugriffsmethoden DCAM und TIAM gibt es den VTSUCB in den Programmiersprachen ASSEMBLER, FORTRAN, PL/1, C und COBOL.

Bei folgenden Ein-/Ausgabeoperationen kann der VTSUCB zur Bestimmung des Codes angegeben werden:

Zugriffsmethode	Sprache	Ein-/Ausgabeoperationen
TIAM	Assembler	WROUT, RDATA, WRTRD
TIAM	COBOL	WROUT, RDATA, WRTRD (CALL-Aufrufe)
TIAM	FORTRAN	WROUT ,RDATA, WRTRD (CALL-Aufrufe)
TIAM	PL/I	WROUT, RDATA, WRTRD (CALL-Aufrufe)
TIAM	C	WROUT, RDATA, WRTRD (CALL-Aufrufe)
DCAM	Assembler	YSEND, (YRECEIVE), YSENDREC
DCAM	COBOL	YSEND, (YRECEIVE) (CALL-Aufrufe)

3.3.1.2 Terminalstatus

Bevor 8-bit-Ein-/Ausgaben oder Unicode-Ein-/Ausgaben verarbeitet werden können, müssen Sie überprüfen welche Möglichkeiten die Datenstation hat. Die Ein-/Ausgaben können von einer Datenstation oder einer Terminal Emulation kommen.

Mit dem VTSU-Makro DCSTA (TYPE=BASIC) erhalten Sie Informationen über die relevanten Charakteristika der Datenstation und der Verbindung. Die Information wird aktualisiert mit dem Aufruf des TSTAT-Makros (TIAM) bzw. des YINQUIRE-Makros (DCAM).

Die folgenden Felder des DSTA-Makros beziehen sich auf XHCS:

STATTYPE	zeigt an, ob die Datenstation im 8-bit-Modus arbeiten kann
STACCSNN	Anzahl der unterstützten 8-bit/Unicode-Zeichensätze
STACSS1-16	Nummern der unterstützten 8-bit/Unicode-Zeichensätze
STACURCH	enthält den Namen des Standard-Benutzerzeichensatzes, wenn die Datenstation 8-bit-fähig ist und dieser Zeichensatz kompatibel zur Datenstation ist.
STAACTCH	enthält den Namen des aktiven erweiterten Zeichensatzes.

Diese Schnittstellen existieren auch für COBOL, C, PL/1 und FORTRAN.

Neben dem VTSU-Makro DCSTA können Sie sich auch über das BS2000-Kommando SHOW-TERMINAL-ATTRIBUTES Informationen über alle Eigenschaften Ihrer Datenstation ausgeben lassen. Mit Hilfe dieses Kommandos können Sie feststellen, in welchem Modus (z.B. 7-bit, 8-bit, Unicode) Ihre Datenstation oder Emulation betrieben wird. Nähere Informationen dazu finden Sie in den Beschreibungen zu den BS2000-Kommandos [5].

3.3.1.3 Programmierhinweise

- Wird XHCS nicht über den VTSUCB oder über das Kommando MODIFY-TERMINAL-OPTIONS aufgerufen, kehrt VTSU automatisch zum 7-bit-Modus zurück. Die folgende Nachrichtenverarbeitung findet dann wieder im 7-bit-Modus statt. Bei diesem Modus-Wechsel wird der Bildschirm gelöscht.
- Zur Code-Umwandlung muss VTSU die ESCAPE-Sequenzen zur Druckersteuerung aus der Nachricht herausnehmen. Damit VTSU diese Sequenzen erkennt, müssen sie im Standard EBCDIC.DF.03 codiert sein. Wenn Sie im physikalischen Modus (MODE=PHYS) mit Druckern arbeiten, können Sie jedoch spezielle ESCAPE-Sequenzen benutzen, die nicht zum EBCDIC-Kern gehören. VTSU erkennt diese speziellen ESCAPE-Sequenzen nicht und versucht, sie gemäß dem festgelegten Zeichensatz umzuwandeln. Dies kann zu Inkonsistenzen bei der Code-Umwandlung führen. Um in diesem Fall eine Code-Umwandlung durch VTSU zu unterdrücken, wird im VTSUCB der Parameter CODETR=NO angeboten.
- Im VTSUCB wird der Name des zu verwendenden Zeichensatzes festgelegt. Sie können Leerzeichen (Standard, kein VTSUCB wird benutzt), *EXTEND (der erweiterte Standard-Anwender-Code wird benutzt) oder explizit einen der aktuell unterstützten Code-Namen angeben.
- Ein leerer Code-Name wirkt genauso wie ein nicht benutzter VTSUCB, d.h. die angeschlossene Datenstation bleibt im 7-bit-Modus und es findet keine Umwandlung durch VTSU statt. Wurde die Datenstation jedoch über das Kommando MODIFY-TERMINAL-OPTIONS in den 8-bit-Modus gesetzt, findet immer eine Umwandlung statt.
- Wird von einem Zeichensatz zu einem anderen gewechselt, wird der Bildschirm automatisch gelöscht. Bei MODE=LINE oder MODE=EXTEND werden Sie aufgefordert einen Modus-Wechsel zu bestätigen. Dadurch können Sie den Bildschirm noch lesen bevor er gelöscht wird. Ein Modus-Wechsel ist ein Wechsel von einem 7-bit-Modus in einen 8-bit-Modus, von einem 8-bit-Modus in einen 7-bit-Modus oder von einem 8-bit-Modus in einen anderen 8-bit-Modus.
- Wenn XHCS nicht geladen ist, werden Nachrichten wie 7-bit-Nachrichten verarbeitet. Werden trotzdem die XHCS-Funktionen CODETR und CCSNAME aufgerufen, wird der Aufruf mit einer entsprechenden Fehlermeldung im VTSUCB abgelehnt.
- Bei der Angabe des Zeichensatznamens müssen Sie immer die EBCDIC-Variante des Code-Namens angeben. Der Name der entsprechenden ISO-Code-Variante wird automatisch abgelehnt. Ebenso müssen Sie Code-Namen von voll kompatiblen Codes angeben, da begrenzt kompatible Codes abgelehnt werden.

3.3.1.4 Unterstützung spezieller Datenstationen durch VTSU-Sonderroutinen

Europäische 7-bit-Terminals, arabische/persische 8-bit-Terminals und Escape-Drucker sind spezielle Datenstationen, die durch VTSU-Sonderroutinen unterstützt werden müssen.

Die Sonderroutinen werden intern durch VTSU aufgerufen; sie sind im Lieferumfang von VTSU enthalten und können in das Gesamtsystem eingebunden werden.

Eine detaillierte Beschreibung der VTSU-Sonderroutinen, der Konfigurationsdatei und der Installationsprozedur finden Sie im Handbuch „VTSU“ [1].

Systemumgebung für die VTSU-Sonderroutinen

VTSU-B ab V11.0A

Europäische 7-bit-Terminals

XHCS unterstützt standardmäßig nur 8-bit-Stationen. Mithilfe der Funktion „Nationale 7-bit-Unterstützung“ in VTSU-B werden nationale 7-bit-Geräte unterstützt.

Folgende 7-bit-Code-Varianten sind dabei möglich:

– EBCDIC.NHC.SWE (schwedischer 7-bit-Code)
– EBCDIC.NHC.HUN (ungarischer 7-bit-Code)
– EBCDIC.NHC.CYR (kyrillischer 7-bit-Code)
– EBCDIC.NHC.GRE (griechischer 7-bit-Code)

Die Definition der Geräte und der gewünschten Code-Variante erfolgt in einer so genannten Konfigurationsdatei, die vom Anwender erstellt werden muss. Die Konfigurationsdatei muss eine SAM-Datei sein, den Namen „SYSPAR.VTSU-B.*xxx*.CONFIG“ (*xxx*=Versionsnummer) haben und unter $TSOS stehen oder mit IMON installiert sein. Die Unterstützung der europäischen 7-bit-Terminals wird über eine Installationsprozedur aktiviert. Die Installationsprozedur ist im Lieferumfang von VTSU-B enthalten.

Escape-Drucker

Escape-Drucker müssen wie europäische 7-bit-Terminals mit Stationsnamen, Rechnernamen und Variante in der Konfigurationsdatei definiert werden. Zur Unterstützung von Escape-Druckern muss die Installationsprozedur nicht aufgerufen werden. Die Escape-Drucker müssen aber über den PDN-Freitextparameter (PDN ab V11) oder in der entsprechenden VTSU-Betriebsparameterdatei als 8-bit-Drucker definiert werden.

ACHTUNG!
Der PDN-Freitextparameter hat eine höhere Priorität als die entsprechenden VTSU-Betriebsparameter.

Arabische/persische 8-bit-Terminals

Das „Standard-XHCS“ unterstützt keine arabischen bzw. persischen Codes. Mithilfe der Funktion „ARA-/FAR-/NAF-Unterstützung“ in VTSU-B werden arabische/persische/arabisch-französische Codes zur Verfügung gestellt. Die Funktion beschränkt sich auf 8-bit-Stationen (DSS-9756, DSS-9758, DSS-9763) mit arabischer, persischer oder arabisch-französischer Firmware.

Folgende Code-Varianten sind möglich:
EBCDIC.EHC.LA.ARA
EBCDIC.EHC.LA.IND
EBCDIC.EHC.LF.INT
EBCDIC.EHC.LF.FAR
EBCDIC.EHC.NA.ARA
EBCDIC.EHC.NA.IND

Die arabischen/persischen 8-bit-Terminals können Sie nur nutzen, wenn Sie die Unterstützung dieser Terminals mit der Installationsprozedur aktiviert haben. Die Installationsprozedur ist im Handbuch „VTSU“ [1] beschrieben.

Aus Sicht des Anwenders werden arabische/persische 8-bit-Terminals im XHCS wie europäische 8-bit-Terminals behandelt. Der 8-bit-Modus kann von Programmen direkt über den VTSUCB oder über die Aktivierung des 8-bit-Anwenderstandards mit dem TIAM-Kommando MODIFY-TERMINAL-OPTIONS oder über die VTSU-Betriebsparameter eingestellt werden.

Namen von arabischen/persischen 8-bit-Codes:

ISO-Code-Variante	XHCS-Name (CCSN)	vollständiger Name	Darstellung
6	EDF046 EEHCLAA	EBCDIC.DF.04-6.ARA EBCDIC.EHC.LA.ARA	Latein/Arabisch mit arabischen Ziffern
A	EDF04A EEHCLAI	EBCDIC.DF.04-6.IND EBCDIC.EHC.LA.IND	Latein/Arabisch mit indischen Ziffern
C	EDF04C EEHCLFI	EBCDIC.DF.04-FAR.INT EBCDIC.EHC.LF.INT	Latein/Farsi mit internationalen Ziffern
D	EDF04D EEHCLFF	EBCDIC.DF.04-FAR.FAR EBCDIC.EHC.LF.FAR	Latein/Farsi mit persischen Ziffern
E	EDF04E EEHCNAA	EBCDIC.DF.04-NAF.ARA EBCDIC.EHC.NA.ARA	Franz./Arabisch mit arabischen Ziffern
F	EDF04F EEHCNAI	EBCDIC.DF.04-NAF.IND EBCDIC.EHC.NA.IND	Franz./Arabisch mit indischen Ziffern

Eine vollständige Liste aller gültigen CCSN (XHCS-Namen) ist ab Seite 145 aufgeführt.

Terminal-Emulation MT9750 im Unicode-Modus

Mit der Terminal-Emulation MT9750 V7.0 ist es möglich, Unicode-Daten an das BS2000/OSD zu senden. Als Terminalstyp muss DSS9763 eingestellt sein. Dazu müssen die Sitzungsparameter der Emulation entsprechend konfiguriert werden. Dadurch wird VTSU mitgeteilt, dass das Terminal neben den EBCDIC-Zeichensätzen auch den Zeichensatz UTFE (Unicode) unterstützt. Dies ist Voraussetzung dafür, dass das BS2000-Kommando /MODIFY-TERMINAL-OPTIONS CCS = UTFE möglich ist.
Wenn eine Anwendung Unicode-Daten an die Emulation senden oder von ihr empfangen will, muss dies VTSU mitgeteilt werden. Dies kann in den Anwendungen über den VTSU-B-Kontrollblock oder über das BS2000-Kommando /MODIFY-TERMINAL-OPTIONS erfolgen. Die Einstellung im VTSU-B-Kontrollblock hat Vorrang vor dem mit /MODIFY-TERMINAL-OPTIONS eingestellten Wert. Er gilt jedoch nur für den aktuellen Aufruf. Nähere Information zur Konfiguration der Terminal-Emulation MT9750 finden Sie im Handbuch „Unicode im BS2000/OSD“ [32].

3.3.2 XHCS-Programmschnittstellen

Das Subsystem XHCS bietet eine zentrale Verwaltung von Code-Tabellen. Um die Vorteile von XHCS zu nutzen, stehen verschiedene Programmschnittstellen zur Verfügung.

- Programmschnittstelle NLSCOD - Code-Informationen
  Versorgt verschiedene Tabellen (eine Tabelle pro Aufruf) und stellt sie bereit.
  Die möglichen Tabellen sind:
  – Übersetzungstabelle in einen anderen Code
  – Umsetzungstabelle von Klein- in Großbuchstaben
  – Tabelle der Sortiergewichte für jedes Zeichen
  – Tabelle, die einen vom Anwender wählbaren Satz von bis zu acht Eigenschaften für jedes Zeichen eines Codes zusammenfasst
  – Tabelle mit der höchsten, von XHCS unterstützte Unicode-Position, der Größe der Sortierinformation für Unicode und der Größe der Normalisierungstabelle
- Programmschnittstelle NLSCNV - Zeichenketten umwandeln
  – Führt die Code-Umsetzung von Zeichenfolgen durch. Eine Zeichenfolge wird von einem CCS in einen anderen übersetzt.
  – Gibt die Länge von Unicode-Zeichenketten aus
  – Setzt Klein- in Großbuchstaben um und umgekehrt
  – Normalisiert Eingabezeichenketten für Unicode
- Programmschnittstelle NLSCMP - kompatible Codes
  Gibt Informationen über die Kompatibilität der Codes und über die Möglichkeit der Umsetzung.
- BS2000-Kommando /ADD-CODE-TABLES - neue Code-Tabellen dynamisch hinzufügen
  Ermöglicht die dynamische Änderung und Erweiterung bestehender Codes im laufenden Betrieb (Die Kommandobeschreibung finden Sie ab Seite 134.)
- Andere Schnittstellen
  Der codierte Zeichensatz einer Datei (PLAM-Bibliothekselemente) wird von verschiedenen DMS-Schnittstellen (ILAM) zur Verfügung gestellt.

Die Beschreibung für NLSCOD, NLSCNV und NLSCMP finden Sie ab Seite 65.

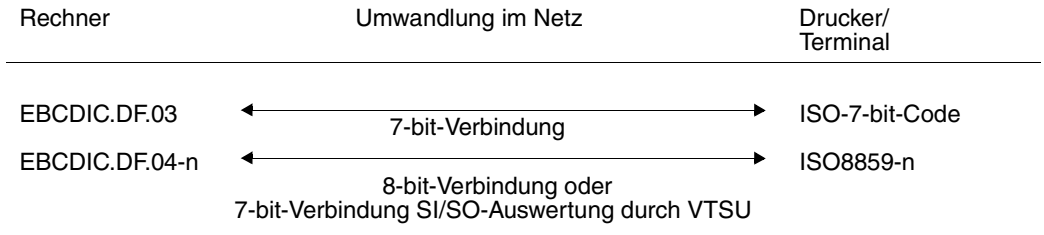
3.4 Code-Umwandlung

Da BS2000-Rechner mit einem EBCDI-Code arbeiten, Terminal und Drucker jedoch grundsätzlich nur ISO-Codes verarbeiten, findet bei der Übertragung zwischen Rechner und Peripherie eine Umcodierung von ISO-Code in EBCDI-Code oder von EBCDI-Code in ISO-Code statt.

Die Netzsoftware des PDN (Programmsystem für Datenfernverarbeitung und Netzsteuerung) wandelt zwischen diesen Codes um. Die Umwandlung ist abhängig von der PDN-Generierung, die sich nach den Hardware-Eigenschaften der Verbindung und den angeschlossenen Geräten richtet (siehe Handbücher zu den Terminals und Druckern). 8-bit-Terminal oder 8-bit-Drucker können dabei mit dem Rechner über zwei verschiedene Verbindungsarten verbunden sein. Zum einen über 7-bit-Verbindung und zum anderen über 8-bit-Verbindung (z.B. HDLC).

Wird ein erweiterter Zeichensatz über eine 7-bit-Verbindung übertragen, wird ein Code-Erweiterungsmechanismus verwendet. Dazu werden die Steuerzeichen Shift In (SI) und Shift Out (SO) benutzt, die dann durch VTSU ausgewertet werden. Durch die Steuerzeichen SI/SO kann sich die Nachrichtenlänge verdoppeln.

Das folgende Bild stellt graphisch die Code-Umwandlung zwischen Rechner und Peripherie dar.



Der EBCDIC.DF.04-n ist eine Erweiterung des EBCDIC.DF.03-IRV (=Internationale Referenz Version). Eine Variante des EBCDIC.DF.03-IRV ist der EBCDIC.DF.03-DRV (=Deutsche Referenz Version). Beide Codes unterstützen Zeichensätze in der gleichen Größe. Sie haben einen gemeinsamen Zeichensatz, den so genannten EBCDIC-Kern und unterscheiden sich nur in einigen Symbolen (siehe Tabelle EBCDIC.DF.03). System- und Anwendungsprogramme, die nicht durch XHCS unterstützt werden, benutzen diese EBCDIC.DF.03-Codes.

Der ISO 8859-n ist eine Erweiterung des ASCII-Codes. Als ASCII-Code wird die US-Variante des 7-bit-Codes gemäß ISO646 bezeichnet. Neben dem internationalen ASCII-Code gibt es noch weitere nationale Varianten des 7-bit-Codes gemäß ISO646. Dieser 7-bit-Code wird von 7-bit-Terminals und von 8-bit-Terminals im 7-bit-Modus benutzt. Um einen Text oder einen Tastendruck richtig zu interpretieren, muss ein 7-bit-Terminal lokal über Tastatur (siehe SIDATA-Menüs) in den entsprechenden Modus (z.B. „Deutsch“) gesetzt werden. Der erweiterte Zeichensatz ISO 8859 besitzt mehrere landesspezifische Varianten.

Die folgenden Bilder zeigen Ihnen Beispiele für die Code-Umwandlung zwischen Rechner und Peripherie.

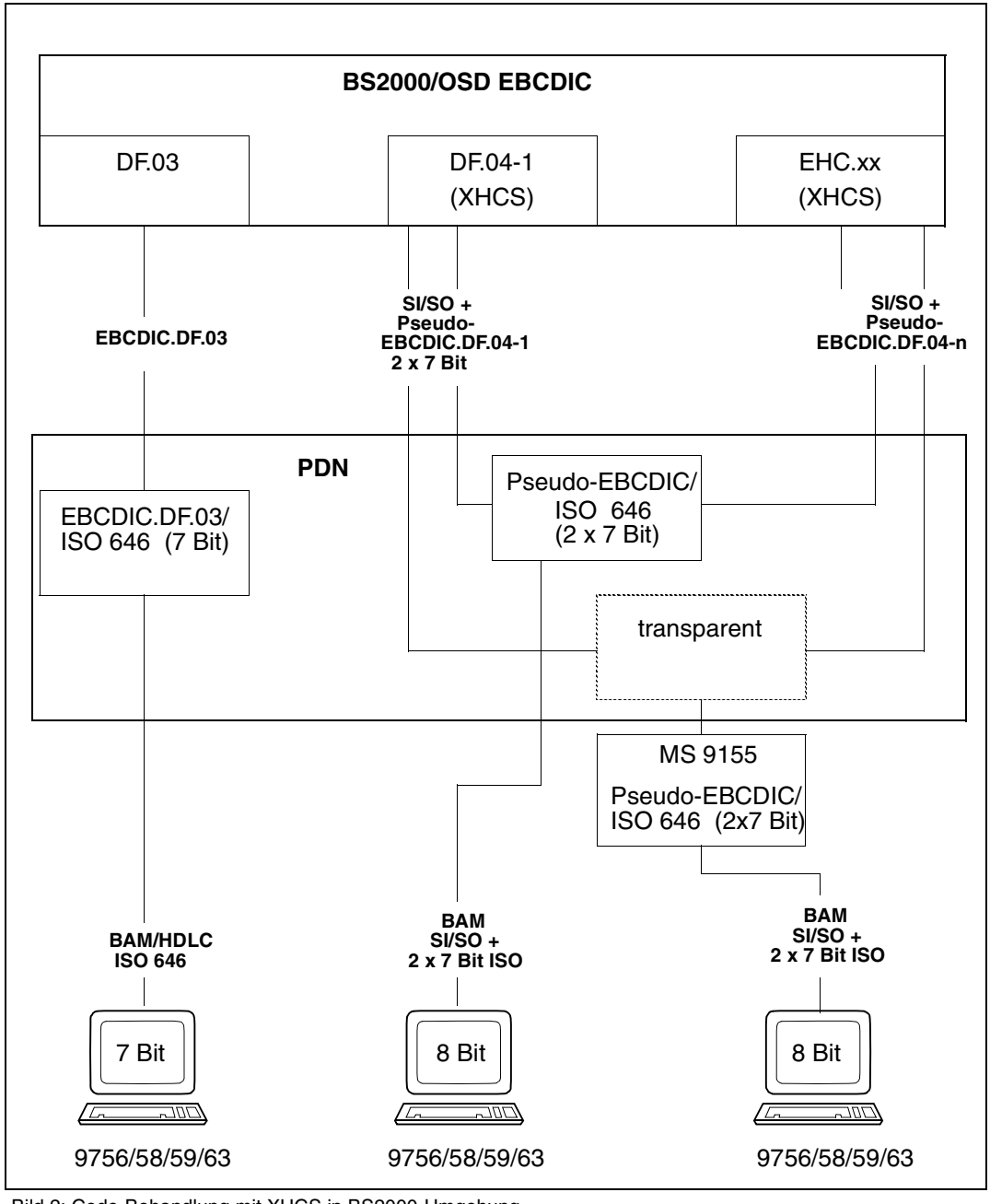


Bild 2: Code-Behandlung mit XHCS in BS2000-Umgebung

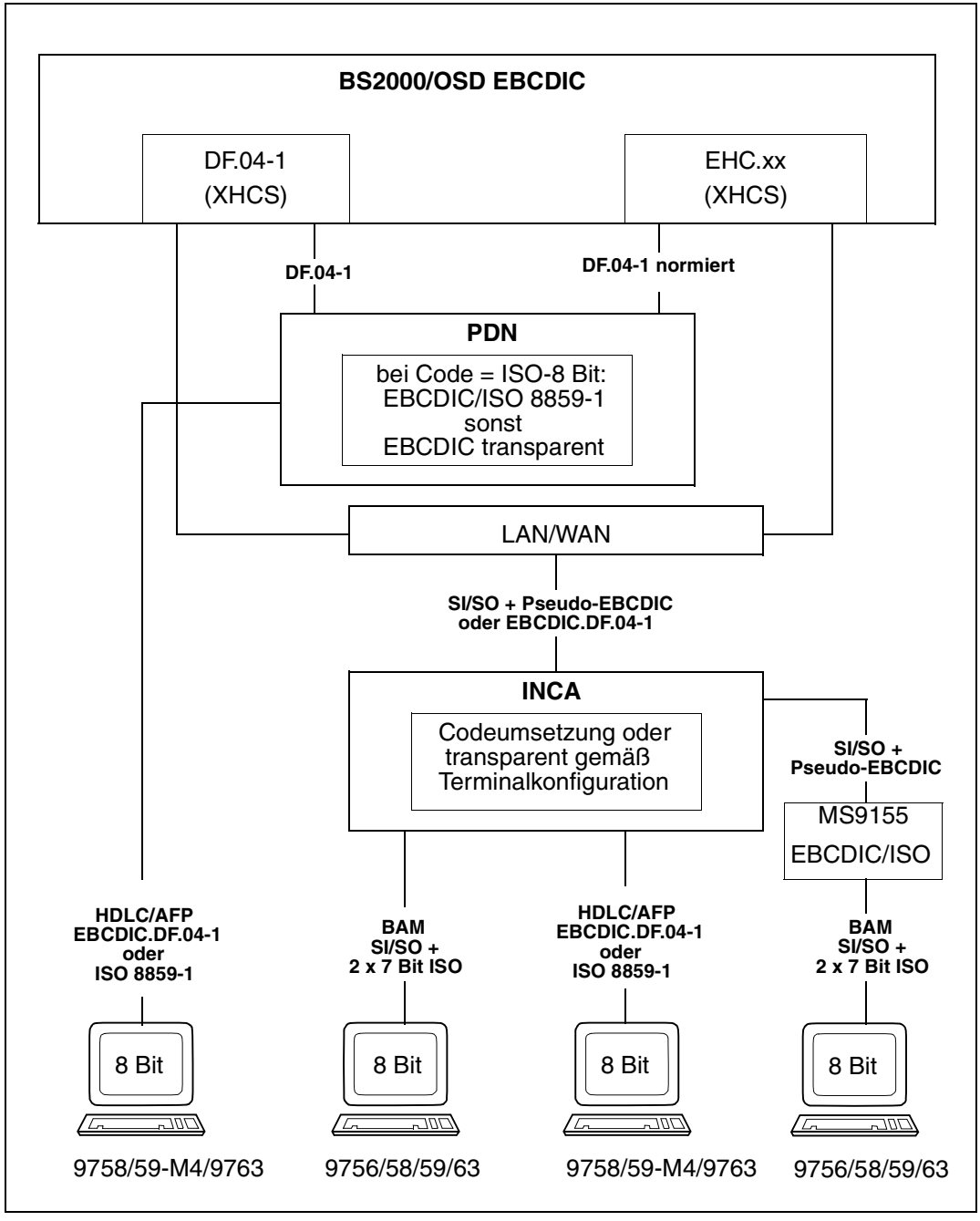


Bild 3: Code-Behandlung mit XHCS in BS2000-Umgebung

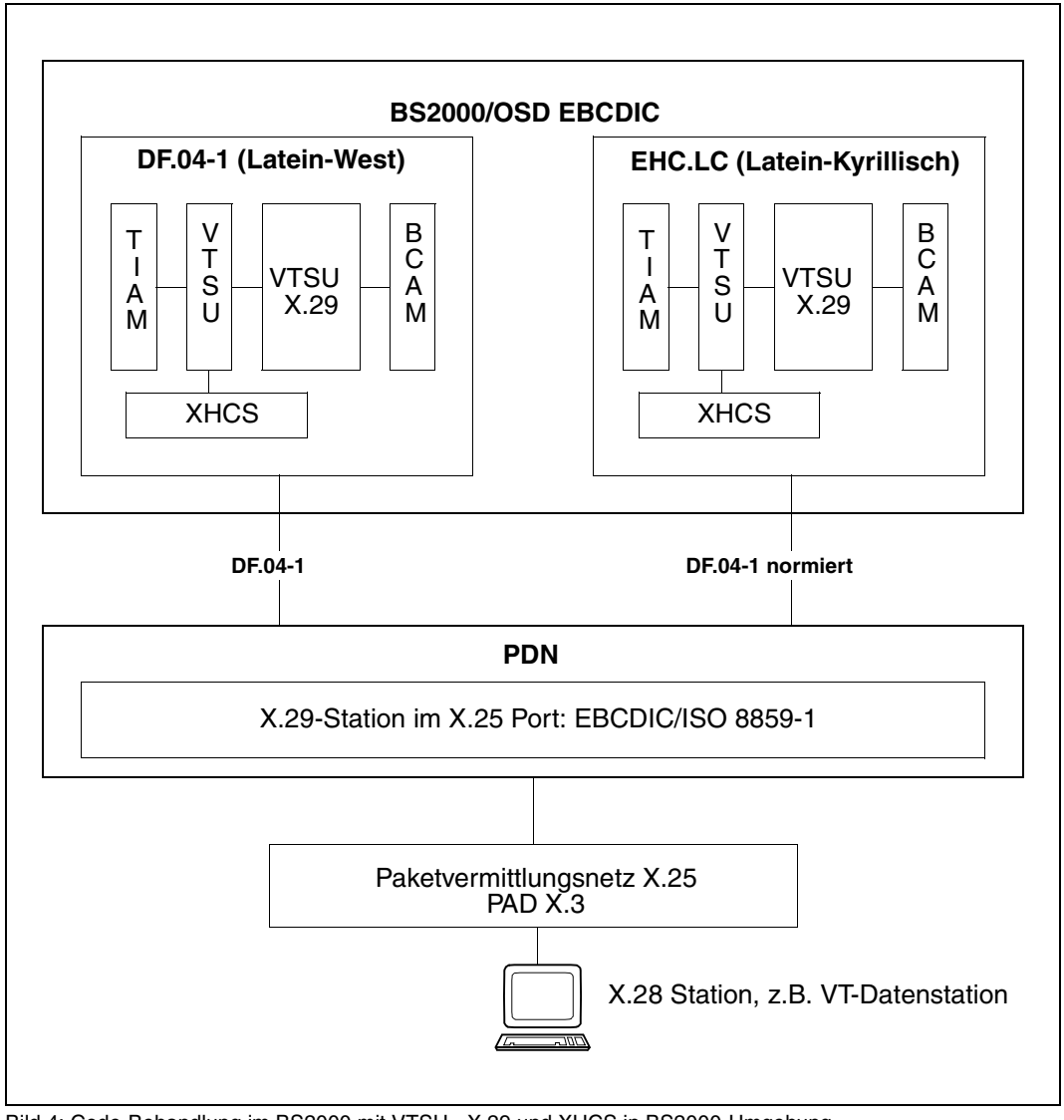


Bild 4: Code-Behandlung im BS2000 mit VTSU - X.29 und XHCS in BS2000-Umgebung

Die nachfolgende Abbildung stellt Unicode im BS2000/OSD-Systemumfeld beispielhaft in Form der Unicode-Codierungen an den einzelnen Schnittstellen dar. Für eine größere Übersichtlichkeit wurde auf einige technische Details verzichtet. Nähere Informationen dazu finden Sie im Handbuch „Unicode im BS2000/OSD“ [32].

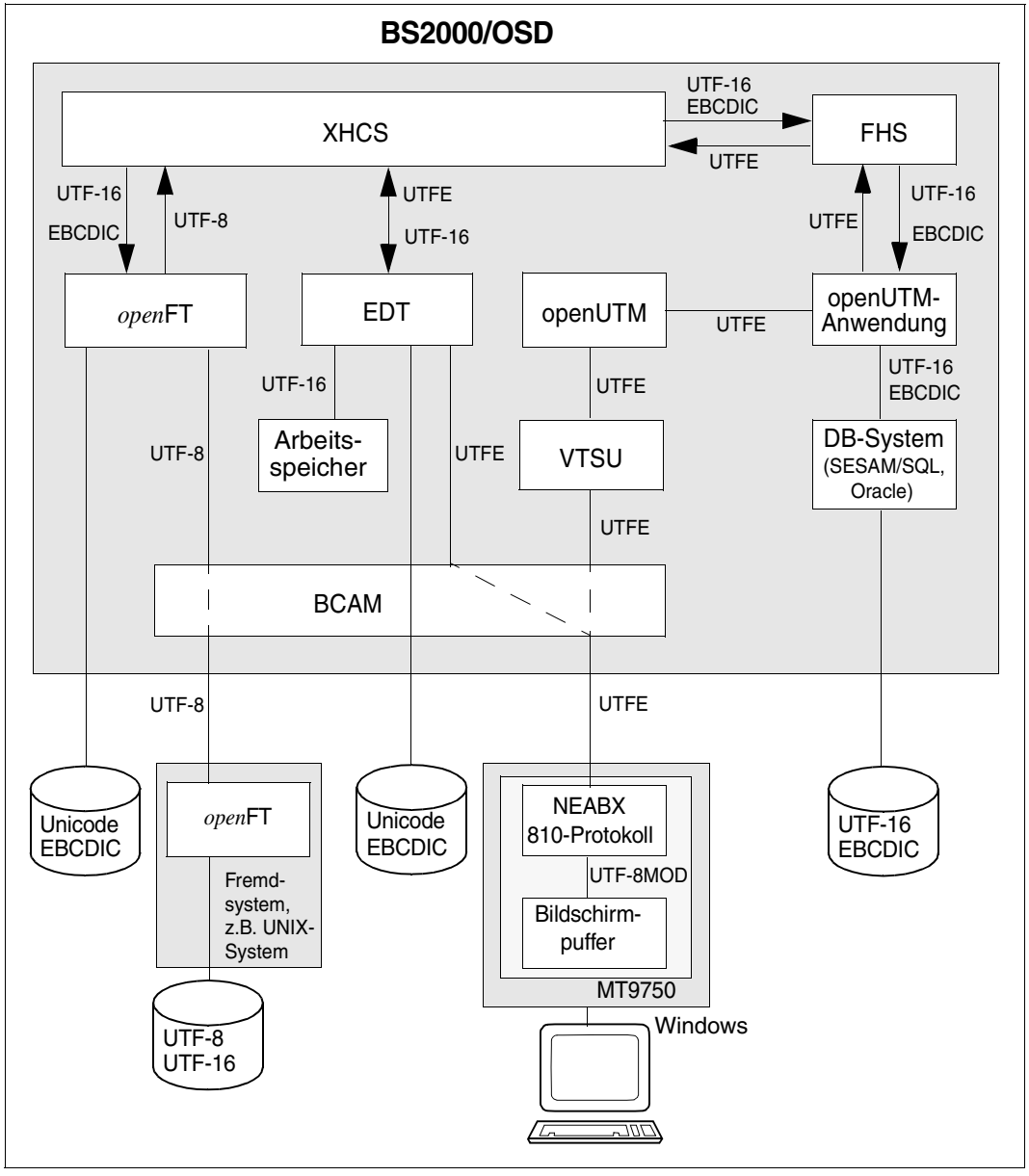


Bild 5: Überblick über die von UNICODE betroffenen Schnittstellen

3.5 Einsatz erweiterter Codes

Der folgende Abschnitt beschreibt, was Sie beim Einsatz erweiterter Codes beachten müssen, und gibt Ihnen Hinweise und Tipps für die Arbeit mit XHCS.

Damit sich die Programme darauf verlassen können, korrekt zu arbeiten, ohne an einen vorgegebenen Zeichensatz fest gebunden zu sein, müssen folgende Bedingungen erfüllt sein:

- 8-bit-fähige Geräte
- Erweitertes Terminalprotokoll und Informationen über die Verbindung
- Identifizieren der Codes von Dateien und Bibliothekselementen
- Steuerung des 8-bit-Modes
- Anzeige des vom Benutzer bevorzugten Rechnercodes
- Zentrale Code-Tabellen, die Umsetztabellen und zusätzliche Informationstabellen enthalten
- 8-bit-Codes, Schriftrichtung von links nach rechts

3.5.1 Erweiterte Terminalprotokolle

Das Übertragungsprotokoll zwischen Rechner und Datenstation ist in neuen Geräten erweitert. Der Gerätemodus (7-bit oder 8-bit) kann dynamisch eingestellt werden, und der Zeichensatz kann von der Anwendung angegeben werden.

Seit einige Datenstationen verschiedene Zeichensätze unterstützen, ist es möglich, Datenstationen abzufragen, ob sie 8-bit-fähig oder Unicode-fähig sind und, wenn ja, welche Zeichensätze zur Verfügung stehen.

Die Terminal-Emulation MT9750 ist ab V7.0 Unicode-fähig. Dazu müssen jedoch die Sitzungsparameter der Emulation entsprechend konfiguriert werden.

Die Information über die 8-bit-Fähigkeit und die unterstützten Zeichensätze erhält man mithilfe der „Terminal-Status“-Funktion, die von der verwendeten Zugriffsmethode abhängt (TSTAT für TIAM, YINQUIRE für DCAM). Mit dem VTSU-Makro DCSTA (TYPE=BASIC) erhalten Sie Informationen über die relevanten Charakteristika der Datenstation und der Verbindung. Die folgenden Felder des DCSTA-Makros beziehen sich auf XHCS:

STATTYPE	zeigt an, ob die Datenstation im 8-bit-Modus arbeiten kann
STACCSNN	Anzahl der unterstützten 8-bit/ Unicode-Zeichensätze
STACSS1-16	Nummern der unterstützten 8-bit/ Unicode-Zeichensätze
STACURCH	enthält den Namen des Standard-Benutzerzeichensatzes, wenn die Datenstation 8-bit-fähig ist und dieser Zeichensatz kompatibel zur Datenstation ist.

STAACTCH enthält den Namen des aktiven erweiterten Zeichensatzes.

Diese Schnittstellen existieren auch für COBOL, C, PL/1 und FORTRAN.

3.5.2 Identifikation der Codes von Dateien und Bibliothekselementen

Da in einem vorhandenen System verschiedene Codes zur gleichen Zeit verwendet werden können, kennzeichnet XHCS Dateien und PLAM-Bibliothekselemente mit dem CCSN (Name des codierten Zeichensatzes). Diese Information kann das Programm verwenden, das diese Daten liest bzw. schreibt.

Die CCSN der Dateien werden als Dateiattribut mit dem Namen CODED-CHARACTER-SET gespeichert. Dieses Attribut kann mit dem Kommando MODIFY-FILE-ATTRIBUTES gesetzt werden, zum Beispiel

```
/MODIFY-FILE-ATTR FILE-NAME=MYFILE,C-C-S=EDF041
```

Das Kommando SHOW-FILE-ATTRIBUTE gibt den CCSN der angegebenen Dateien zurück, zum Beispiel

```
/SHOW-FILE-ATTRIBUTE FILE-NAME=MYFILE,INFO=PAR(ORG=YES)
```

3.5.3 8-bit-Modus / Unicode-Modus aktivieren

Der 8-bit-Modus oder Unicode-Modus muss eingestellt werden. Damit wird vermieden, versehentlich 8-bit-Daten oder Unicode-Daten zu erzeugen, die dann bei der Rückgabe an 7-bit-Datenstationen Probleme bereiten.
Eine genaue Beschreibung zu Unicode finden Sie im im Handbuch „Unicode im BS2000/OSD“ [32].

Die folgenden Tipps beziehen sich nur auf 8-bit-fähige oder Unicode-fähige Datenstationen und setzen voraus, dass der gewünschte CCS (Zeichensatz) angezeigt werden kann. Versuche, einen CCS zu aktivieren, der nicht vom Gerät unterstützt wird, werden im 7-bit-Mode bearbeitet.

3.5.3.1 Explizite Aktivierung per Programm

Programme, die 8-bit-Dateien oder Unicode-Dateien verarbeiten können, können den 8-bit-Modus oder den Unicode-Modus selbst einstellen. Da Dateien und Bibliothekselemente ein Attribut haben, das den Code anzeigt, können diese Komponenten entscheiden, ob 8-bit-Dateien oder Unicode-Dateien an der Datenstation angezeigt werden können oder nicht.

Die explizite Aktivierung erfolgt durch den VTSU-Control-Block (nur bei TIAM und DCAM).

3.5.3.2 Implizite Aktivierung durch Benutzerkommandos

Es ist auch möglich im 8-bit-Modus oder Unicode-Modus zu arbeiten, ohne dass die Anwendung das anfordert. Der Anwender kann veranlassen, dass VTSU einen 8-bit-Code oder Unicode für seine Ein-/Ausgaben verwendet. Dies erfolgt mit dem TIAM-Kommando

```
/MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=<ccs-name eines 8-bit
Zeichensatzes>
oder
/MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=UTFE
```

Damit wird der 8-bit-Modus mit dem CCS des Standard-Anwenderzeichensatzes auf der Rechnerseite aktiviert und es wird ein „permanenter 8-bit-Modus“ erzeugt.

Dieses Kommando muss in manchen Fällen angegeben werden, bevor die Komponente aufgerufen wird, die XHCS benötigt.

3.5.3.3 Automatische Aktivierung durch Systemkonfiguration

Eine dritte Möglichkeit ist weniger flexibel, da sie für das gesamte BS2000-System gilt; VTSU kann so konfiguriert werden, dass der Standard-Anwenderzeichensatz angewendet wird und zwar bei TIAM, openUTM und/oder DCAM-Anwendungen, die mit einer 8-bit-Datenstation verbunden sind. Damit wird ein „permanenter 8-bit-Modus“ erzeugt.

3.5.4 Standard-Anwenderzeichensatz

Jedem Anwender kann ein Standard-Anwenderzeichensatz zugewiesen werden; dies ist der gleiche CCS, der verwendet wird, wenn der 8-bit-Modus aktiviert ist und kein Code explizit angefordert wird. Er kann vom Systemadministrator mit dem Kommando ADD-USER oder MODIFY-USER-ATTRIBUTES zugewiesen werden. Über das Kommando SHOW-USER-ATTRIBUTES wird er zurückgeliefert (siehe Handbücher „BS2000/OSD-BC Kommandos Band1 - 5“ [4]).

3.5.5 Zentralisierung von Code-Tabellen

Programme müssen Informationen über Zeichensätze nicht fest speichern. Sie erhalten verschiedene Informationen vom XHCS-Subsystem mithilfe des Zeichensatznamen (CCSN).

Ein Code kann nur verwendet werden, wenn er in XHCS definiert ist, dies liegt in der Verantwortung des Systembetreuers. Es gibt keinen direkten Weg festzustellen, welche Codes im System verfügbar sind. Indirekt erhält man sie mithilfe des EDT ab V16.4; die Anweisung @SHOW CCS liefert eine Liste der verfügbaren Zeichensätze.

3.5.6 Anwendungen mit XHCS-Unterstützung

Vorhandene Programme (außer openUTM-Anwendungen) können nur dann mit XHCS zusammenarbeiten, wenn das Kommando

```
/MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET= 8-BIT-DEFAULT
```

vor dem Programmstart eingegeben wurde. Wird das Kommando nicht eingegeben, wird mit dem Standard-Code für 7-bit-Geräte gearbeitet. Der Rechner-Code ist dann der EBCDIC.DF.03.

Code-Transparenz

Programme, die XHCS nutzen, müssen hinsichtlich des Zeichensatzes unabhängig sein. Code-Transparenz bedeutet, dass ein Programm keine Annahme macht über die Struktur des Datencodes. Diese Informationen (Zeichenklassifikation, Vergleichssequenzen, Informationen über Großbuchstaben) müssen von einem externen Mechanismus beschafft werden.

3.6 XHCS mit FHS und IFG

Ab IFG V8.3 können Sie Formate für 8-bit-Terminals (9763 und 9758) oder Terminals im Unicode-Modus (MT9750) erzeugen und aktualisieren. Dazu müssen Sie 8-bit-Terminals oder Terminals im Unicode-Modus benutzen. Die Formate, die ausschließlich an Terminals im 8-bit-Modus erstellt wurden, heißen 8-bit-Formate. Formate die an Terminals im Unicode-Modus erstellt wurden heißen Unicode-Formate. Formate älterer IFG-Versionen können Sie in 8-bit-Formate umwandeln, indem Sie die entsprechende Datenstationsgruppe ändern. Der Zeichensatzname (CCSN) eines Formates wird später von FHS verwendet, um die notwendigen Code-Tabellen zu erhalten, für die Ermittlung der darstellbaren Zeichen oder die Umwandlung von Klein- in Großbuchstaben.

Durch eine Änderung der Datenstationsgruppe können auch Formate für 8-bit-Drucker erzeugt werden.

Wie Sie die 8-bit-Unterstützung oder Unicode-Unterstützung im Format-Modus nutzen können, finden Sie in den entsprechenden FHS- und IFG-Handbüchern.

3.6.1 TIAM-Anwendung

Eine TIAM-Anwendung kann 8-bit-Nachrichten oder Unicode-Nachrichten nur an 8-bit-Terminals oder Terminals im Unicode-Modus (in der Regel eine Terminal Emulation) senden. Bei einer TIAM-Anwendung wird zum Zeitpunkt des Verbindungsaufbaus automatisch der Status des Terminals abgefragt. Nach Eintreffen der Statusantwort können Sie die Informationen über den Datenstationsstatus (z.B. 8-bit-Fähigkeit der Datenstationen) mit dem Makro TSTAT abfragen. Der Makro TSTAT liefert die folgenden Informationen:

– Typ des Terminals. Das Terminal ist entweder ein 7-bit-Terminal, ein 8-bit-Terminal oder ein Terminal im Unicode-Modus.

– Name des erweiterten Standard-Codes, wenn im 8-bit-Modus gearbeitet werden kann.

– Name des Unicodes, wenn im Unicode-Modus gearbeitet werden kann.

– Name des aktivierten erweiterten Standard-Codes, wenn ein 8-bit-Modus eingeschaltet wird (Kommando MODIFY-TERMINAL-OPTIONS).

– Name des aktivierten Unicodes, wenn ein Unicode-Modus eingeschaltet wird (Kommando MODIFY-TERMINAL-OPTIONS).

– Liste der unterstützten ISO-Code-Varianten

Anschließend wird bei der LOGON-Verarbeitung überprüft, ob der benutzte Standard-Code zu den Codes kompatibel ist, die durch die aktuelle Datenstation unterstützt werden. Wird der Code nicht unterstützt, wird eine Warnung ausgegeben und die Verbindung als 7-bit-Verbindung angesehen.

Mit dem Parameter CCSNAME im VTSUCB können Sie für eine TIAM-Anwendung eindeutig den 8-bit-Modus oder Unicode-Modus festlegen. Wird der Modus nicht eindeutig festgelegt, arbeitet das Terminal automatisch im Standard-Modus (7-bit-Modus oder ein mit dem Kommando MODIFY-TERMINAL-OPTIONS aktivierter 8-bit-Modus). Dadurch wird die volle Daten-Kompatibilität garantiert. Im VTSUCB können Sie für den Parameter CCSNAME zwei Werte angeben:

– Bei CCSNAME = *EXTEND wird automatisch der erweiterte Standard-Code benutzt.

– Bei CCSNAME = ccsname wird explizit der Name des zu verwendenden Zeichensatzes angegeben. Dies kann EDF03IRV, der Name eines erweiterten Standard-Codes oder ein Unicode sein.

Die Zuordnung des Codes zum entsprechenden Format legen Sie fest. Diese Zuordnung wird durch das System nicht überprüft. Sollte dem Format ein inkompatibler Code zugeordnet sein, können folgende Fehler auftreten:

– Wird im VTSUCB ein 8-bit-Code oder Unicode festgelegt und ein 7-bit-Format gesendet, kann es zu einem Formatierungsfehler kommen, da FHS Zeichen eines erweiterten Zeichensatzes nicht erkennt.

– Wird im VTSUCB ein 7-bit-Code festgelegt und ein 8-bit-Format oder Unicode-Format gesendet, kann es zu einem Ausgabefehler kommen, da VTSU Zeichen eines erweiterten Zeichensatzes oder Unicode-Zeichensatzes nicht erkennt.

– Wird ein 8-bit-Format gesendet und im VTSUCB ein anderer 8-bit-Code festgelegt, kann es zu Fehlern kommen, da FHS zur Formatierung und VTSU zur Ausgabeaufbereitung unterschiedliche Codes benutzen.

– Wird ein Unicode-Format gesendet und im VTSUCB ein 8-bit-Code festgelegt, kann es zu Fehlern kommen, da FHS zur Formatierung und VTSU zur Ausgabeaufbereitung unterschiedliche Codes benutzen.

Eine TIAM-Anwendung, die 8-bit-Formate oder Unicode verwendet, arbeitet korrekt, wenn der CCS des Formates und der CCS, den VTSU verwendet, übereinstimmen.

3.6.2 DCAM-Anwendung

Eine DCAM-Anwendung kann 8-bit-Nachrichten oder Unicode-Nachrichten nur an 8-bit-Terminals, an Terminals im Unicode-Modus und an 8-bit-Drucker senden. Um den Typ des Terminals zu ermitteln, müssen Sie bei DCAM-Anwendungen selbst den Status des Terminals (PROC = TERMSTAT in CCB beim YOPNCON-Aufruf) abfragen. Nach Eintreffen der Statusantwort können Sie die Informationen über den Datenstationsstatus (z.B. 8-bit-Fähigkeit der Datenstationen) mit dem Makro YINQUIRE abfragen. Der Makro YINQUIRE liefert die folgenden Informationen:

– Typ des Terminals. Das Terminal ist entweder ein 7-bit-Terminal, ein 8-bit-Terminal oder ein Terminal im Unicode-Modus.

– Name des erweiterten Standard-Codes, wenn im 8-bit-Modus gearbeitet werden kann.

– Name des Unicodes, wenn im Unicode-Modus gearbeitet werden kann.

– Liste der unterstützten ISO-Code-Varianten

Beim Verbindungsaufbau wird überprüft, ob der benutzte Standard-Code zu den Codes kompatibel ist, die durch die aktuelle Datenstation unterstützt werden. Wird der Code nicht unterstützt, wird die Verbindung als 7-bit-Verbindung angesehen. Es wird keine Warnung ausgegeben.

Mit dem Parameter CCSNAME im VTSUCB können Sie für eine DCAM-Anwendung eindeutig den 8-bit-Modus oder Unicode-Modus festlegen. Wird der Modus nicht eindeutig festgelegt, arbeitet die Datenstation automatisch im 7-bit-Modus. Dadurch wird die volle Daten-Kompatibilität garantiert.

Im VTSUCB können Sie für den Parameter CCSNAME zwei Werte angeben:

– Bei CCSNAME = *EXTEND wird automatisch der erweiterte Standard-Code benutzt.

– Bei CCSNAME = ccsname wird explizit der Name des zu verwendenden Zeichensatzes angegeben. Dies kann EDF03IRV,der Name eines erweiterten Standard-Codes sein oder ein Unicode sein.

Die Zuordnung des Codes zum entsprechenden Format legen Sie fest. Diese Zuordnung wird durch das System nicht überprüft. Sollte dem Format ein inkompatibler Code zugeordnet sein, können folgende Fehler auftreten:

– Wird im VTSUCB ein 8-bit-Code oder Unicode festgelegt und ein 7-bit-Format gesendet, kann es zu einem Formatierungsfehler kommen, da FHS Zeichen eines erweiterten Zeichensatzes nicht erkennt.

– Wird im VTSUCB ein 7-bit-Code festgelegt und ein 8-bit-Format oder Unicode-Format gesendet, kann es zu einem Ausgabefehler kommen, da VTSU Zeichen eines erweiterten Zeichensatzes oder Unicode-Zeichensatzes nicht erkennt.

– Wird ein 8-bit-Format gesendet und im VTSUCB ein anderer 8-bit-Code festgelegt, kann es zu Fehlern kommen, da FHS zur Formatierung und VTSU zur Ausgabeaufbereitung unterschiedliche Codes benutzen.

– Wird ein Unicode-Format gesendet und im VTSUCB ein 8-bit-Code festgelegt, kann es zu Fehlern kommen, da FHS zur Formatierung und VTSU zur Ausgabeaufbereitung unterschiedliche Codes benutzen.

3.6.3 openUTM-Anwendung

In openUTM-Anwendungen kann für jeden USER und für jeden LTERM-Partner der Parameter CCSNAME direkt generiert werden. Der angegebene CCS-Name muss zu einem im BS2000-System definierten EBCDIC-Zeichensatz oder zur Unicode-Variante UTFE gehören. Ist das Terminal oder die Terminal-Emulation (z.B. MT9750) 8-bit-fähig, wird der 8-bit-Modus unterstützt.
Ist das von einer openUTM-Anwendung gesendete Format ein 8-bit-Format oder ein Unicode-Format und ist der benutzte Standard-Code kompatibel zu den Codes, die durch die aktuelle Datenstation unterstützt werden, arbeitet das 8-bit-Terminal oder die Emulation automatisch im 8-bit-Modus oder im Unicode-Modus. Die Kompatibilität wird beim Verbindungsaufbau überprüft. Ist der Code nicht kompatibel, wird die Verarbeitung als 7-bit-Verarbeitung angesehen. Es wird keine Warnung ausgegeben.

Bei der Generierung der openUTM-Anwendung ist zu beachten, dass 8-bit-Terminals oder Terminals im Unicode-Modus entsprechend generiert werden müssen (PTERM-Anweisung bei KDCDEF). Die Deklarierung der KDCFILE muss mit der PDN-Generierung übereinstimmen.

Der erweiterte Standard-Code oder der Unicode, der für die Formatverarbeitung benutzt wird, muss mit dem Standard-Code oder dem Unicode der openUTM-Anwendung identisch sein. Ist dies nicht der Fall, kann es zu Fehlern kommen, da FHS zur Formatierung und VTSU zur Aufbereitung der Ausgabe unterschiedliche Codes benutzen.

3.7 XHCS ohne FHS und IFG

In den folgenden Abschnitten wird der Gebrauch von XHCS bei TIAM-, DCAM-, und openUTM-Anwendungen beschrieben, ohne FHS und IFG zu nutzen.

3.7.1 TIAM-Anwendung

Eine TIAM-Anwendung kann 8-bit-Nachrichten oder Unicode-Nachrichten nur an 8-bit-Terminals oder Terminals im Unicode-Modus senden. Bei einer TIAM-Anwendung wird zum Zeitpunkt des Verbindungsaufbaus automatisch der Status des Terminals abgefragt. Nach Eintreffen der Statusantwort können Sie die Informationen über den Datenstationsstatus (z.B. 8-bit-Fähigkeit der Datenstationen) mit dem Makro TSTAT abfragen. Der Makro TSTAT liefert die folgenden Informationen:

– Typ des Terminals. Das Terminal ist entweder ein 7-bit-Terminal, ein 8-bit-Terminal oder ein Terminal im Unicode-Modus.

– Name des erweiterten Standard-Codes, wenn im 8-bit-Modus gearbeitet werden kann.

– Name des aktivierten erweiterten Standard-Codes, wenn ein 8-bit-Modus eingeschaltet wird (Kommando MODIFY-TERMINAL-OPTIONS).

– Name des aktivierten Unicodes, wenn ein Unicode-Modus eingeschaltet wird (Kommando MODIFY-TERMINAL-OPTIONS).

– Liste der unterstützten ISO-Code-Varianten.

Anschließend wird bei der LOGON-Verarbeitung überprüft, ob der benutzte Standard-Code zu den Codes kompatibel ist, die durch die aktuelle Datenstation unterstützt werden. Wird der Code nicht unterstützt, wird eine Warnung ausgegeben und die Verbindung als 7-bit-Verbindung angesehen.

Mit dem Parameter CCSNAME im VTSUCB können Sie für eine TIAM-Anwendung eindeutig den 8-bit-Modus oder Unicode-Modus festlegen. Wird der Modus nicht eindeutig festgelegt, arbeitet das Terminal automatisch im Standard-Modus (7-bit-Modus oder ein mit dem Kommando MODIFY-TERMINAL-OPTIONS aktivierter Modus). Dadurch wird die volle Daten-Kompatibilität garantiert.

Im VTSUCB können Sie für den Parameter CCSNAME zwei Werte angeben:

– Bei CCSNAME=*EXTEND wird automatisch der erweiterte Standard-Code benutzt.

– Bei CCSNAME = ccsname wird explizit der Name des zu verwendenden Zeichensatzes angegeben. Dies kann EDF03IRV, der Name eines erweiterten Standard-Codes oder ein Unicode sein.

Der Parameter CCSNAME kann im erweiterten Line-Modus und im physikalischen Modus benutzt werden. Im physikalischen Modus müssen Sie dafür sorgen, dass im Nachrichtenvorspann die von VTSU benötigten Felder (z.B. ISOA, PAR01L) zur Verfügung stehen.

3.7.2 DCAM-Anwendung

Eine DCAM-Anwendung kann 8-bit-Nachrichten oder Unicode-Nachrichten nur an 8-bit-Terminals, an Terminals im Unicode-Modus und an 8-bit-Drucker senden. Um den Typ des Terminals zu ermitteln, müssen Sie bei DCAM-Anwendungen selbst den Status des Terminals (PROC = TERMSTAT in CCB beim YOPNCON-Aufruf) abfragen. Nach Eintreffen der Statusantwort können Sie die Informationen über den Datenstationsstatus (z.B. 8-bit-Fähigkeit der Datenstationen) mit dem Makro YINQUIRE abfragen. Der Makro YINQUIRE liefert die folgenden Informationen:

– Typ des Terminals. Das Terminal ist entweder ein 7-bit-Terminal, ein 8-bit-Terminal oder ein Terminal im Unicode-Modus.

– Name des erweiterten Standard-Codes, wenn im 8-bit-Modus gearbeitet werden kann.

– Name des Unicodes, wenn im Unicode-Modus gearbeitet werden kann.

– Liste der unterstützten ISO-Code-Varianten

Beim Verbindungsaufbau wird überprüft, ob der benutzte Standard-Code zu den Codes kompatibel ist, die durch die aktuelle Datenstation unterstützt werden. Wird der Code nicht unterstützt, wird die Verbindung als 7-bit-Verbindung angesehen. Es wird keine Warnung ausgegeben.

Mit dem Parameter CCSNAME im VTSUCB können Sie für eine DCAM-Anwendung eindeutig den 8-bit-Modus oder Unicode-Modus festlegen. Wird der Modus nicht eindeutig festgelegt, arbeitet die Datenstation automatisch im 7-bit-Modus. Dadurch wird die volle Daten-Kompatibilität garantiert.

Im VTSUCB können Sie für den Parameter CCSNAME zwei Werte angeben:

– Bei CCSNAME=*EXTEND wird automatisch der erweiterte Standard-Code benutzt.

– Bei CCSNAME = ccsname wird explizit der Name des zu verwendenden Zeichensatzes angegeben. Dies kann EDF03IRV, der Name eines erweiterten Standard-Codes oder ein Unicode sein.

Der Parameter CCSNAME kann im erweiterten Line-Modus und im physikalischen Modus benutzt werden. Im physikalischen Modus müssen Sie dafür sorgen, dass im Nachrichtenvorspann die von VTSU benötigten Felder (z.B. ISOA, PAR01L) zur Verfügung stehen.

3.7.3 openUTM-Anwendung

In openUTM-Anwendungen kann für jeden USER und für jeden LTERM-Partner der Parameter CCSNAME direkt generiert werden. Der angegebene CCS-Name muss zu einem im BS2000-System definierten EBCDIC-Zeichensatz oder zur Unicode-Variante UTFE gehören. Ist das Terminal oder die Terminal-Emulation (z.B. MT9750) 8-bit-fähig, wird der 8-bit-Modus unterstützt.

3.7.4 Empfehlungen

Dieser Abschnitt enthält einige Richtlinien für den Programmierer von TIAM- oder DCAM-Anwendungen.

3.7.4.1 Name des Datenzeichensatzes (CCSN) ermitteln

Wenn der zu lesende Input eine Datei oder ein PLAM-Bibliothekselement ist, wird der CCSN von verschiedenen DMS- bzw. ILAM-Schnittstellen zurückgegeben. Ist kein solcher Input vorhanden, muss festgestellt werden, welchen Namen der verwendete Rechner-Code hat.

Bei Datenstationsein-/ausgaben erhalten Sie den verwendeten CCSN abhängig vom Datentyp der Ein-/Ausgabe:

– Tasks, die interaktive Datenstationsein-/ausgaben verarbeiten (WRTRD), erhalten den CCSN aus der Rückgabe des TIAM TSTAT-Makros (Feld STAACTCH).

– Tasks, die ihre Eingabe von der SYSDTA-System-Datei lesen, erhalten den CCSN von SYSFILE GCCSN (Feld CCSNCCSN).

3.7.4.2 Verfügbarkeit von XHCS

XHCS-SYS, das Subsystem von XHCS, ist nicht standardmäßig im BS2000 vorhanden. Ob es verfügbar ist, sollte immer vorher überprüft werden. Um inkonsistente Reaktionen von Komponenten, die XHCS aufrufen, zu vermeiden, darf das XHCS-Subsystem nicht gelöscht oder entladen werden.

3.7.4.3 ISO-Codes erkennen

Der Typ des Codes (EBCDIC oder ISO/ASCII) wird von XHCS erkannt (Schnittstelle NLSCMP). ISO/ASCII-Codes sollten zurückgewiesen werden, da das BS2000 ein EBCDIC-orientiertes System ist.

3.7.4.4 Möglichkeiten der Datenstation

Bei der Verarbeitung von 8-bit-Datenstationsein-/ausgaben muss sichergestellt werden, dass der gewünschte Zeichensatz an der Datenstation dargestellt werden kann (8-bit-Fähigkeit und Unterstützung der relevanten ISO-Variante).

Der TSTAT- (TIAM) und YINQUIRE- (DCAM) Makro liefern Informationen (DCSTA) über die 8-bit-Fähigkeit (Feld STATTYPE) und über die unterstützten Codes in Form einer Liste der unterstützen ISO-Varianten. Alle 8-bit-Codes, deren ISO-Varianten in dieser Liste stehen, können verwendet werden. Die Nummer der ISO-Variante eines Codes wird vom NLSCMP-Makro an XHCS zurückgegeben.

3.7.4.5 Komplette und eingeschränkte Codes

VTSU unterstützt folgende Codes:

– im 7-bit-Mode: Unterstützung der internationalen Version von EBCDIC.DF.03 (ohne Unterstützung von XHCS-Services, d.h. mit internen Tabellen)

– im 8-bit-Mode: Unterstützung von 8-bit-Codes, die in den XHCS-Tabellen definiert sind und kompatibel zu der ISO 8859-Variante sind, die die Datenstation unterstützt.

– im Unicode-Mode: Unterstützung von Unicodes-Codes, die in den XHCS-Tabellen definiert sind und kompatibel zu der ISO 8859-Variante sind, die die Datenstation unterstützt.

Daten, deren CCSN leer oder EDF03IRV ist, müssen im 7-bit-Mode bearbeitet werden.

Um festzustellen, ob ein vorhandener Code-Name einen 7-bit- oder 8-bit-Code kennzeichnet, rufen Sie XHCS über die NLSCMP-Programmschnittstelle auf.

3.7.4.6 Arbeiten im 8-bit-Mode

– Datenstations-Ein-/Ausgaben
  Der VTSU-Control-Block (VTSUCB) spezifiziert den CCSN gegenüber VTSU. Wird im 8-bit-Mode mit einer 8-bit-Datenstation gearbeitet, muss der Wert des CCSN im VTSUCB auf den Namen eines 8-bit-Code gesetzt werden, der von der Datenstation unterstützt wird, oder auf Blank, wenn das Kommando

  ```
  /MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=8-BIT-DEFAULT
  ```

  verwendet wird.

– CCSN von Anweisungen
  Der SDF-Makro RDSTMT sollte mit dem CCSNAME durchgeführt werden, der vom SYSFILE-Makro GCCSN zurückgegeben wird. Anweisungen, die im 7-bit-Mode gelesen werden, sind nicht inkompatibel mit einer Verarbeitung im 8-bit-Mode.

– CCSN von Ausgabedateien
  Der entsprechende CCSN wird explizit als Datei-/Elementattribut verwendet. Das CCSN-Attribut von SYSOUT und SYSLST kann über das SYSFILE-Makro GCCSN abgefragt werden.

– Andere Informationen über Codes
  Informationen wie Umsetzung von Klein- in Großbuchstaben, Merkmale von Zeichen oder Sortiersequenzen werden von XHCS versorgt, wenn der NLSCOD-Makro aufgerufen wird.

3.7.4.7 Arbeiten im Unicode-Mode

– Datenstations-Ein-/Ausgaben
  Der VTSU-Control-Block (VTSUCB) spezifiziert den CCSN gegenüber VTSU. Wird im Unicode-Mode mit einer Datenstation im 8-bit-Modus gearbeitet, muss der Wert des CCSN im VTSUCB auf den Namen eines Unicodes gesetzt werden, der von der Datenstation unterstützt wird, oder auf Blank, wenn das Kommando

  ```
  /MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=UTFE
  ```

  verwendet wird.

– CCSN von Anweisungen
  Der SDF-Makro RDSTMT sollte mit dem CCSNAME durchgeführt werden, der vom SYSFILE-Makro GCCSN zurückgegeben wird. Anweisungen, die im 7-bit-Mode gelesen werden, sind nicht inkompatibel mit einer Verarbeitung im Unicode-Mode.

– CCSN von Ausgabedateien
  Der entsprechende CCSN wird explizit als Datei-/Elementattribut verwendet. Das CCSN-Attribut von SYSOUT und SYSLST kann über das SYSFILE-Makro GCCSN abgefragt werden.

– Andere Informationen über Codes
  Informationen wie Umsetzung von Klein- in Großbuchstaben, Merkmale von Zeichen oder Sortiersequenzen werden von XHCS versorgt, wenn der NLSCOD-Makro aufgerufen wird.

3.7.4.8 Arbeiten im 7-bit-Mode

- Datenstations-Ein-/Ausgaben
  Wird im 7-bit-Mode an einer 8-bit-Datenstation gearbeitet, muss der Wert des CCSN im VTSUCB auf „EDF03IRV“ gesetzt werden. Dieser Wert garantiert eine Verarbeitung im 7-bit-Mode von VTSU mit seinen eigenen Tabellen. Der einzige 7-bit-Code für Datenstations-Ein-/Ausgaben ist EDF03IRV; andere 7-bit-Codes werden von VTSU zurückgewiesen.
- CCSN von Ausgabedateien
  Der CCSN von Dateien oder PLAM-Bibliothekselementen muss auf Blank gesetzt werden, wenn deren Inhalt in einem 7-bit-Kontext interpretiert werden muss.
- Andere Informationen über Codes
  Wenn Sie im 7-bit-Mode arbeiten, verwenden Sie interne Tabellen. Das hat die Vorteile, dass die Arbeit unabhängig von XHCS (Verfügbarkeit, Änderung von Tabellen) ist und volle Kompatibilität mit älteren Versionen der Anwendung besteht. Sind allerdings 7-bit-Codes erforderlich, können Sie die XHCS-Tabellen für 7-bit-Codes verwenden.

3.8 Systemsoftware

Der folgende Abschnitt zeigt, welche Systemsoftware ab welcher Version XHCS-fähig ist und was Sie ggf. beachten müssen.

3.8.1 EDT V17.0

EDT kann Dateien in allen Zeichensätzen erzeugen, die von XHCS unterstützt werden.

Der verwendete Code kann explizit (@CODENAME statement) oder implizit (wenn der CCSN der Datei verwendet wird, die eingelesen wird) modifiziert werden.

Eine aktive EDT-Anwendung braucht keine homogene Code-Umgebung eingestellt zu haben. EDT führt immer eine Code-Umsetzung durch und läßt unterschiedliche Zeichensätze zu.

Werden erweiterte Zeichen nicht durch die Datenstation unterstützt, werden diese durch Schmierzeichen angezeigt, z.B. bei der Ausgabe von 8-bit-Dateien auf einer 7-bit-Datenstation.

3.8.2 SHOW-FILE

Dateien oder PLAM-Bibliothekselemente werden vom SHOW-FILE-Kommando korrekt dargestellt, wenn deren CCSN-Attribut einen 8-bit-Code kennzeichnet, der von der Datenstation unterstützt wird. In diesem Fall können erweiterte Zeichen auch in einem Suchstring (FIND statement) eingegeben werden.

Werden erweiterte Zeichen nicht durch die Datenstation unterstützt, werden diese durch Schmierzeichen angezeigt (Unicode erst ab BS2000/OSD V7), z.B. bei der Ausgabe von 8-bit-Dateien auf einer 7-bit-Datenstation.

3.8.3 IFG V8.3

Mit dem IFG können 8-bit-CCS oder Unicode-CCs definiert werden, die zur Erstellung eines Formates verwendet werden. Diese Information wird mit dem Format gespeichert und kann anschließend von FHS zur Ein-/Ausgabeformatierung verwendet werden.

8-bit-Formate können nur auf einer 8-bit-Datenstation generiert, angezeigt und modifiziert werden, außerdem muss die im Benutzerprofil ausgewählte Datenstationsgruppe mit der 8-bit-Datenstation zusammenpassen.

Unicode-Formate können nur auf einer Datenstation im Unicode-Modus generiert, angezeigt und modifiziert werden, außerdem muss die im Benutzerprofil ausgewählte Datenstationsgruppe mit der Datenstation im Unicode-Modus zusammenpassen.

Einige Einschränkungen für 8-bit-Formate oder Unicode-Formate existieren: schnelle Formatierung kann nicht ausgewählt werden und die Formate können nicht mit dem ICE verwendet werden.

3.8.4 FHS V8.3

Die Ein-/Ausgabeformatierung, die von FHS durchgeführt wird, erfordert verschiedene Tabellen (individuelle Zuordnung der Zeichen, Umsetzung von Klein- in Großbuchstaben).

Bis zu drei verschiedene Tabellensätze können von FHS verwendet werden:

– der benutzereigenen Tabellensatz (im MFHSCTAB-Modul)

– der XHCS-Tabellensatz

– der FHS-Standard-Tabellensatz

Im Gegensatz zum benutzereigenen bzw. XHCS-Tabellensatz steht der FHS-Standard-Tabellensatz immer zur Verfügung

Von diesen drei Tabellensätzen hat der benutzereigene Tabellensatz die höchste Priorität. Das bedeutet, gibt es einen benutzereigenen Tabellensatz, wird dieser zur Bearbeitung verwendet. Der XHCS-Tabellensatz wird nur dann verwendet, wenn kein benutzereigener Tabellensatz vorhanden ist. Gibt es weder einen benutzereigenen Tabellensatz noch einen XHCS-Tabellensatz, wird der Standard-Tabellensatz verwendet.

3.8.5 PERCON

PERCON bietet ab V2.5 XHCS-Unterstützung und ab V2.9 Unicode-Unterstützung an.

PERCON ist das BS2000-Programm zur Übertragung und Umsetzung von Dateien. Erweiterte Codes werden von PERCON unterstützt. PERCON berücksichtigt bei der Verarbeitung den Anweisungs-, Eingabe- und Ausgabe-CCS. PERCON setzt auch um, wenn

– der Eingabe-CCS sich vom Ausgabe-CCS unterscheidet. Ein- und Ausgabe-CCS müssen aber kompatibel sein.

– der Anweisungs-CCS sich vom Eingabe-CCS unterscheidet. Werden in der SELECT-INPUT-RECORDS-Anweisung c-strings verwendet, müssen Anweisungs- und Eingabe-CCS derselben Code-Familie angehören und alle Zeichen der c-strings im Eingabe-CCS vorhanden sein.

– der Anweisungs-CCS sich vom Ausgabe-CCS unterscheidet. Werden in den Anweisungen SET-GROUP-ATTRIBUTES, SET-RECORD-MAPPING- oder SET-PAGE-LAYOUT c-strings verwendet, müssen Anweisungs- und Ausgabe-CCS derselben Code-Familie angehören und alle Zeichen der c-strings im Ausgabe-CCS vorhanden sein.

Außerdem können sortierte Dateien auf auf- oder absteigende Reihenfolge, unabhängig vom CCS, überprüft werden.

Neben der Konvertierung kann PERCON ab V2.9 für UTF-16 auch die Normalisierungsfunktion nutzen.

3.8.6 SORT

SORT bietet ab V7.4 XHCS-Unterstützung und ab V7.9 Unicode-Unterstützung an.

SORT kann seine Verarbeitung auf die Sortiersequenzen stützen, die in den XHCS-Tabellen verschlüsselt sind. Das bedeutet, dass die Sortierung nicht auf der Darstellung der Zeichen oder auf SORT-internen Daten basiert. Die Voraussetzungen dafür sind:

– es muss das EXTENDED-CHARACTER-Format gewählt werden (im SORT-RECORDS-Statement)

– Dateien, die sortiert werden sollen, haben einen CCSN der einen vorhandenen 8-bit-CCS bezeichnet

Beachten Sie, dass die XHCS-Sortiergewicht-Tabellen - in denen jedem Zeichen eine Position in der Sortiersequenz zugeordnet ist - eindeutig sein müssen; jedes Sortiergewicht erscheint nur einmal in einer Tabelle. Die Konsequenz daraus ist, dass zwei Zeichen nicht gleichwertig sein können bezüglich des Sortierprozesses. Der Daten-CCS muss kein 8-bit-CCS sein.

Ab SORT V7.9 kann SORT die von XHCS bereitgestellte Sortiertabelle für die Unicode-Variante UTF-16 nutzen. Voraussetzung dafür ist, dass Sie das Format UNICODE-CHARACTER in der SORT-Anweisung wählen.

3.8.7 RSO V3.5

RSO verwendet den codierten Zeichensatznamen der Datei, die gedruckt werden soll, um die Umsetzung auf den Dateiinhalt durchzuführen. RSO fordert die geeignete Umsetzungstabelle von XHCS an, vorausgesetzt der Remote-Drucker ist als Typ „8-bit“ in der RSO-Konfigurationsdatei definiert und der CCSN ist weder leer noch EDF03IRV. Beachten Sie, dass die Auswahl des Druckerzeichensatzes und des Datei-CCSN zwei völlig unterschiedliche Aspekte sind.

3.8.8 LMS V3.3B

Genau wie Dateien haben auch Bibliothekselemente ein CCSN-Attribut. Wird eine Datei in ein PLAM-Bibliothek übertragen bleibt der CCSN der Datei erhalten, umgekehrt erzeugt ein ausgewähltes Element eine Datei mit dem CCSN des Elements. Der CCSN bleibt auch dann erhalten, wenn Elemente dupliziert werden.

Codierte Zeichensatznamen können auch explizit gesetzt werden mithilfe des Statements `MODIFY-ELEMENT-ATTRIBUTES`. Den CCSN eines Elementes erhält man mit den Operanden `SHOW-ELEMENT-ATTRIBUTES` und `INFORMATION=MAXIMUM`.

3.8.9 FT-BS2000 V5.0

Werden Dateien zwischen zwei BS2000-Systemen übertragen, wird das CCSN-Attribut von FT-BS2000 behalten. Für den Austausch von Textdateien mit FTAM-Partnern, setzt FT-BS2000 die Zeichen um unter Verwendung der EBCDIC.DF.04/ISO 8859-Umsetztabelle.

FT-BS2000 prüft, ob der übermittelte CCSN im Empfangssystem vorhanden und mit dem Referenzcode kompatibel ist. Ist das der Fall, werden die Empfangsdaten gemäß der Code-Tabelle konvertiert. Das CCSN-Attribut wird in der Empfangsdatei hinterlegt.

3.8.10 OMNIS V6.3

Um im 8-bit-Mode zu arbeiten, ist OMNIS ab V6.3 erforderlich. Wird OMNIS verwendet, um eine Verbindung zu einem Remote-Prozessor herzustellen, muss Folgendes berücksichtigt werden:

– die verwendeten codierten Zeichensätze müssen in beiden Systemen definiert sein (gleicher Name, selber Inhalt)

– beide VTSU-Versionen (Version 10.0 oder größer) müssen XHCS unterstützen

– der permanente 8-bit-Mode ist nur möglich mit VTSU ab V11.0 auf beiden Seiten.

3.8.11 Beispiele

Beispiel 1: Umsetzen einer Datei von einem Zeichensatz in einen anderen

Das folgende Beispiel zeigt, wie Sie mit dem Programm PERCON den Inhalt einer Datei von einem Zeichensatz in einen anderen umsetzen können. Die Eingabedatei, in diesem Beispiel PERCON.IN, hat das Zeichensatzattribut entsprechend ihrem Inhalt, z.B. EDF03IRV.

1. Legen Sie eine leere Ausgabedatei mit dem gewünschten CCS an:

```
/CREATE-FILE PERCON.OUT,C-C-S=EDF041
/SET-FILE-LINK LINK-NAME=POUT,FILE-NAME=PERCON.OUT,ACCESS-METHOD=SAM
```

2. Rufen Sie PERCON auf:

```
/START-PERCON
```

```
//ASS-INPUT-FILE FILE=DISK-FILE(NAME=PERCON.IN),LINK-NAME=PIN
```

weist die Eingabedatei zu

```
//ASS-OUTPUT-FILE LINK-NAME=POUT
```

weist die Ausgabedatei zu

```
//END
```

startet die Umsetzung und beendet das Programm

Beispiel 2: Sortieren einer Datei unter Verwendung der XHCS-Tabellen

Das Programm SORT kann zum Sortieren die XHCS-Tabellen verwenden, wie das folgende Beispiel zeigt.

```
/START-SORT
```

startet das SORT-Programm

```
//ASSIGN-FILES INPUT-FILE=SORT.8.IN,OUTPUT-FILE=SORT.8.OUT
```

weist die Ein- und Ausgabedatei zu

```
//SORT-RECORDS FIELDS=FIELD-EXPLICIT(5,10,FORMAT=EXTENDED-CHARACTER)
```

die Verwendung der XHCS-Tabellen muss explizit mit dem Formatparameter angegeben werden

```
//END
```

4 XHCS-Makros

XHCS stellt drei Assemblermakros zur Verfügung, die folgende Funktionen unterstützen:

– der Makro NLSCOD liefert unterschiedliche Informationen über den Ausgangszeichensatz. Die Informationen sind vom TABLE-Parameter abhängig. Je nach Wert des TABLE-Parameters wird Ihnen eine Tabelle zur Umwandlung zwischen Code-Sets, zur Umwandlung von Klein- in Großbuchstaben oder umgekehrt und zur Festlegung der Sortierfolge von Zeichen oder von Zeicheneigenschaften geliefert. Für UTFE ist die Ausgabe einer Byte-Eigenschaftstabelle möglich.

– der Makro NLSCMP informiert über die Code-Kompatibilität. Die Information ist von der angegebenen ISO-Code-Variantennummer oder vom angegebenen Code-Namen abhängig. Je nach Angabe wird Ihnen eine Liste kompatibler Codes geliefert.

– der Makro NLSCNV wandelt Zeichenketten direkt um. Hier angegebene Zeichenketten werden direkt in einen kompatiblen Zielcode umgewandelt. Außerdem setzt er Klein- in Großbuchstaben um und umgekehrt, gibt die Länge von Unicode-Zeichenketten aus und normalisiert Eingabezeichenketten. Eine eindeutig umkehrbare Konversion kann mithilfe der entsprechenden Ersatzzeichenbehandlung für 7-/8-bit-Zeichensätze sichergestellt werden. Zudem wird ein SVC-freier Ansprung der Funktionen von NLSCNV aus TU ermöglicht.

An den Schnittstellen NLSCNV und NLSCMP haben Sie die Möglichkeit, die Ein- und Ausgabezeichenketten nicht mehr nur im V-Format anzugeben bzw. zu empfangen, sondern die Zeichenketten über eine Adresse und eine Länge zu handhaben.

4.1 Code-Information: NLSCOD

4.1.1 Makroaufrufformat

Operation	Operanden	
NLSCOD	MF=D	[,PREFIX=<prefix>]
	MF=C	[,PREFIX=<prefix>][,MACID=<macid>]
	MF=E	,PARAM=<paramlist> / (register)
	MF=L / **S**	,CCSNAME= <7-/8-bit-Code> / UNICODE / *SYSDEF / *USRDEF ,TABLE= CONVERT,TOCCS= <7-/8-bit-Code>/UNICODE / *SYSDEF / *USRDEF ,TABADDR= <addr> ,TABLEN= <addr>
	MF=L / **S**	,CCSNAME= <7-/8-bit-Code> / UNICODE / UTFE / *SYSDEF / *USRDEF ,TABLE= TOUPPER / TOLOWER / SORT / INFORMATION (<prop1>,...,<prop8>) ,TABADDR= <addr> ,TABLEN= <addr>
	MF=L / **S**	,CCSNAME= UTFE ,TABLE= (<<utfeprop1>,...,<utfeprop8>) ,TABADDR= <addr> ,TABLEN= <addr>

4.1.2 Operandenbeschreibung

MF

=D Generiert eine DSECT.

=E Generiert einen Befehlsteil. Es werden keine Befehle zur Auswertung des Returncodes erzeugt.

=C Generiert das Layout des NLSCOD in der aktuellen Datenstruktur. Dabei wird jedes Feld benannt und es werden alle Equates abgesetzt. Der Standard-Header wird nicht versorgt.

=S	Generiert eine Parameterliste. Die Felder werden gemäß den angegebenen Schlüsselwortparametern des Makros versorgt. Namen und Equates werden nicht abgesetzt. Zusätzlich wird ein Befehlsteil generiert. Der Standard-Header wird automatisch versorgt. Es werden keine Befehle zur Auswertung des Returncodes erzeugt.
=L	Generiert eine Parameterliste. Die Felder werden gemäß den angegebenen Schlüsselwortparametern des Makros versorgt. Namen und Equates werden nicht abgesetzt. Der Standard-Header wird automatisch versorgt.
PREFIX	
=x	Gibt das erste Zeichen an, das den Namen vorangestellt wird, die mit MF=D oder MF=C definiert werden. Der Standardwert ist PREFIX=G.
MACID	
=xxx	Gibt das 2.-4. Zeichen der Namen an, die mit MF=C definiert werden. Der Standardwert ist MACID=NLT (auch für MF=D).
PARAM	
=paramlist	Gibt den Namen der Parameterliste an.
=(register)	Enthält die Adresse der Parameterliste.
CCSNAME	Der Name des Ausgangscodes wird festgelegt. Der Name ist max. 8 Byte lang. Wird kein Name angegeben, wird automatisch der EDF03IRV als Ausgangscode angenommen. Als Unicode-Varianten können nur UNICODE oder UTFE angegeben werden. Die Angaben UTF16 und UTF8 sind an dieser Schnittstelle nicht erlaubt.
=<7-/8-bit-Code>	Name des 7-/8-bit-Codes.
=UNICODE	Der Unicode-Zeichensatz UTF16 ist der Ausgangscode.
=UTFE	Der Unicode-Zeichensatz UTFE ist der Ausgangscode.
=*SYSDEF	Der System-Standard-Code ist der Ausgangscode.
=*USRDEF	Der Standard-Anwender-Code ist der Ausgangscode.

TABLE — Art der zu liefernden Tabelle.

=CONVERT — Tabelle zur Umwandlung des Ausgangscodes (CCSNAME) in den Zielcode (TOCCS).

CCSNAME = 7-/8-bit-Code und TOCCS = 7-/8-bit-Code:
Ausgabe einer 256 Byte-Umwandlungstabelle

CCSNAME = 7-/8-bit-Code und TOCCS = UNICODE:
Es wird eine 2*256 Byte-Tabelle ausgegeben, die jedem Code-Punkt der 8-bit-Tabelle den entsprechenden Wert in der Unicode-Tabelle zuordnet.

CCSNAME = UNICODE und TOCCS = 7-/8-bit-Code:
Es wird eine 64 K-Tabelle ausgegeben, die jedem Punkt der Unicode-Tabelle den entsprechenden Wert in der angegebenen 8-bit-Tabelle zuordnet, falls definiert, und 0, falls nicht definiert.

CCSNAME = UNICODE und TOCCS = UNICODE:
Diese Kombination ist nicht erlaubt.

CCSNAME=UTFE:
Diese Angabe ist nicht erlaubt.

=TOUPPER — Tabelle zur Umwandlung von Kleinbuchstaben in die entsprechenden Großbuchstaben.

CCSNAME =7-/8-bit-Code:
Jedem Kleinbuchstaben wird der entsprechende Großbuchstabe zugeordnet. Alle anderen Positionen werden mit der identischen Abbildung versorgt.

CCSNAME = UNICODE:
Jedem Kleinbuchstaben der Unicode-Tabelle wird der entsprechenden Großbuchstabe zugeordnet.
Alle anderen Unicode-Positionen werden mit der identischen Abbildung versorgt.
Die ausgegebene Tabelle ist 64K*2 groß.

CCSNAME=UTFE:
Jedem Kleinbuchstaben, der in 1 Byte codiert ist, wird der entsprechende Großbuchstabe zugeordnet.
Alle anderen Positionen werden mit der identischen Abbildung versorgt.
Die ausgegebene Tabelle ist 256 Byte groß.

=TOLOWER

Tabelle zur Umwandlung von Großbuchstaben in die entsprechenden Kleinbuchstaben.

CCSNAME = 7-/8-bit-Code:
Jedem Großbuchstaben wird der entsprechende Kleinbuchstabe zugeordnet. Alle anderen Positionen werden mit der identischen Abbildung versorgt.

CCSNAME = UNICODE:
Jedem Großbuchstaben der Unicode-Tabelle wird der entsprechende Kleinbuchstabe zugeordnet.
Alle anderen Unicode-Positionen werden mit der identischen Abbildung versorgt.
Die ausgegebene Tabelle ist 64K*2 groß.

CCSNAME=UTFE:
Jedem Großbuchstaben, der in 1 Byte codiert ist, wird der entsprechende Kleinbuchstabe zugeordnet.
Alle anderen Positionen werden mit der identischen Abbildung versorgt.
Die ausgegebene Tabelle ist 256 Byte groß.

=SORT

Tabelle, die die Sortierreihenfolge angibt. Diese Tabelle kann mit dem Dienstprogramm SORT zum Sortieren von Feldern (Format=EXTENDED-CHARACTER bei nicht Unicode, Format=UNICODE-CHARACTER bei CCSN=UNICODE) verwendet werden (siehe Handbuch „SORT V7.9A (BS2000/OSD)“ [28]).

CCSNAME = 7-/8-bit-Code:
Ausgabe einer 256 Byte-Sortiertabelle für 7-/8-bit-Codes

CCSNAME = UNICODE:
Informationen zum Unicode Collation Algorithm finden Sie unter http://www.unicode.org/reports/tr10/.
Aus der Datei
http://www.unicode.org/Public/UCA/4.0.0/allkeys-4.0.0.txt, die eine Standard-Collation-Tabelle darstellt, wurde der Teil herausgefiltert, der den von XHCS unterstützten Zeichen entspricht.

Für jeden Code-Punkt bis zum größten unterstützten Wert gibt es je ein Collation-Element der Länge 8. Zu nicht genutzten Code-Punkten gehören Elemente mit der Belegung X'00..00'.

Die einzelnen Tabelleneinträge haben folgende Form (Nähere Informationen zu variablen Collation-Elementen und Sortiergewicht entnehmen Sie dem Handbuch „SORT V7.9A (BS2000/OSD)“ [28]:

Byte 1-2	höchstwertiges Bit	1: variables Collation-Element 0: andernfalls
	Restliche 15 Bits	Sortiergewicht für Level 1 von Element 1
Byte 3-4	Byte 2	Sortiergewicht für Level 2 von Element 1
	Byte 3	Sortiergewicht für Level 3 von Element 1
Byte 5-6	Sortiergewicht für Level 4 von Element 1	
Byte 7-8	0 oder Verweis auf ein eventuell vorhandenes 2. Collation-Element: Nummer des Eintrags des 2. Collation-Elements. Der Abstand dieses Elements zum Tabellenanfang ergibt sich aus der Multiplikation dieser Nummer mit 8.	

Das zweite Collation-Element ist aufgebaut wie das erste, es kann auch auf ein drittes Element zeigen. Das dritte Element ist genauso aufgebaut, hat aber immer einen Wert X'0000' in seinem Zeiger.

Die Tabelle besteht zunächst aus X’3000’=12288 Einträgen je 8 Bytes, was den maximal möglichen Unicode-Positionen von 0 bis 2FFF entspricht.

Die Nummer eines Eintrags entspricht seiner Unicode-Position. Die Einträge sind belegt mit dem ersten Collation-Element eines Collation Entries.

Hinter diesem Bereich der Größe X’3000’*8 = X’24000’ Bytes beginnt der Bereich der 2. und 3. Collation-Elemente. Dieser hat die Größe von X’2000’ Bytes und somit Platz für 1024 (=X’2000’ / 8) Einträge von 2. und 3. Collation-Elementen.

CCSNAME=UTFE:
Diese Angabe ist nicht erlaubt.

=INFORMATION

CCSNAME = 7-/8-bit-Code:
Diese Angabe ist nicht erlaubt.

CCSNAME = UNICODE:
Gibt drei Worte mit folgenden Informationen aus:

1. Wort:
   Höchste, von XHCS unterstützte Unicode-Position. Alle Unicode-Tabellen sind ab dieser Position leer. Die höchste, unterstützte Unicode-Position ist zurzeit 12287 dezimal (= 2FFF hexadezimal).

2. Wort:
   Größe der Sortierinformation für Unicode:
   zurzeit 106496 dezimal (= 1A000 hexadezimal).

3. Wort:
   Größe der Normalisierungstabelle:
   zurzeit 2777736 dezimal (= 2A6288 hexadezimal)

CCSNAME=UTFE:
Diese Angabe ist nicht erlaubt.

=(<prop1>,...,<prop8>)

Es wird eine Tabelle mit Eigenschaften geliefert.

Die Tabelle ist wie folgt aufgebaut: Das i-te Bit eines jeden Byte der Tabelle wird auf binär 1 gesetzt, wenn für das entsprechende Zeichen die i-te Eigenschaft der Liste zutrifft. Maximal können 8 Eigenschaften in beliebiger Reihenfolge angegeben werden. Die Eigenschaft, die am weitesten rechts steht, wird auf das niederwertigste Bit abgebildet. Werden weniger als 8 Eigenschaften angegebenen, werden die Bytes von links mit binären Nullen aufgefüllt.

Folgende Eigenschaften sind möglich:

LETTER: Das Zeichen ist ein Buchstabe.

DIGIT: Das Zeichen ist eine Ziffer.

ARITH: Das Zeichen ist ein arithmetisches Zeichen.

SPECIAL: Das Zeichen ist ein Sonderzeichen.

LOWER: Das Zeichen ist ein Kleinbuchstabe.

KERNEL: Das Zeichen gehört zum EBCDIC-Kern.

DEFINED: Das Zeichen ist im CODE definiert.

DISPL: Das Zeichen ist darstellbar/druckbar.

TRUE: Es wird immer binär 1 geliefert.

FALSE: Es wird immer binär 0 geliefert.

CCSNAME = 7-/8-bit-Code:
Ausgabe einer 256 Byte-Eigenschaftstabelle

CCSNAME = UNICODE:
Die Eigenschaftstabelle für Unicode ist im Tabellenmodul GNLMTAB definiert.
Für jeden Code-Punkt bis zum größten unterstützten Wert gibt es 1 Byte in der Tabelle.
Die ausgegebene Tabelle ist 64K groß.

CCSNAME = UTFE:
Diese Angabe ist nicht erlaubt.

Beispiel 1

Es soll überprüft werden, ob ein Zeichen die Eigenschaften DISPL, DEFINED, KERNEL und ARITH erfüllt. Die Eigenschaft ARITH soll auf das niederwertigste Bit abgebildet werden. Die restlichen Eigenschaften (LETTER, SPECIAL, DIGIT, LOWER) des Zeichens werden nicht abgefragt. Der Code ist EBCDIC.DF.04-1.

Aufbau der Parameterliste:

```
NLSCOD MF=L,TABLE=(DISPL,DEFINED,KERNEL,ARITH),
       CCSNAME=EDF041,TABADDR=PROPTAB
```

Ergebnis für das Zeichen '?' (X'6F'):

Die Eigenschaft ARITH trifft auf dieses Zeichen nicht zu.

Darstellung:

Bit	0	0	0	0	1	1	1	0
Eigenschaften					DISPL	DEFINED	KERNEL	ARITH

Beispiel 2

Es soll überprüft werden, ob ein Zeichen die Eigenschaften DISPL, DEFINED, KERNEL und ARITH erfüllt. Die restlichen Bits werden auf '1' gesetzt.

Aufbau der Parameterliste:

```
NLSCOD MF=L,TABLE=(DISPL,DEFINED,KERNEL,
ARITH,TRUE,TRUE,TRUE,TRUE),
CCSNAME=EDF041,TABADDR=PROPTAB
```

Ergebnis für das Zeichen '?' (X'6F'):

Die Eigenschaft ARITH trifft auf dieses Zeichen nicht zu.

Darstellung:

Bit	1	1	1	0	1	1	1	1
Eigenschaften	DISPL	DEFINED	KERNEL	ARITH	restl. Bits			

=(<utfeprop1>,...,<utfeprop8>)

Ausgabe einer 256 Byte großen Tabelle mit den Byte-Eigenschaften für UTFE.

Die Tabelle ist wie folgt aufgebaut:

Das i-te Bit eines jeden Byte der Tabelle wird auf binär 1 gesetzt, wenn für das entsprechende Byte die i-te Eigenschaft der Liste zutrifft. Es können bis zu acht Eigenschaften in beliebiger Reihenfolge angegeben werden. Die Eigenschaft, die am weitesten rechts steht, wird auf das niederwertigste Bit abgebildet. Wenn weniger als acht Eigenschaften angegeben werden, werden die Bytes von links mit binär Nullen aufgefüllt.

Folgende Eigenschaften sind möglich:

UTFE1	Das Byte ist das erste Byte einer Ein-Byte-Codierung
UTFE2	Das Byte ist das erste Byte einer Zwei-Bytes-Codierung
UTFE3	Das Byte ist das erste Byte einer Drei-Bytes-Codierung
UTFE4	Das Byte ist das erste Byte einer Vier-Bytes-Codierung
UTFE5	Das Byte ist das erste Byte einer Fünf-Bytes-Codierung
UTFEF	Das Byte ist ein Folgebyte einer Mehr-Bytes-Codierung
UTFEC	Das Byte ist ein Kontrollzeichen (Ein-Byte-Codierung)
NONUTFE	Das Byte ist kein zulässiges UTFE-Byte

CCSNAME = 7-/8-bit-Code:
Diese Angabe ist nicht erlaubt.

CCSNAME = UNICODE:
Diese Angabe ist nicht erlaubt.

CCSNAME=UTFE:
Ausgabe einer 256 Byte-Eigenschaftstabelle.

TOCCS	Der Name des Zielcodes wird festgelegt. Der Name ist 8 Byte lang. Wird kein Name angegeben, wird automatisch der EDF03IRV als Zielcode angenommen. Die Angabe ist nur für TABLE=CONVERT sinnvoll.
=<7-/8-bit-Code>	Name des 7-/8-bit-Codes
=UNICODE	Ein Unicode-Zeichensatz ist der Zielcode.
TABADDR	Anfangsadresse der Ergebnistabelle
TABLEN	Adresse eines Wortes, das nach erfolgreicher Bearbeitung der beim Operanden TABLE angegebenen Funktion die Länge der gelieferten Tabelle enthält.

4.1.3 Funktionsübersicht

Der Makro NLSCOD (NATIONAL LANGUAGE SUPPORT: CODE TABLES) wird über einen Supervisor-Call (SVC) aufgerufen. NLSCOD liefert Tabellen verschiedener Größe.

Die Informationen sind vom Parameter TABLE abhängig:

- Wenn sowohl der Ausgangszeichensatz (CCSNAME) als auch der Zielcode (TOCCS) ein 7-/8-bit-Code ist, so gilt:

  Die gelieferte Tabelle hat eine Länge von 256 Bytes. Das n-te Byte der gelieferten Tabelle enthält Informationen über das Zeichen des Ausgangszeichensatzes, dessen Wert "n" ist.

  – Ist TABLE=CONVERT, enthält das Byte den Zielcode (TOCCS) für dieses Zeichen. Ausgangs- und Zielcode müssen kompatibel sein.

  – Ist TABLE=TOUPPER, enthält das Byte die Position des entsprechenden Großbuchstaben im Ausgangszeichensatz (CCSNAME).

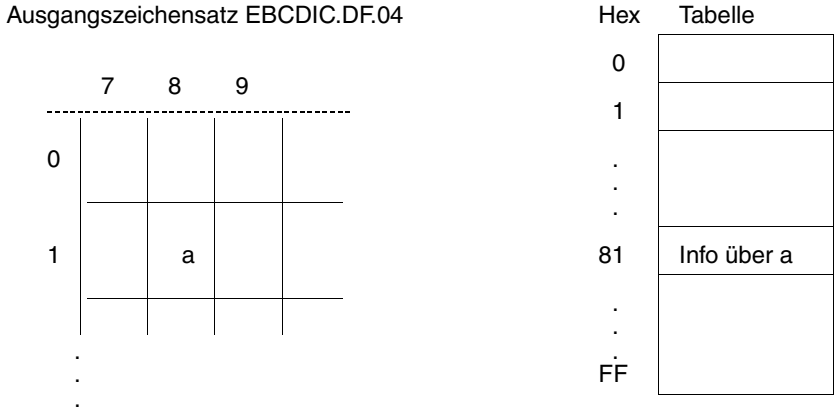
  – Ist TABLE=TOLOWER, enthält das Byte die Position des entsprechenden Kleinbuchstaben im Ausgangszeichensatz (CCSNAME).

  – Ist TABLE=SORT, enthält das Byte die Sortierfolgenummer des Zeichens.

  – Bei einer Liste von Eigenschaften (<prop>), ist das Byte als geordnete Folge von Bits zu verstehen, bei der jedes Bit eine Eigenschaft darstellt.
    Binär 1 bedeutet „Eigenschaft trifft zu“, binär 0 bedeutet „Eigenschaft trifft nicht zu“.
    Es können bis zu 8 Eigenschaften angegeben werden.

  Beispiel

  Der Buchstabe a hat im EBCDIC.DF.04 den Wert X’81’. Das heißt, das Byte X’81’ dieser Tabelle enthält Informationen über den Buchstaben a des Ausgangszeichensatz.



- Wenn der Ausgangszeichensatz (CCSNAME) oder der Zielcode (TOCCS) gleich UNICODE ist, hängen Größe und Aufbau der gelieferten Tabelle vom Parameter TABLE ab.

  Näheres hierzu siehe bei der Beschreibung der einzelnen Werte des Parameters TABLE.

- Wenn der Ausgangszeichensatz (CCSNAME) UTFE ist, werden folgende Werte für TABLE unterstützt:

  – die Byte-Eigenschaftstabelle (<utfeprop>)

    Die gelieferte Tabelle hat eine Länge von 256 Byte. Sie beinhaltet für jedes Byte die Byte-Eigenschaften. Folgende Eigenschaften sind möglich:

    – Das Byte ist das erste Byte einer Ein-Byte-Codierung.
    – Das Byte ist das erste Byte einer Mehr-Bytes-Codierung (Zwei-, Drei-, Vier- oder Fünf-Bytes-Codierung).
    – Das Byte ist ein Folgebyte einer Mehr-Bytes-Codierung.
    – Das Byte ist ein Kontrollzeichen (Ein-Byte-Codierung).
    – Das Byte ist kein zulässiges UTFE-Byte.

  – TOUPPER / TOLOWER für die Ein-Byte-Codierungen

    Die gelieferte Tabelle hat eine Länge von 256 Byte.
    Sie beinhaltet für jede Ein-Byte-Codierung, die einem Kleinbuchstaben entspricht, den entsprechenden Großbuchstaben (oder umgekehrt). Alle anderen Abbildungen bleiben identisch.

4.1.4 Returncodes im Standard-Header

SUBCODE 2 | 1		MAINCODE 2 1	Bedeutung
X' 00'	X' 00'	X' 00' X' 00'	erfolgreiche Bearbeitung; gewünschte Tabelle verfügbar
X' 00'	X' 00'	X' 00' X' 24'	erfolgreiche Bearbeitung; gewünschte Umwandlungstabelle verfügbar Achtung: Einige Zeichen des Ausgangscodes haben kein Äquivalent im Zielcode
X' 00'	X' 01'	X' 00' X' 04'	Name des Codes (CCSN) unbekannt
X' 00'	X' 01'	X' 00' X' 14'	inkonsistente Parameterliste
X' 00'	X' 01'	X' 00' X' 20'	Ausgangscode (CCSNAME) und Zielcode (TOCCS) sind inkompatibel
X' 00'	X' 01'	X' 00' X' 40'	Eigenschaft ungültig
X' 00'	X' 01'	X' 00' X' 44'	Tabelle ungültig
X' 00'	X' 01'	X' 00' X' 80'	Fehler bei der Speicherzuordnung
X' 00'	X' 01'	X' FF' X' FF'	Die angeforderte Funktion wird nicht unterstützt. Nicht behebbarer Fehler.
X' 00'	X' 02'	X' FF' X' FF'	Die angeforderte Funktion ist nicht verfügbar. Nicht behebbarer Fehler. Mögliche Ursache: XHCS-SYS fehlt in der aktuellen Konfiguration
X' 00'	X' 03'	X' FF' X' FF'	Die angegebene Version der Schnittstelle wird nicht unterstützt (falsche Versionsangabe im Standard-Header). Nicht behebbarer Fehler.
X' 00'	X' 04'	X' FF' X' FF'	Parameterliste ist nicht auf Wortgrenze ausgerichtet.
X' 00'	X' 20'	X' 00' X' C0'	fehlerhafte Code-Tabellen
X' 00'	X' 41'	X' FF' X' FF'	Das Subsystem ist nicht vorhanden; es muss explizit erzeugt werden.
X' 00'	X' 42'	X' FF' X' FF'	Der aufrufende Ablauf ist mit dieser Schnittstelle nicht konnektiert; er muss explizit konnektiert werden.
X' 00'	X' 81'	X' FF' X' FF'	Subsystem zurzeit nicht verfügbar.
X' 00'	X' 82'	X' FF' X' FF'	Subsystem im DELETE- oder HOLD-Zustand.

Bedeutung der einzelnen Felder:

SUBCODE 1	Anzeigen der Fehlerklasse
= X'00':	Erfolgreiche Bearbeitung
≠ X'00':	Fehler
SUBCODE 2	wird immer auf den Wert X'00' gesetzt.
MAINCODE 1	Fehlerinformationen für das Anwendungsprogramm. Das Anwendungsprogramm kann damit Bedienungsfehler abfangen.
MAINCODE 2	wird zurzeit nicht belegt und auf den Wert X'00' gesetzt. Wenn im Standard-Header Fehler sind, die der Selbstidentifikation des Produktes dienen (z.B. falsche Version), wird der Wert auf X'FF' gesetzt.

Bei MAINCODE X'0024' müssen Sie die Umwandlungstabelle mit besonderer Sorgfalt verwenden, da einige Zeichen aus dem Ausgangscode (z.B. Buchstaben mit Akzent) im Zielcode nicht vorkommen. Falls die Tabellen ordnungsgemäß definiert sind, ist der zugehörige Wert in der Umwandlungstabelle X'00' (NIL).

Beispiel

```
EXMPL    START
*
STRUCT   NLSCOD MF=D ---------------------------------------------(01)
*
EXMPL    CSECT
*
         BALR   10,0
         USING  *,10
         USING  STRUCT,11
         LA     11,PL
*
         NLSCOD MF=E,PARAM=PL ---------------------------------(02)
*
* Modifizieren der Parameterliste
         MVC    GNLTCCSN,=CL8'ISO88591' ---------------------------(03)
         LA     5,LOWUP2 ------------------------------------------(04)
         ST     5,GNLTTADD
*
         NLSCOD MF=E,PARAM=PL ---------------------------------(05)
*
         TERM
*
PL       NLSCOD MF=L,CCSNAME=EDF041,TABLE=TOUPPER,TABADDR=LOWUP1 --(06)
*
         DS     0D
LOWUP1   DS     XL256 ----------------------------------------------(07)
LOWUP2   DS     XL256 ----------------------------------------------(08)
*
         END
```

(01) Die Struktur der Parameterliste wird definiert (DSECT).

(02) XHCS wird mit einer von MF=L initialisierten Parameterliste aufgerufen.

(03) Der Code-Name wird modifiziert.

(04) Die Tabellenadresse wird modifiziert.

(05) XHCS wird mit der modifizierten Parameterliste aufgerufen.

(06) Die Parameterliste wird deklariert und initialisiert. Durch die Initialisierung wird die Tabelle zur Umwandlung von Klein- in Großbuchstaben für die Zeichensatztabelle EBCDIC.DF.04-1 angefordert.

(07) Tabelle Nr.1

(08) Tabelle Nr.2

4.2 Kompatible Codes: NLSCMP

4.2.1 Makroaufrufformat

Operation	Operanden	
NLSCMP	MF=D	[,PREFIX= <prefix>]
	MF=C	[,PREFIX= <prefix>][,MACID= <macid>]
	MF=E	,PARAM= <paramlist> / (register)
	MF=L / S	,CCSNAME= <7-/8-bit-Code> / <unicodevariante> / *SYSDEF / *USRDEF [,INFO= *LIST-OF-CCS / *ONLY-SPECIFIED-CCS / *REFERENCE-CODE] ,LSTADDR= <addr> [,LSTLEN= <addr>]
	MF=L / S	,ISOVAR= <isovariante> / *ALL [,INFO= *LIST-OF-CCS / *REFERENCE-CODE] ,LSTADDR= <addr> [,LSTLEN= <addr>]

4.2.2 Operandenbeschreibung

MF

=D Generiert eine DSECT.

=E Generiert einen Befehlsteil. Es werden keine Befehle zur Auswertung des Returncodes erzeugt.

=C Generiert das Layout des NLSCMP in der aktuellen Datenstruktur. Dabei wird jedes Feld benannt und es werden alle Equates abgesetzt. Der Standard-Header wird nicht versorgt.

=S Generiert eine Parameterliste. Die Felder werden gemäß den angegebenen Schlüsselwortparametern des Makros versorgt. Namen und Equates werden nicht abgesetzt. Zusätzlich wird ein Befehlsteil generiert. Der Standard-Header wird automatisch versorgt. Es werden keine Befehle zur Auswertung des Returncodes erzeugt.

=L Generiert eine Parameterliste. Die Felder werden gemäß den angegebenen Schlüsselwortparametern des Makros versorgt. Namen und Equates werden nicht abgesetzt. Der Standard-Header wird automatisch versorgt.

PREFIX

=x — Gibt das erste Zeichen an, das den Namen vorangestellt wird, die mit MF=D oder MF=C definiert werden.
Der Standardwert ist PREFIX=G.

MACID

=xxx — Gibt das 2.-4. Zeichen der Namen an, die mit MF=C definiert werden. Der Standardwert ist MACID=NLI (auch für MF=D).

PARAM

=<paramlist> — Gibt den Namen der Parameterliste an.

=(register) — Enthält die Adresse der Parameterliste.

CCSNAME — Festlegung des Code-Namens. Der Name ist max. 8 Byte lang. Wird kein Name angegeben, wird automatisch der EDF03IRV benutzt.

=<7-/8-bit-Code> — Name eines 7-bit- oder 8-bit-Codes.

=<unicodevariante> — Name des Unicode-Ausgangscodes.
Als Unicode-Variante können Sie UNICODE, UTF16 und UTF8 oder UTFE für UTF-EBCDIC angeben. Die Angabe von 'UNICODE' ist gleichbedeutend mit UTF16. Die ausgegebene Liste enthält die Namen aller im System definierten 7-bit- oder 8-bit-Codes einschließlich der drei Unicode-Varianten UTF-16, UTF-8 und UTFE.

=*SYSDEF — Der System-Standard-Code wird benutzt.

=*USRDEF — Der Standard-Anwender-Code wird benutzt.

ISOVAR

=<isovariante> — Gibt die Nummer einer ISO-Code-Variante an, zu der eine Liste kompatibler Codes geliefert wird. Die Nummer muss eine ganze Zahl sein.

=*ALL — Es werden die Namen aller im System definierten 7-bit- und 8-bit-Codes ausgegeben einschließlich der drei Unicode-Varianten UTF-16, UTF-8 und UTFE.

INFO Art und Umfang der Ausgabeinformation.

Jeder Eintrag zur Ausgabeinformation hat folgende Struktur:

Byte	0 - 7	Code-Name (CCS)
Byte	8	' ', falls der Code mit der entsprechenden ISO-Code-Variante ISO 8859 voll kompatibel ist. 'P' , falls der Code begrenzt kompatibel ist.
Byte	9	'I' (ISO) oder 'E' (EBCDIC), je nach der Art des Codes. Handelt es sich um den Eintrag einer Unicode-Variante, so wird Byte 9 folgendermaßen belegt (siehe auch Seite 85): UTF16 'I' UTF8 'I' UTFE 'E'

Wird der Operand INFO nicht angegeben, wird der Standardwert *LIST-OF-CCS angenommen.

=*LIST-OF-CCS Es wird eine Liste kompatibler Codes geliefert. Pro Code wird ein Eintrag ausgegeben.

Mit diesem Operandenwert erhält man das Schnittstellenverhalten wie in XHCS V1.5.

=*ONLY-SPECIFIED-CCS
Es wird der Eintrag des angegebenen CCS ausgegeben.
Der Operand INFO=*ONLY-SPECIFIED-CCS ist nur in Kombination mit dem Operanden CCSNAME erlaubt.

=*REFERENCE-CODE
Bei Angabe des Operanden ISOVAR wird der Referenzcode der angegebenen ISO-Code-Variante ausgegeben.
ISOVAR = *ALL ist nicht erlaubt. Bei ISOVAR = *ALL wird der Returncode X'00010014' gesetzt.

Bei Angabe des Operanden CCSNAME wird der Referenzcode der ISO-Code-Variante ausgegeben, zu der der angegebene CCS gehört.

Bei CCSNAME = <unicodevariante> wird UTFE als Referenzcode ausgegeben.

LSTADDR

Adresse der Ausgabeliste.
Diese Adresse muss auf Halbwortgrenze beginnen.

Die Struktur der Ausgabeliste hängt von der Angabe beim Operanden LSTLEN ab:

LSTLEN ist nicht angegeben:
Die Ausgabeliste wird im V-Format geliefert:

Byte 0 - 1	Gesamtlänge der Zeichenkette (einschließlich Header). Vor dem Aufruf muss hier die maximale Länge der Ausgabeliste (einschließlich Header) eingetragen werden. Nach dem Aufruf enthalten die beiden Bytes die tatsächliche Länge (einschließlich Header).
Byte 2 - 3	werden mit binären Nullen aufgefüllt
Byte 4 - n	Daten

LSTLEN ist angegeben:
Die Ausgabeliste enthält nur die Daten.

LSTLEN

Adresse eines Wortes, das die Länge der Ausgabeliste enthält.

Die Ausgabeliste wird an der mit LSTADDR angegebenen Adresse geliefert, ohne vorausgehendes Längenfeld.

Vor dem Aufruf muss das Wort, dessen Adresse in LSTLEN angegeben wird, die maximale Länge der Ausgabeliste enthalten.
Nach dem Aufruf enthält dieses Wort die tatsächliche Länge der Ausgabeliste.

Wenn Sie LSTLEN nicht angeben, wird die Ausgabeliste im V-Format geliefert.

4.2.3 Hinweise zur Programmierung

Der Makro NLSCMP (NATIONAL LANGUAGE SUPPORT: COMPATIBILITY TABLE) wird über einen Supervisor-Call (SVC) aufgerufen. NLSCMP liefert eine Liste kompatibler Codes. Dazu muss entweder der Name eines Zeichensatzes oder die Nummer einer ISO-Code-Variante angegeben werden.

Bei Angabe des Zeichensatznamens (CCSNAME) wird die Nummer der entsprechenden ISO-Code-Variante in die Parameterliste übergeben (Strukturelement &PR.ISON). Wenn Sie als Zeichensatznamen eine Unicode-Variante angeben, d.h. UNICODE, UTF16, UTF8 oder UTFE, wird als ISO-Code-Variante der Wert 240 = X'F0' zurückgegeben. Standardmäßig beginnt das Strukturelement mit den Buchstaben 'GNLI'. Sie können diese Buchstaben jedoch durch eine selbstgewählte Zeichenfolge ersetzen.

Wenn Sie CCSNAME = <unicodevariante> oder ISOVAR=*ALL angeben, enthält die Ausgabeliste neben den Namen aller im System definierten 7- bzw. 8-bit-Codes auch die drei Unicode-Varianten UTF-16, UTF-8 und UTFE.

ISOVAR kann sowohl für ISO-Codes wie auch für EBCDIC-Codes stehen.

Die ausgegebene Liste muss auf Halbwortgrenze beginnen. Die Struktur hängt davon ab, ob Sie den Operanden LSTLEN angegeben haben.

Struktur, wenn LSTLEN nicht angegeben wurde (V-Format):
- Byte 0 - 1 Gesamtlänge der Liste (einschließlich Header)
- Byte 2 - 3 reserviert
- Byte 4 - n Daten

Struktur, wenn LSTLEN angegeben wurde:
- Byte 0 - n Daten

Der Datenteil setzt sich aus einer Folge von 10 Byte langen Zeichenketten zusammen. Davon enthalten die ersten 8 Byte den Code-Namen (CCSN). Das 9-te Byte wird auf den Wert '␣' gesetzt, falls der Code mit der entsprechenden ISO-Code-Variante ISO 8859 voll kompatibel ist. Ist der Code begrenzt kompatibel, enthält das 9. Byte ein 'P' (siehe Abschnitt „Gruppierung kompatibler Codes“ auf Seite 116).

Das 10-te Byte der jeweils 10 Bytes umfassenden Einträge enthält die Art des Codes:

Art des Codes	**Wert**	
ISO-Code	'I'	(ISO)
EBCDIC-Code	'E'	(EBCDIC)
UTF-16	'I'	(ISO)
UTF-8	'I'	(ISO)
UTFE	'E'	(EBCDIC)

4.2.4 Returncodes

SUBCODE 2 | 1		MAINCODE 2 1	Bedeutung
X' 00'	X' 00'	X' 00' X' 00'	erfolgreiche Bearbeitung; gewünschte Daten verfügbar
X' 00'	X' 01'	X' 00' X' 04'	Name des Codes (CCSN) oder der ISO-Variante unbekannt
X' 00'	X' 01'	X' 00' X' 14'	inkonsistente Parameterliste
X' 00'	X' 01'	X' 00' X' 80'	Fehler bei der Speicherzuordnung
X' 00'	X' 01'	X' 00' X' 84'	Ausgabetabelle ist zu klein, Daten werden abgeschnitten
X' 00'	X' 01'	X' 00' X' 88'	Länge der Zeichenkette ungültig
X' 00'	X' 01'	X' 00' X' 8C'	Fehler bei der Ausrichtung auf Halbwortgrenze
X' 00'	X' 01'	X' FF' X' FF'	Die angeforderte Funktion wird nicht unterstützt. Nicht behebbarer Fehler.
X' 00'	X' 02'	X' FF' X' FF'	Die angeforderte Funktion ist nicht verfügbar. Nicht behebbarer Fehler. Mögliche Ursache: XHCS-SYS fehlt in der aktuellen Konfiguration
X' 00'	X' 03'	X' FF' X' FF'	Die angegebene Version der Schnittstelle wird nicht unterstützt (falsche Versionsangabe im Standard-Header). Nicht behebbarer Fehler.
X' 00'	X' 04'	X' FF' X' FF'	Parameterliste ist nicht auf Wortgrenze ausgerichtet.
X' 00'	X' 41'	X' FF' X' FF'	Das Subsystem ist nicht vorhanden; es muss explizit erzeugt werden.
X' 00'	X' 42'	X' FF' X' FF'	Der aufrufende Ablauf ist mit dieser Schnittstelle nicht konnektiert; er muss explizit konnektiert werden.
X' 00'	X' 81'	X' FF' X' FF'	Subsystem zurzeit nicht verfügbar.
X' 00'	X' 82'	X' FF' X' FF'	Subsystem im DELETE- oder HOLD-Zustand.

Bedeutung der einzelnen Felder:

SUBCODE 1 Anzeigen der Fehlerklasse

= X'00': Erfolgreiche Bearbeitung

≠ X'00': Fehler

SUBCODE 2 wird immer auf den Wert X'00' gesetzt.

MAINCODE 1 Fehlerinformationen für das Anwendungsprogramm. Das Anwendungsprogramm kann damit Bedienungsfehler abfangen.

MAINCODE 2 wird zurzeit nicht belegt und auf den Wert X'00' gesetzt. Wenn im Standard-Header Fehler sind, die der Selbstidentifikation des Produktes dienen (z.B. falsche Version), wird der Wert auf X'FF' gesetzt.

Beispiel

```
EXMPL    START
*
         BALR   10,0
         USING  *,10
*
         NLSCMP MF=E,PARAM=PL ------------------------------------(01)
*
*
         TERM
*
*
PL       NLSCMP MF=L,CCSNAME=*SYSDEF,LSTADDR=LSTCOMP -------------(02)
*
*
*
*        Speicherbereich der Liste kompatibler Codes (V-Format)
*
         DS     0H
LSTCOMP  DC     Y(LSTEND-LSTCOMP)---------------------------------(03)
RES      DC     Y(0) ----------------------------------------------(04)
DATA     DS     CL60 ----------------------------------------------(05)
LSTEND   EQU    *
*
*
         END
```

(01) Aufruf von XHCS

(02) Die Parameterliste wird deklariert und initialisiert. Es wird eine Liste der Codes angefordert, die mit dem Systemstandard kompatibel sind.

(03) Maximale Länge

(04) reserviert

(05) Datenteil

4.3 Zeichenkette umwandeln: NLSCNV

4.3.1 Makroaufrufformat

Operation	Operanden
NLSCNV	MF=D [,PREFIX= <prefix>] [,XPAND= PARAM / LOADCB]
	MF=C [,PREFIX= <prefix>][,MACID = <macid>] [,XPAND= PARAM / LOADCB]
	MF=E [,MODE=*SVC / *BASR / *LOAD] ,PARAM=<paramlist> / (register)
	Aktion: Zeichenketten in kompatiblen Zielcode umwandeln MF=L / S [,ACTION= CONVERT] ,CCSNAME= <7-/8-bit-Code> / <unicodevariante> / *SYSDEF / *USRDEF ,TOCCS= <7-/8-bit-Code> / <unicodevariante> / *SYSDEF / *USRDEF ,INSTRG= <addr> [,INLEN= <addr>] ,OUTSTRG= <addr> [,OUTLEN= <addr>] [,DEFAULT= *STD / *NO / *UTF16 / *TARGET / *SPECIAL] [,DEFVAL= *NONE / xx / xxxx / xxxxxx] [,DEFLEN= *NONE / 1 / 2 / 3] [,TABADDR= *NONE / <tabaddr>]
	Aktion: Klein- in Großbuchstaben umwandeln und umgekehrt MF=L / S ,ACTION= TOUPPER / TOLOWER ,CCSNAME= <7-/8-bit-Code> / <unicodevariante> / *SYSDEF / *USRDEF ,INSTRG= <addr> [,INLEN= <addr>] ,OUTSTRG= <addr> [,OUTLEN= <addr>] [,DEFAULT= *STD / *NO / *UTF16 / *TARGET] [,DEFVAL= *NONE / xx / xxxx / xxxxxx] [,DEFLEN= *NONE / 1 / 2 / 3] [,TABADDR= *NONE / <tabaddr>]

Operation	Operanden
NLSCNV	**Aktion: Eingabezeichenkette normalisieren** MF=L / S ,ACTION= COMPOSE / DECOMPOSE ,INSTRG= <addr> [,INLEN= <addr>] ,OUTSTRG= <addr> [,OUTLEN= <addr>] [,DEFAULT= *STD / *NO / *UTF16 / *TARGET] [,DEFVAL= *NONE / xx / xxxx / xxxxxx] [,DEFLEN= *NONE / 1 / 2 / 3]
	Aktion: Länge von UTF-8- bzw. UTFE-Zeichenketten ausgeben MF=L / S ,ACTION= LENGTH ,CCSNAME= <7-/8-bit-Code> / <unicodevariante> / *SYSDEF / *USRDEF [,TOCCS= <7-/8-bit-Code> / <unicodevariante> / *SYSDEF / *USRDEF] ,INSTRG= <addr> [,INLEN= <addr>] ,LENGTH= <addr>

4.3.2 Operandenbeschreibung

MF

=D	Generiert eine DSECT.
=E	Generiert einen Befehlsteil. Es werden keine Befehle zur Auswertung des Returncodes erzeugt.
=C	Generiert das Layout des NLSCNV in der aktuellen Datenstruktur. Dabei wird jedes Feld benannt und es werden alle Equates abgesetzt. Der Standard-Header wird nicht versorgt.
=S	Generiert eine Parameterliste. Die Felder werden gemäß den angegebenen Schlüsselwortparametern des Makros versorgt. Namen und Equates werden nicht abgesetzt. Zusätzlich wird ein Befehlsteil generiert. Der Standard-Header wird automatisch versorgt. Es werden keine Befehle zur Auswertung des Returncodes erzeugt.
=L	Generiert eine Parameterliste. Die Felder werden gemäß den angegebenen Schlüsselwortparametern des Makros versorgt. Namen und Equates werden nicht abgesetzt. Der Standard-Header wird automatisch versorgt.

PREFIX

=x	Gibt das erste Zeichen an, das den Namen vorangestellt wird, die mit MF=D oder MF=C definiert werden. Der Standardwert ist PREFIX=G.

MACID

=xxx	Gibt das 2.-4. Zeichen der Namen an, die mit MF=C definiert werden. Der Standardwert ist MACID=NLC (auch für MF=D).

XPAND	Angabe, welche Datenstruktur expandiert werden soll. Wenn Sie XPAND nicht angeben, gilt XPAND= PARAM. XPAND wird nur bei MF=D / C angeboten.
=PARAM	Die NLSCNV-Parameterliste wird expandiert.
=LOADCB	Der Kontrollblock für den Ansprung mit MODE= *LOAD wird expandiert.

PARAM

=<paramlist>	Gibt den Namen der Parameterliste an.
=(register)	Enthält die Adresse der Parameterliste.

MODE

Angabe, wie die funktionsausführende Komponente angesprungen wird (siehe auch Seite 85).
Der Operand wird nur bei MF=E ausgewertet.

Wenn Sie den Operand MODE nicht angeben, wird der Wert *SVC angenommen.

=*SVC

Es wird ein SVC generiert.
R1 wird vom Makro mit der Adresse von <paramlist> geladen. Bei PARAM = <paramlist> wird die Angabe der NLSCNV-Parameterliste erwartet.

=*BASR

Der funktionsausführende Entry GNLCNV wird direkt angesprungen.

Das LLM GNLCNV muss fest zum Anwenderprogramm dazugebunden sein.
Das LLM GNLCNV ist in der Modulbibliothek SYSLNK.XHCS-SYS.020.TU enthalten.

Dieser Modus ist für Anwenderprogramme gedacht, die ausschließlich im Klasse-6-Speicher ablaufen.

Dieser Modus setzt voraus, dass das aufrufende Programm ILCS-fähig ist, und dass mit dem Adressierungsmodus ANY (= 31-bit-fähig) gearbeitet wird.

Es werden folgende Befehle generiert:

L	R15, = V(GNLCNVC)
LA	1, <addrparamlist>
BASR	R14, R15

Beim Aufruf des Makros NLSCNV mit MODE=*BASR werden folgende Register benötigt:

R1	wird vom Makro mit der Adresse von <addrparamlist> geladen. Bei PARAM = <addrparamlist> wird die Adresse eines Feldes erwartet, das die Adresse der NLSCNV-Parameterliste enthält. Beachten Sie den Unterschied zu MODE=*SVC, dort wird die Adresse der NLSCNV-Parameterliste direkt in R1 angegeben.
R13	ist vor dem Makroaufruf mit der Adresse eines 18 Worte langen Sicherstellungsbereiches zu laden, den das aufrufende Programm zur Verfügung stellen muss.
R14	wird vom Makro mit der Rückkehradresse des Anwenderprogramms geladen.

R15 = V (GNLCNVC)

=*LOAD

Es wird der reentrant-fähige Adaptermodul GNLADPT angesprungen, der mit BIND den funktionsausführenden Modul GNLCNV aus der Modulbibliothek SYSLNK.XHCS-SYS.020.TU in den Klasse-6-Speicher nachlädt, und anschließend dorthin verzweigt.

Der Adaptermodul GNLADPT muss zum Anwenderprogramm dazugebunden sein.
Dieser Adaptermodul liegt als R-Modul vor und ist in der Bibliothek SYSOML.XHCS-SYS.020 enthalten.

Dieser Modus ist gedacht für Anwenderprogramme, bei denen die Schnittstelle NLSCNV an einer Stelle aufgerufen wird, die reentrant-fähig sein muss.

Bei der Generierung des Befehlsteils wird zwar wie bei MODE=*BASR eine V-Konstante erzeugt, aber GNLADPT ist reentrant-fähig und kann daher für alle Tasks an der gleichen Stelle liegen.

Nachgeladen wird nur beim ersten Aufruf mit MODE=*LOAD, d.h. wenn die Adresse im LOAD-Kontrollblock GNLCLCB (siehe unten und Seite 107) noch nicht versorgt ist.

In diesem Modus ist es nicht zwingend erforderlich, dass das aufrufende Programm ILCS-fähig ist, und dass mit dem Adressierungsmodus ANY (= 31-bit-fähig) gearbeitet wird.

Es werden folgende Befehle generiert:

```
L     R15, = V(GNLADPT)
LA    1,<addrcontrolblock>
BASR  R14, R15
```

Beim Aufruf des Makros NLSCNV mit MODE=*LOAD werden folgende Register benötigt:

R1 wird vom Makro mit der Adresse von <addrcontrolblock> geladen. Bei PARAM = <addrcontrolblock> wird die Angabe eines 20 Byte langen tasklokalen Kontrollblocks erwartet, der folgende Informationen enthält :

- Adresse des Entries GNLCNVC, die von GNLADPT versorgt wird
- Indikator, ob dieses Adressfeld versorgt ist, d.h. ob das Nachladen bereits erfolgte
- Adresse der NLSCNV-Parameterliste
- Returncode

Das Layout dieses Kontrollblocks kann mit XPAND = LOADCB generiert werden.
Beim ersten Aufruf mit MODE=*LOAD müssen alle Felder des LOAD-Kontrollblocks (mit Ausnahme der Adresse der NLSCNV-Parameterliste) mit 0 vorbelegt sein. Das Adaptermodul GNLADPT versorgt die restlichen Felder des Kontrollblocks mit gültigen Werten.
Bei den folgenden Aufrufen muss dann immer der von GNLADPT versorgte Kontrollblock mitgegeben werden.
Das aufrufende Programm sollte darauf achten, dass der von GNLADPT versorgte Kontrollblock für einen weiteren NLSCNV-Aufruf mit MODE=*LOAD erhalten bleibt.

R13 ist vor dem Makroaufruf mit der Adresse eines 18 Worte langen Sicherstellungsbereiches zu laden, den das aufrufende Programm zur Verfügung stellen muss.

R14 wird vom Makro mit der Rückkehradresse des Anwenderprogramms geladen.

R15 = V (GNLADPT)

Der Returncode der NLSCNV-Funktion wird im Standard-Header der NLSCNV-Parameterliste geliefert. Der Returncode für das Nachladen von GNLCNV mit BIND wird im Returncode-Feld des LOAD-Kontrollblocks geliefert.

ACTION Angabe der gewünschten Aktion.

Detaillierte Information dazu entnehmen Sie dem Abschnitt „Funktionsübersicht“ auf Seite 102.

CCSNAME Der Name des Ausgangscodes wird festgelegt. Der Name ist max. 8 Byte lang. Wird kein Name angegeben, wird automatisch der EDF03IRV als Ausgangscode angenommen.

bei ACTION=LENGTH:
CCSNAME muss angegeben werden. Es gibt keinen Standard.

=<7-/8-bit-Code> Name eines 7-bit oder 8-bit-Ausgangscodes

=*SYSDEF Der System-Standard-Code ist der Ausgangscode.

=*USRDEF Der Standard-Anwender-Code ist der Ausgangscode.

=<unicodevariante>
Name des Unicode-Ausgangscodes. Als Unicode-Variante können Sie UNICODE, UTF16 und UTF8 oder UTFE für UTF-EBCDIC angeben. Die Angabe von 'UNICODE' ist gleichbedeutend mit UTF16.

bei ACTION=LENGTH:
nur UTF16 bzw. bei Längenberechnung von vorgegebenen Strings nur UTF8- oder UTFE-Strings.

TOCCS Der Name des Zielcodes wird festgelegt. Der Name ist max. 8 Byte lang. Wird kein Name angegeben, wird automatisch der EDF03IRV als Zielcode angenommen.

=<7-/8-bit-Code> Name eines 7-bit oder 8-bit-Zielcodes

=*SYSDEF Der System-Standard-Code ist der Zielcode.

=*USRDEF Der Standard-Anwender-Code ist der Zielcode.

=<unicodevariante>
Name des Unicode-Zielcodes. Als Unicode-Variante können Sie UNICODE, UTF16 und UTF8 oder UTFE für UTF-EBCDIC angeben. Die Angabe von 'UNICODE' ist gleichbedeutend mit UTF16.

bei ACTION=LENGTH:
Sie können nur UTF8 oder UTFE angeben. Alle anderen Angaben werden mit Fehler abgewiesen.

INSTRG

Adresse der Eingabezeichenkette.

Die Struktur der Eingabezeichenkette hängt von der Angabe beim Operanden INLEN ab:

INLEN ist nicht angegeben:
Die Eingabezeichenkette wird im V-Format erwartet:
Byte 0-1: Gesamtlänge der Zeichenkette (einschließlich Header)
Byte 2-3: reserviert
Byte 4-n: Zeichenkette
Die Länge der Eingabezeichenkette darf höchstens 32767 Bytes betragen.
Die Adresse der Eingabezeichenkette muss auf Halbwortgrenze beginnen.

INLEN ist angegeben:
Die Eingabezeichenkette enthält nur die zu bearbeitenden Zeichen.
Die Länge der Eingabezeichenkette darf höchstens 65536 Bytes betragen.
Die Adresse der Eingabezeichenkette muss nicht zwingend auf Halbwortgrenze beginnen.

INLEN

Adresse eines Wortes, das die Länge der Eingabezeichenkette enthält. Dieser Wert darf maximal 65536 Bytes betragen.

Die Eingabezeichenkette wird an der mit INSTRG angegebenen Adresse ohne vorausgehendes Längenfeld erwartet.

Wenn Sie INLEN nicht angeben, wird die Eingabezeichenkette im V-Format erwartet.

OUTSTRG

Adresse der Ausgabezeichenkette.

Die Struktur der Ausgabezeichenkette hängt von der Angabe beim Operanden OUTLEN ab:

OUTLEN ist nicht angegeben:
Die Ausgabezeichenkette wird im V-Format geliefert:
Byte 0 - 1: Gesamtlänge der Zeichenkette (einschließlich Header)
Byte 2 - 3: werden mit binären Nullen aufgefüllt
Byte 4 - n: Zeichenkette
Die Adresse der Ausgabezeichenkette muss auf Halbwortgrenze beginnen.

OUTLEN ist angegeben:
Die Ausgabezeichenkette enthält nur die umgewandelten Zeichen.
Die Adresse der Ausgabezeichenkette muss nicht zwingend auf Halbwortgrenze beginnen.

OUTLEN

Adresse eines Wortes, das nach einer erfolgreichen Umwandlung die Länge der Ausgabezeichenkette enthält.

Die Ausgabezeichenkette wird an der mit OUTSTRG angegebenen Adresse ohne vorausgehendes Längenfeld geliefert.

Vor dem Aufruf muss das Wort, dessen Adresse in OUTLEN angegeben wird, einen der beiden folgenden Werte enthalten:

0	keine Vorgabe, wie groß die Ausgabezeichenkette maximal werden darf
wert > 0	Vorgabe, wie groß die Ausgabezeichenkette maximal werden darf

Nach dem Aufruf enthält dieses Wort die tatsächliche Länge der Ausgabezeichenkette.
Dabei gilt: Wenn Sie DEFAULT=*NO angegeben haben, und die Verarbeitung wegen des Auftretens eines ungültigen Zeichens (d.h. mit Returncode X'00010028') abgebrochen wird, enthält dieses Wort die Länge der bis dahin erzeugten Ausgabezeichenkette (in Bytes).
Wenn die Ausgabezeichenkette abgeschnitten wird, weil die vorgegebene Maximallänge erreicht wurde, wird der Returncode X'00000084' gesetzt.

Wenn Sie OUTLEN nicht angeben, wird die Ausgabezeichenkette im V-Format geliefert.

DEFAULT

Art der Ersatzzeichenbehandlung

XHCS V2.0 bietet die Möglichkeit, ein Ersatzzeichen anzugeben, welches in die Ausgabezeichenkette eingesetzt werden soll, falls einer der folgenden Fälle auftritt:

- Ein Zeichen in der Eingabezeichenkette ist nicht im Quell-CCS definiert.
- Der Quell-CCS ist eine Unicode-Variante und ein Zeichen in der Eingabezeichenkette gehört nicht zu den von XHCS unterstützten Unicode-Zeichen.
- Bei der Aktion CONVERT ist ein Zeichen in der Eingabezeichenkette nicht im Ziel-CCS definiert.

Mit "Ziel-CCS" ist im Folgenden gemeint:

bei ACTION=	**Ziel-CCS**
CONVERT	der mit TOCCS angegebene CCS
TOUPPER /TOLOWER	der mit CCSNAME angegebene CCS
COMPOSE /DECOMPOSE	UTF-16

Wenn Sie den Operand DEFAULT nicht angeben, wird der Wert *STD angenommen.

=*STD

Der Wert C'.' (= U+002E "full stop") wird als Ersatzzeichen eingesetzt. XHCS wandelt diesen Wert in den gewünschten Ziel-CCS um.

Angaben für die Operanden DEFVAL und DEFLEN werden ignoriert.

Hinweis zur Kompatibilität zu XHCS V1.5
Wenn in XHCS V1.5 bei der Konvertierung von 7-/8-bit-Code nach 7-/8-bit-Code ein nicht definiertes Zeichen auftritt, wird ein NIL-Zeichen eingesetzt.
Wenn Sie ein mit den Makros aus XHCS V1.5 übersetztes Programm ablaufen lassen, d.h. wenn bei XHCS eine NLSCNV-Parameterliste im Layout von XHCS V1.5 eintrifft, wird aus Kompatibilitätsgründen das NIL-Zeichen als Ersatzzeichen verwendet, um das bisherige Verhalten zu bewirken.
Wenn Sie eine alte Programm-Source neu mit den Makros aus XHCS V2.0 übersetzen, und zwar ohne Source-Änderung, d.h. insbesondere ohne den neuen Operanden DEFAULT anzugeben, wird als Ersatzzeichen der Punkt verwendet. Dies stellt eine Inkompatibilität gegenüber der Vorversion dar.

=*NO

Es wird kein Ersatzzeichen verwendet. Wenn ein Zeichen gefunden wird, das nicht im angegebenen Code vorkommt bzw. das von XHCS nicht unterstützt wird, wird die Konvertierung abgebrochen und der Returncode X'00010028' zurückgeliefert.

Die Ausgabezeichenkette enthält die bis dahin umgesetzten Zeichen.

Die Länge der bis zum Abbruch erzeugten Ausgabezeichenkette (in Bytes) steht,

- falls der Operand OUTLEN nicht angegeben wurde, im Längenfeld am Beginn des Ausgabebereichs.
- falls der Operand OUTLEN angegeben wurde, in der bei OUTLEN angegebenen Adresse.

Die relative Position des ungültigen Zeichens in der Eingabezeichenkette steht im Ausgabefeld &PR.PICI (Position of Invalid Character in Input String). In allen anderen Fällen enthält dieses Ausgabefeld den Wert 0.
Beachten Sie, dass hier nicht ein Offset in Bytes geliefert wird, sondern die Angabe, um das wievielte Zeichen in der Eingabezeichenkette es sich bei dem ungültigen Zeichen handelt.
Beispiel
In der UTF-8-Eingabezeichenkette C2A0C3A2CCCC hat das ungültige Zeichen CCCC die Position 3, d.h. nach den beiden gültigen UTF-8-Zeichen C2A0 und C3A2 ist das 3. Zeichen in der Zeichenkette ungültig.

Angaben für die Operanden DEFVAL und DEFLEN werden ignoriert.

=*UTF16 /*TARGET

Verwendung eines frei gewählten Ersatzzeichens.

=*UTF16

Der beim Operanden DEFVAL angegebene Wert wird unabhängig vom gewählten Quell- und Ziel-CCS als UTF-16-Wert interpretiert. XHCS wandelt den Wert in den gewünschten Ziel-CCS um.

Ist das angegebene Zeichen im Ziel-CCS nicht definiert, wird ein negativer Returncode zurückgeliefert.

Der Operand DEFVAL wird in der Form xxxx erwartet.
Beispiel
Die Angabe DEFVAL=003F wird als U+003F (= C'?') interpretiert.

Der Operand DEFLEN wird ignoriert

=*TARGET

Der beim Operanden DEFVAL angegebene Wert wird als Zeichen des Ziel-CCS interpretiert.

Die Operanden DEFVAL und DEFLEN werden ausgewertet. Wenn die bei DEFLEN angegebene Länge des Ersatzzeichens nicht zum gewählten Ziel-CCS passt (z.B. Ziel-CCS = 8-bit-Code und DEFLEN=2, oder Ziel-CCS = UTF16 und DEFLEN=1, usw.), wird ein negativer Returncode zurückgeliefert.
Ansonsten wird der bei DEFVAL angegebene Wert unverändert in der mit DEFLEN angegebenen Länge übernommen.

Beispiel
Ziel-CCS = EDF041, DEFVAL = 6F, DEFLEN = 1
Ersatzzeichen ist C'?'
Ziel-CCS = UTF16, DEFVAL = 003F, DEFLEN = 2
Ersatzzeichen ist C'?'

=*SPECIAL

Spezielle Umsetzung für eine eindeutig umkehrbare Konversion von Ersatzzeichen (siehe auch Abschnitt „Eindeutig umkehrbare Konversion für alle existierenden 7-/8-bit-Zeichensätze“ auf Seite 108).
Dieser Wert darf nur angegeben werden bei ACTION = CONVERT und
– Quell-CCS = 7- / 8-bit-Code und Ziel-CCS = UNICODE / UTF16 oder
– Quell-CCS = UNICODE /UTF16 und Ziel-CCS = 7- / 8-bit-Code

Für jeden nicht definierten Code-Punkt im 7- oder 8-bit-Quell-CCS wird ein unterschiedliches Ersatzzeichen aus dem Bereich U+E000 bis U+F8FF ("for private use") eingesetzt, und zwar das Zeichen U+F200 + <nicht definierter Codepunkt>. Mit Hilfe dieser Zuordnung werden die entsprechenden Zeichen abgebildet und wieder zurück abgebildet.
Beispiel
Wenn der Codepunkt X'ED' nicht definiert ist, wird der Wert X'F2ED' eingesetzt.

Wenn die so erzeugte Ausgabezeichenkette wieder von Unicode in den ursprünglichen Quell-CCS zurückkonvertiert wird, werden die Ersatzzeichen wieder in den Originalwert umgesetzt. Dieser ist ein nicht definierter Codepunkt des Quell-CCS.

Angaben für die Operanden DEFVAL und DEFLEN werden ignoriert.

DEFVAL

Wert des gewünschten Ersatzzeichens, der in die Ausgabezeichenkette eingesetzt werden soll.

Zwingend ist der Operand bei DEFAULT=*UTF16 / *TARGET. Bei DEFAULT=*STD / *NO / *SPECIAL wird der Operand ignoriert.

= xx

Angabe eines hexadezimalen Ein-Byte-Wertes.

= xxxx

Angabe eines hexadezimalen Zwei-Bytes-Wertes.

= xxxxxx

Angabe eines hexadezimalen Drei-Bytes-Wertes.

Der angegebene Wert wird in das Feld &PR.DVAL linksbündig eingetragen.

Achtung
Wenn Sie den Wert nicht via MF=L (Ass) angeben, sondern das entsprechende Feld in der Parameterliste durch explizite Zuweisung versorgen, geben Sie im Feld &PR.DIND den Wert &PR.DU16 oder &PR.DTRG an. Wenn Sie &PR.DTRG angegeben haben, muss auch das Längenfeld &PR.DLEN versorgt werden.

DEFLEN

Angabe der Länge des Ersatzzeichens.

Der Operand ist zwingend bei DEFAULT=*TARGET. Bei allen anderen Angaben für DEFAULT wird er ignoriert.

Bei Angabe von DEFLEN=n werden die ersten n Bytes im Feld &PR.DVAL als Ersatzzeichen genommen.

LENGTH

bei ACTION=LENGTH:
Adresse des Wortes, wohin die ermittelte Länge geschrieben werden soll

TABADDR — Adresse einer Umsetzungstabelle

Mit dem Operanden TABADDR haben Sie die Möglichkeit, sich für eine bestimmte Aktion die Umsetzungstabelle herausreichen zu lassen, und bei jeder Aktion dann wieder die passende Umsetzungstabelle mit TABADDR mitzugeben, so dass XHCS sich auch bei wechselnden Aktionen den SVC zur Besorgung der Umsetzungstabelle spart. Dazu ist es jedoch erforderlich, dass Sie die verschiedenen Umsetzungstabellen selbst verwalten.

Wenn Sie den Operanden TABADDR nicht angeben, wird der Wert *NONE verwendet.

Wird die Eingabezeichenkette normalisiert (ACTION = COMPOSE / DECOMPOSE), wird der Operand TABADDR ignoriert.

= *NONE — Es wird weder eine Umsetzungstabelle an XHCS mitgegeben, noch wird eine Umsetzungstabelle von XHCS hinausgereicht. XHCS besorgt sich die benötigte Konvertierungstabelle selbst.

= <tabaddr> — Adresse eines Bereichs, der aus zwei Worten besteht

1. Wort — Bedeutung der Adresse im 2. Wort

-1	Das 2. Wort enthält die Adresse einer Tabelle, die von XHCS bei der gewünschten NLSCNV-Aktion als Umsetzungstabelle verwendet werden soll.
Wert ≠ -1	Die von XHCS bei der gewünschten NLSCNV-Aktion verwendete Umsetzungstabelle soll an die im 2. Wort angegebene Adresse kopiert werden.

2. Wort — Adresse an der die Umsetzungstabelle erwartet wird oder an die die Umsetzungstabelle kopiert werden soll.

4.3.3 Funktionsübersicht

Die Funktionen des Makro NLSCNV sind vom Operanden ACTION abhängig:

ACTION=	Bedeutung	siehe auch
CONVERT	Zeichenketten in kompatiblen Zielcode umwandeln	Seite 102
TOUPPER/TOLOWER	Klein- in Großbuchstaben umwandeln und umgekehrt	Seite 104
COMPOSE/DECOMPOSE	Eingabezeichenkette normalisieren	Seite 104
LENGTH	Länge von UTF-8- bzw. UTFE-Zeichenketten ausgeben	Seite 105

ACTION=CONVERT: Zeichenketten in kompatiblen Zielcode umwandeln

NLSCNV wandelt Zeichenketten, die entsprechend des Ausgangscodes (CCSNAME) codiert sind, direkt in den kompatiblen Zielcode (TOCCS) um.

Weder der Ausgangscode noch der Zielcode ist Unicode:

Die Ein- und Ausgabeketten können teilweise oder vollständig (INSTRG = OUTSTRG) den gleichen Speicherbereich belegen.

Die Länge der Ausgabekette ist gleich der Länge der Eingabekette.

Für Zeichenketten, die mit Unicode codiert sind:

Außer den Umwandlungen zwischen kompatiblen 8-bit-Codes, gibt es folgende zusätzlichen Umwandlungsmöglichkeiten:

- Ein spezifizierter 7-bit- oder 8-bit-Code (CCSNAME) kann in eine Unicode-Variante (TOCCS) umgewandelt werden.

  Achten Sie darauf, dass der Ausgabebereich groß genug ist, da die Ausgabelänge bis zu dreimal so groß wie die Eingabelänge sein kann, siehe auch nachfolgende Tabellen.

- Eine Unicode-Zeichenkette kann in einen spezifizierten 7-bit- oder 8-bit-Code umgewandelt werden. Bei Angabe von ’UNICODE’ oder ’UTF16’ wird die Ausgabelänge halbiert, bei ’UTF8’ oder ’UTFE’ ist sie von der Art der verwendeten Zeichen abhängig.

- Eine Unicode-Variante kann in eine andere umgewandelt werden, z.B. eine UTF-16-Zeichenkette nach UTF-8 oder UTFE und umgekehrt.

  Wenn ein Zeichen gefunden wird, das nicht im angegebenen 7-bit oder 8-bit-Code bzw. nicht in der Unicode-Unterstützung von XHCS vorkommt,
  wird,

  – falls DEFAULT=*NO angegeben wurde, die Verarbeitung abgebrochen und der Returncode X'00010028' gesetzt,

  – falls für DEFAULT ein Wert ungleich *NO angegeben wurde, das angegebene (evtl. in den Ziel-CCS konvertierte) Ersatzzeichen anstelle des nicht definierten Zeichens eingefügt und der Returncode X'00010024'gesetzt.

  – falls DEFAULT nicht angegeben wurde, das Standard-Ersatzzeichen eingefügt und der Returncode X'00010024' gesetzt. Als Standard-Ersatzzeichen gilt '.'.

Maximale Länge der Ausgabezeichenkette (L_E = Länge der Eingabezeichenkette):

Zielcode / Ausgangscode	8-bit-Code	UTF-16	UTF-8	UTFE
8-bit-Code	L_E	$2*L_E$	$3*L_E$	$3*L_E$
UTF-16	$0,5*L_E$	L_E	$1,5*L_E$	$2*L_E$
UTF-8	L_E	$2*L_E$	L_E	$1,5*L_E$
UTFE	L_E	$2*L_E$	$2*L_E$	L_E

Bei einer maximalen Länge der Eingabezeichenkette von 64 K ergibt sich als maximal mögliche Länge der Ausgabezeichenkette:

Zielcode / Ausgangscode	8-bit-Code	UTF-16	UTF-8	UTFE
8-bit-Code	64 K	128 K	192 K	192 K
UTF-16	32 K	64 K	96 K	128 K
UTF-8	64 K	128 K	64 K	96 K
UTFE	64 K	128 K	128 K	64 K

ACTION=TOUPPER / TOLOWER: Klein- in Großbuchstaben umwandeln und umgekehrt

Diese Funktion wandelt in Zeichenketten, die entsprechend des Ausgangscodes (CCSNAME) codiert sind, die Kleinbuchstaben in Großbuchstaben (TOUPPER) bzw. die Großbuchstaben in Kleinbuchstaben um (TOLOWER).

Wenn ein Zeichen im unterstützten Code-Vorrat nicht vorhanden ist, greift die mit den Operanden DEFAULT, DEFVAL und DEFLEN definierte Ersatzzeichenbehandlung:

– Falls DEFAULT=*NO angegeben wurde, wird die Verarbeitung abgebrochen und der Returncode X'00010028' gesetzt.

– Falls für DEFAULT ein Wert ungleich *NO angegeben wurde, wird das angegebene Ersatzzeichen eingefügt und der Returncode X'00010024' gesetzt.

– Falls DEFAULT nicht angegeben wurde, wird das Standard-Ersatzzeichen '.' (=U+002E) eingefügt und der Returncode X'00010024' gesetzt.

Ansonsten bleibt es bei der Umwandlung unverändert.

Die Ein- bzw. Ausgabeketten stehen an den durch INSTRG bzw. OUTSTRG angegebenen Adressen und können sich teilweise oder vollständig überlappen.

Ausgabezeichenkette und Eingabezeichenkette sind gleich lang.

ACTION=COMPOSE / DECOMPOSE: Eingabezeichenkette normalisieren

Bei dieser Angabe normalisiert XHCS die Eingabezeichenkette.

– COMPOSE: Umwandlung in die Normalization Form C
– DECOMPOSE: Umwandlung in die Normalization Form D
(siehe Abschnitt „Unicode-Unterstützung“ auf Seite 18)

Die Eingabezeichenkette wird in UTF-16 erwartet.

Wenn ein Zeichen im unterstützten Code-Vorrat nicht vorhanden ist, greift die mit den Operanden DEFAULT, DEFVAL und DEFLEN definierte Ersatzzeichenbehandlung:

– Falls DEFAULT=*NO angegeben wurde, wird die Verarbeitung abgebrochen und der Returncode x'00010028' gesetzt.

– Falls für DEFAULT ein Wert ungleich *NO angegeben wurde, wird das angegebene Ersatzzeichen eingefügt und der Returncode x'00010024' gesetzt.

– Falls DEFAULT nicht angegeben wurde, wird das Standard-Ersatzzeichen '.' (=U+002E) eingefügt und der Returncode x'00010024' gesetzt.

Maximal mögliche Länge der Ausgabezeichenkette

ACTION	**Maximal mögliche Länge der Ausgabezeichenkette**
COMPOSE	Länge der Eingabezeichenkette
DECOMPOSE	3 * Länge der Eingabezeichenkette Da die Eingabe in UTF-16 erwartet wird, ist ein Zeichen 2 Bytes lang. Im "ungünstigsten" Fall ist jedes Eingabezeichen ein Zeichen mit drei Collation-Elementen.

ACTION=LENGTH: Länge von UTF-8- bzw. UTFE-Zeichenketten ausgeben

Diese Funktion bietet Ihnen folgende zwei Möglichkeiten:

- Für 8-bit- bzw. UTF-16-Zeichenketten wird die Länge der entsprechenden UTF-8 (UTFE)-Zeichenkette ausgegeben.

  Geben Sie dazu die Parameter INSTRG, LENGTH, CCSNAME und TOCCS an. Weitere Angaben, wie DEFAULT oder OUTSTRG, werden ignoriert.

- Diverse Längenberechnungen von vorgegebenen UTF-8 (UTFE)-Zeichenketten

  Aus der Anzahl von Bytes der Ausgangszeichenkette wird die entsprechende Zeichenanzahl ermittelt, oder aus einer angegebenen Zeichenanzahl die entsprechende Byte-Anzahl der Ausgangszeichenkette.

  Geben Sie die Parameter INSTRG, LENGTH und CCSNAME an. Weitere Angaben, wie DEFAULT oder OUTSTRG, werden ignoriert.

  An der durch LENGTH angegebenen Adresse wird ein Wort erwartet, wobei das erste Halbwort (HW) die Byte-Anzahl enthält und das zweite Halbwort die entsprechende Zeichenanzahl und umgekehrt. Es gilt:

  – 1. HW ≠ 0 und 2. HW = 0: Im 2. HW wird die Anzahl von Zeichen hinterlegt, die der im 1. HW angegebenen Anzahl von Bytes entspricht. Im 1. HW wird die Anzahl von Bytes hinterlegt, die der im 2. HW angegebenen Anzahl von Zeichen entspricht.

  – 1. HW = 0 und 2. HW ≠ 0: Die Anzahl von Bytes, die im 2. HW angegeben ist, wird im 1. HW hinterlegt.

  – 1. HW ≠ 0 und 2. HW ≠ 0: Fehler !

  – 1. HW = 0 und 2. HW = 0: Im 1.HW wird die Anzahl von Bytes der Ausgangszeichenkette hinterlegt. Im 2. HW wird die Anzahl von Zeichen hinterlegt.

– Die Längenangabe für die Ausgangszeichenkette ist die höchste Bearbeitungslänge in Bytes.
  Wenn Sie im 2. HW eine Zeichenanzahl angeben, der eine Byte-Anzahl entspricht, die über diese höchste Bearbeitungslänge hinausgeht, wird nur bis zu dieser Länge ermittelt.

Beispiel

Gegeben sei die Zeichenkette ßiölÄ.

Hexadezimaldarstellung in ISO8859-1	DF69F66CC4
UTF-16-Darstellung	00DF006900F6006C00C4
Zeichenanzahl	5

Die Umwandlung in UTF-8 ergibt: C39F 69 C3B6 6C C384. Das sind 8 Bytes.

Ausgangszeichenkette in UTF-8 mit Längenfeld: 0x00080000C39F69C3B66CC384.

Wenn Sie zur Byte-Anzahl 5 die entsprechende Anzahl von Zeichen ermitteln wollen, versorgen Sie das Wort, auf das LENGTH zeigt, mit 0x00050000. Nach Aufruf von NLSCNV erhalten Sie 0x00050003, da der Byte-Anzahl 5 genau 3 Zeichen der Ausgangszeichenkette entsprechen. Bei Angabe einer Byte-Anzahl von 4 erhalten Sie 0x00040003, da das unvollständige Zeichen mitgezählt wird.

Wenn Sie umgekehrt die Byte-Anzahl wissen wollen, die der Zeichenanzahl 3 entspricht, versorgen Sie das Wort, auf das LENGTH zeigt, mit 0x00000003. Nach Aufruf von NLSCNV erhalten Sie in diesem Feld den Wert 0x00050003. Wenn Sie als Zeichenanzahl beispielsweise 7 angeben, wird 8 als entsprechende Byte-Anzahl zurückgeliefert, da dies die im Längenfeld der Ausgangszeichenkette angegegebene maximale Länge ist.

4.3.4 Hinweise zur Programmierung

SVC-freier Ansprung aus TU

Der Makro NLSCNV (NATIONAL LANGUAGE SUPPORT: CONVERSION FUNCTION) kann - wie bisher - über einen Supervisor-Call (SVC) aufgerufen werden.

Wenn Sie die Konvertierungsschnittstelle NLSCNV häufig aufrufen, wobei jedes Mal eine komplette SVC-Behandlung durchgeführt wird, können Performance-Probleme auftreten.

Um dem vorzubeugen, gibt es ab XHCS V2.0 die Möglichkeit, die gewünschte Funktion über eine Call-Schnittstelle, d.h. via BASR, aufzurufen.

Auch wenn die gewünschte Funktion nicht über SVC, sondern über BASR angesprungen wird, ist nicht in allen Fällen der gesamte Ablauf SVC-frei.

Um SVCs einzusparen, ist Folgendes vorgesehen:

– Wenn zwei- oder mehrmals hintereinander dieselbe Aktion angefordert wird, z.B. Konvertierung von EDF041 nach ISO88591, merkt sich XHCS die dafür notwendige(n) Tabelle(n) und besorgt sie nicht noch einmal.

  Die SVCs werden nur beim ersten Aufruf einer bestimmten Aktion abgesetzt.

– Wenn Sie mit dem Operanden TABADDR (siehe Seite 101) bereits die "eigentliche" Umsetzungstabelle mitgeben, muss diese nicht mehr von XHCS besorgt werden.

Zur Realisierung des Ansprungs der gewünschten Funktion mit BASR wird das LLM GNLCNV bereitgestellt, das der Anwender zu seinem Benutzerprogramm dazubinden kann. Das LLM GNLCNV ist in der Modulbibliothek SYSLNK.XHCS-SYS.020.TU enthalten.

Alternativ zu der Möglichkeit, GNLCNV fest einzubinden, bietet XHCS den Ansprung zu einem reentrant-fähigen Adaptermodul an, der seinerseits das LLM GNLCNV mit BIND dynamisch nachlädt.

Dieser Adaptermodul liegt als R-Modul vor und kann zum aufrufenden Programm dazugebunden werden. Er liegt somit für alle Aufrufer des shared Code an der gleichen Adresse. Der Adaptermodul GNLADPT ist in der Bibliothek SYSOML.XHCS-SYS.020 enthalten.

Im Einzelnen führt dieser Adaptermodul GNLADPT folgende Aktionen durch:
– Register des Aufrufers in den vom Aufrufer bereitgestellten Speicherbereich sichern
– mit REQM Speicher für die "Automatischen Daten", z.B. die BIND-Parameterliste, anfordern
– Installationspfad ermitteln
– das LLM GNLCNV mit BIND in den Klasse-6-Speicher nachladen
– Entry GNLCNVC anspringen
– Standard-Returncode bei Misserfolg setzen

Der Aufrufer muss einen taskspezifischen "Kontrollblock" mitliefern, in dem GNLADPT die vom BIND gelieferte Adresse hinterlegen kann. Außerdem wird dort vermerkt, ob der BIND bereits erfolgt ist, damit das Nachladen nur beim ersten Aufruf (und nicht mehr bei allen nachfolgenden Aufrufen) erfolgt. Außerdem muss der Kontrollblock auch die Adresse der eigentlichen NLSCNV-Parameterliste enthalten.

Die Beschreibung der dafür neu eingeführten Operanden MODE und XPAND finden Sie auf Seite 90.

Eindeutig umkehrbare Konversion für alle existierenden 7-/8-bit-Zeichensätze

Bei einigen der unterstützten ISO-Codes und den entsprechenden EBCDIC-Codes gibt es Definitionslücken, z.B. bei ISO88597 bzw. EDF047.

Wenn eine zu konvertierende Zeichenkette Zeichen enthält, die im Quell-CCS nicht definiert sind, können in der Zielzeichenkette Ersatzzeichen enthalten sein. Da bei der Konvertierung alle nicht definierten Code-Positionen auf ein- und dasselbe Ersatzzeichen abgebildet werden, ist bei einer Rückumwandlung nicht mehr ersichtlich, aus welchem Zeichen das Ersatzzeichen resultiert, so dass die Ausgangszeichenkette nicht mehr rekonstruiert werden kann.

Um die Original-Eingabezeichen bei der Rückumwandlung wieder einsetzen zu können, ist es notwendig, diejenigen Zeichen in der Eingabezeichenkette, die nicht im Quell-CCS definiert sind, in *unterschiedliche* Ersatzzeichen umzuwandeln.

Wenn Sie beim Konvertieren kein Einsetzen jeweils ein- und desselben Ersatzzeichens, sondern eine eindeutig umkehrbare Konversion haben wollen, geben Sie für den Operanden DEFAULT den Wert *SPECIAL an.

Wenn in der Eingabezeichenkette mindestens ein Zeichen vorkommt, das im Quell-CCS nicht definiert ist und gemäß der "Spezialabbildung" umgesetzt werden muss, wird der Returncode X'00010024' gesetzt.

Details dazu entnehmen Sie der Beschreibung des Operanden DEFAULT ab Seite 97 sowie dem Abschnitt „Returncodes“ auf Seite 110.

Länge der Ein- und Ausgabezeichenketten

- Die Ein- und Ausgabezeichenketten können teilweise oder vollständig (INSTRG = OUTSTRG) den gleichen Speicherbereich belegen, falls die Länge der Ausgabezeichenkette gleich der Länge der Eingabezeichenkette ist, d.h. falls
  - ACTION = CONVERT, und weder der Ausgangscode (CCSNAME) noch der Zielcode (TOCCS) eine Unicode-Variante ist,
  - ACTION = TOUPPER / TOLOWER
    unabhängig davon, ob der angegebene CCS eine Unicode-Variante ist oder nicht.

  In allen anderen Fällen müssen Sie davon ausgehen, dass die Länge der Ausgabezeichenkette nicht gleich der Länge der Eingabezeichenkette ist. In diesem Fall dürfen sich Eingabe- und Ausgabezeichenkette nicht überlappen.

- Die tatsächliche Länge der Eingabezeichenkette muss vom Aufrufer festgelegt werden.

  Zusätzlich zur Angabe der Ein- und Ausgabezeichenketten im V-Format, haben Sie die Möglichkeit die Zeichenketten mithilfe der Operanden INLEN und OUTLEN zu handhaben. Sie enthalten jeweils die Adresse eines Wortes, das die Länge der Zeichenkette enthält.

  Die maximale Länge beträgt
  - 65536 (= X'10000') Bytes, wenn die Länge der Eingabezeichenkette mit INLEN angegeben wird.
  - 32767 (= X'7FFF') Bytes, wenn die Eingabezeichenkette im V-Format angegeben wird.

- Die Vorgabe eines Maximalwertes für die Länge der Ausgabezeichenkette ist nur mit dem neuen Operanden OUTLEN möglich.
  Bei Verwendung des V-Formats bleibt aus Kompatibilitätsgründen alles beim Alten.

- Nachdem erfolgreich umgewandelt wurde, wird die Gesamtlänge der Ausgabezeichenkette von XHCS eingetragen.

  Die Länge der Ausgabezeichenkette hängt von der angegebenen Aktion und dem gewünschten Ausgangs- bzw. Zielcode ab.

  Wenn der Aufrufer mit dem Operanden OUTLEN eine maximale Ausgabelänge vorgegeben hat, und die Ausgabezeichenkette länger als diese vorgegebene Länge wird, werden die Ausgabedaten abgeschnitten, und der Returncode X'00 00 0084' ("truncated") wird zurückgeliefert.

  ACHTUNG!

  Wenn der Aufrufer keine maximale Ausgabelänge vorgegeben hat, muss er sicherstellen, dass der Speicherplatz für die Ausgabezeichenkette lang genug ist, um die vollständige, umgewandelte Zeichenkette aufzunehmen. Andernfalls gehen Benutzerdaten verloren.

  Hinweise zur je nach Funktion maximal möglichen Länge der Ausgabekette entnehmen Sie dem Abschnitt „Funktionsübersicht“ auf Seite 102.

4.3.5 Returncodes

SUBCODE 2 | 1		MAINCODE 2 1		Bedeutung
X' 00'	X' 00'	X' 00'	X' 00'	erfolgreiche Bearbeitung; Umwandlung erfolgt
X' 00'	X' 00'	X' 00'	X' 24'	Umwandlung erfolgreich durchgeführt. Achtung: Einige Zeichen des Ausgangscodes haben kein Äquivalent im Zielcode
X' 00'	X' 00'	X' 00'	X' 84'	Ausgabe abgeschnitten. Die Länge der Ausgabezeichenkette überschreitet die mit dem Operanden OUTLEN angegebene maximale Ausgabelänge, und Ausgabedaten wurden abgeschnitten.
X' 00'	X' 01'	X' 00'	X' 04'	Name des Codes (CCSN) unbekannt
X' 00'	X' 01'	X' 00'	X' 20'	Ausgangscode (CCSNAME) und Zielcode (TOCCS) sind inkompatibel
X' 00'	X' 01'	X' 00'	X' 24'	Umwandlung erfolgreich durchgeführt. Einige Zeichen der Eingabezeichenkette sind nicht im Ausgangscode definiert oder haben kein Äquivalent im Zielcode. oder Der Quell-CCS ist eine Unicode-Variante und einige Zeichen in der Eingabezeichenkette gehören nicht zu den von XHCS unterstützten Unicode-Zeichen. In der Ausgabezeichenkette sind diese Zeichen durch das angegebene Ersatzzeichen ersetzt worden.
X' 00'	X' 01'	X' 00'	X' 28'	Umwandlung abgebrochen. Einige Zeichen der Eingabezeichenkette sind nicht im Ausgangscode definiert oder haben kein Äquivalent im Zielcode. oder Der Quell-CCS ist eine Unicode-Variante und einige Zeichen in der Eingabezeichenkette gehören nicht zu den von XHCS unterstützten Unicode-Zeichen. Die Verarbeitung wurde abgebrochen, weil DEFAULT=*NO gesetzt war.
X' 00'	X' 01'	X' 00'	X' 80'	Fehler bei der Speicherzuordnung

SUBCODE 2 | 1		MAINCODE 2 1	Bedeutung
X' 00'	X' 01'	X' 00' X' 88'	Länge der Zeichenkette ungültig. – Die bei INLEN angegebene Länge ist < 0 oder > 65536. – Die bei OUTLEN angegebene Länge ist < 0. – Die in den ersten 2 Bytes beim V-Format angegebene Länge ist < 4 oder > 32767. – Ein- und Ausgabestring überlappen sich und Ausgangs- und/oder Zielcode ist eine Unicode-Variante und ACTION ≠ TOUPPER oder TOLOWER.
X' 00'	X' 01'	X' 00' X' 8C'	Fehler bei der Ausrichtung auf Halbwortgrenze
X' 00'	X' 01'	X' 00' X' C4'	Ungültige Angaben bei der Ersatzzeichen-Angabe. – Das angegebene Ersatzzeichen ist nicht im Ziel-CCS definiert (bei DEFAULT=*UTF16). – Die beim Operanden DEFLEN angegebene Länge passt nicht zum Ziel-CCS (bei DEFAULT= *TARGET). – Ungültiger Wert bei DEFAULT – DEFAULT=*SPECIAL und die Aktion ist keine Konvertierung eines 7-/8-bit-Codes nach UTF-16 oder umgekehrt.
X' 00'	X' 01'	X' 00' X' C8'	Fehler bei ACTION=LENGTH – Weder der Quell-CCS noch der Ziel-CCS ist UTF-8 oder UTFE. – Beide Angaben beim Operanden LENGTH sind ungleich 0.
X' 00'	X' 01'	X' FF' X' FF'	Die angeforderte Funktion wird nicht unterstützt. Nicht behebbarer Fehler.
X' 00'	X' 02'	X' FF' X' FF'	Die angeforderte Funktion ist nicht verfügbar. Nicht behebbarer Fehler. Mögliche Ursache: XHCS-SYS fehlt in der aktuellen Konfiguration
X' 00'	X' 03'	X' FF' X' FF'	Die angegebene Version der Schnittstelle wird nicht unterstützt (falsche Versionsangabe im Standard-Header). Nicht behebbarer Fehler.
X' 00'	X' 04'	X' FF' X' FF'	Parameterliste ist nicht auf Wortgrenze ausgerichtet.
X' xx'	X' 20'	X' 00' X' C0'	fehlerhafte Code-Tabellen

SUBCODE 2 | 1		MAINCODE 2 1	Bedeutung
X' 00'	X' 41'	X' FF' X' FF'	Das Subsystem ist nicht vorhanden; es muss explizit erzeugt werden.
X' 00'	X' 42'	X' FF' X' FF'	Der aufrufende Ablauf ist mit dieser Schnittstelle nicht konnektiert; er muss explizit konnektiert werden.
X' 00'	X' 81'	X' FF' X' FF'	Subsystem zurzeit nicht verfügbar.
X' 00'	X' 82'	X' FF' X' FF'	Subsystem im DELETE- oder HOLD-Zustand.

Bedeutung der einzelnen Felder:

SUBCODE 1 Anzeigen der Fehlerklasse

= X'00': Erfolgreiche Bearbeitung

≠ X'00': Fehler, mit Ausnahme beim MAINCODE X'00' X'24'

SUBCODE 2 wird immer auf den Wert X'00' gesetzt, mit Ausnahme beim MAINCODE X'00' X'C0'.

MAINCODE 1 Fehlerinformationen für das Anwendungsprogramm. Das Anwendungsprogramm kann damit Bedienungsfehler abfangen.

MAINCODE 2 wird zurzeit nicht belegt und auf den Wert X'00' gesetzt. Wenn im Standard-Header Fehler sind, die der Selbstidentifikation des Produktes dienen (z.B. falsche Version), wird der Wert auf X'FF' gesetzt.

Beispiel

```
EXMPL    START
*
         BALR   10,0
         USING  *,10
*
*
         NLSCNV MF=E,PARAM=PL ------------------------------------(01)
*
*
         TERM
*
*
PL       NLSCNV MF=L,CCSNAME=ISO88591,TOCCS=EDF041,INSTRG=INS,   C
                OUTSTRG=INS ------------------------------------(02)
*
*
*        Speicherbereich für Ein- und Ausgabedaten
*
         DS     0H
INS      DC     Y(INSTEND-INS)-----------------------------------(03)
RES      DC     Y(0) ---------------------------------------------(04)
TXT      DC     C'Umzuwandelnder Text'
INSTEND  EQU    *
*
*
         END
```

(01) XHCS wird mit einer von MF=L initialisierten Parameterliste aufgerufen.

(02) Die Parameterliste wird deklariert und initialisiert. Der westeuropäische ISO 8859-1-Code wird in einen westeuropäischen EBCDI-Code (EBCDIC.DF.04-1) umgewandelt. Die Zeichenkette (V-Format) beginnt auf der Adresse 'INS'.

(03) Länge des V-Formats

(04) reserviert

5 Definition von Code-Tabellen

Die wesentlichen Daten, mit denen das Produkt XHCS arbeitet, sind Beschreibungen der unterstützten Codes, die vom Systembetreuer definiert und verwaltet werden.

Diese Beschreibungen sind in Form von Tabellen, den so genannten Code-Tabellen, im Modul GNLMTAB enthalten, das durch die Übersetzung eines Assembler-Quellcodes entsteht. Der Quellcode von GNLMTAB ist in der Bibliothek SYSSRC.XHCS-SYS.020.GNLMTAB enthalten.

5.1 Code-Namen

Die einzelnen Codes werden durch ihren jeweiligen Code-Namen (CCSN) gekennzeichnet. Der Name darf maximal 8 Byte lang sein. Seine Syntax entspricht dem SDF-Datentyp <name 1..8>. Er kann die Zeichen A-Z, 0-9, @, #, $ enthalten, wobei das erste Byte ein alphabetisches Zeichen (A-Z, @, #, $) sein muss. Wenn Sie genormte Codes verwenden, benutzen Sie bitte die Namen aus dem Kapitel „Tabellen“ auf Seite 145. Sie sind gemäß ihrer Kompatibilität gruppiert. Genormte Codes sollten nicht redefiniert werden.

5.1.1 Standard-System-Code

Der Standard-System-Code ist der vom System angebotene Zeichensatz.

Der Standardzeichensatz ist ein erweiterter Zeichensatz. Sie können auswählen, ob Sie mit dem Standard-Anwender-Code oder dem Standard-System-Code arbeiten wollen. Der Standard-System-Code ist als ein Klasse-2-Systemparameter definiert. Dieser Systemparameter heißt HOSTCODE. Der Standardwert von HOSTCODE ist EDF03IRV.

5.1.2 Standard-Anwenderzeichensatz

Der Systembetreuer kann jedem Anwender einen Standard-Anwenderzeichensatz zuordnen. Der Standard-Anwenderzeichensatz ist ein erweiterter Zeichensatz.

Über die Kommandos ADD-USER oder MODIFY-USER kann der Systembetreuer jedem Anwender einen Standard-Anwenderzeichensatz zuordnen. Über das Kommando SHOW-USER-ATTRIBUTES wird der Standard-Anwenderzeichensatz zurückgeliefert.

Rufen Sie XHCS über den Parameter *USRDEF auf, wird der Standard-Anwenderzeichensatz benutzt. Wird kein Standard-Anwenderzeichensatz zugeordnet, wird stattdessen der System-Standard benutzt.

Es wird empfohlen, als Standard-System-Code EDF03IRV zu benutzen. Dadurch werden Warnungen an Anwender von 7-bit-Terminals vermieden. Anwender von 8-bit-Terminals sollten als Anwender-Standard einen 8-bit-Code angeben. Anwender von 7-bit-Terminals sollten keinen Anwender-Standard angeben.

VTSU unterstützt als Standard-Anwender-Codes nur EBCDIC-Standards, vollständig kompatible Codes und den Code EDF03IRV (Standard 7-bit-Modus).

5.2 Gruppierung kompatibler Codes

Die Umwandlung eines Codes in einen anderen ist nur zwischen kompatiblen Codes möglich. Deswegen muss die Information über die Kompatibilität von Codes zusammen mit ihren Daten gespeichert werden.

Die einzelnen Codes sind zu Gruppen kompatibler Codes zusammengefasst, die über so genannte (pseudo) ISO-Code-Variantennummer (-n) identifiziert werden. Diese Nummern weisen ursprünglich auf die Nummern der entsprechenden ISO 8859-Variante hin; dies gilt nicht mehr, da die Code-Tabellen für Arabisch, Persisch und Nord-Afrikanisch (Französisch/Arabisch) einem anderen Nummernschema folgen.

Einen Sonderstatus in jeder Code-Gruppe hat der Referenzcode. Alle Codes derselben Code-Gruppe werden durch den Referenzcode definiert. Der Referenzcode enthält alle Zeichen, die in den anderen Codes derselben Gruppe vorkommen. Er heißt deshalb auch Gruppenreferenzcode. Als Referenzcode müssen Sie einen EBCDIC.DF.04-n-Code wählen. Zum Initialisierungszeitpunkt des Subsystems wird überprüft, ob dieser Code als Referenzcode angegeben wurde. Nur in diesem Fall wird das Subsystem geladen.

Die Umwandlungstabelle jedes einzelnen Codes der Gruppe in den zugehörigen Referenzcode ist gespeichert verfügbar (siehe Seite 118). Für Umwandlungen in andere Codes derselben Gruppe oder eines Gruppenreferenzcodes in einen anderen Code müssen Umwandlungstabellen erstellt werden. XHCS benutzt dazu bereits bestehende Tabellen.

Eine Code-Gruppe kann Codes enthalten, die zu einem ISO 8859-Code voll kompatibel (z.B. EBCDIC.DF.04-n) oder begrenzt kompatibel sind (z.B. EBCDIC.DF.03-DRV oder ISO646-IRV). Begrenzt kompatible Codes bestehen aus einer Teilmenge der Zeichen des entsprechenden ISO 8859-Codes. Da eine Umwandlung von begrenzt kompatiblen Codes in den Referenzcode immer möglich ist, während eine Umwandlung vom Referenzcode in die begrenzt kompatiblen Codes nicht für alle Zeichen des Gruppenreferenzcodes möglich ist, werden begrenzt kompatible Codes gekennzeichnet.

Beispiel

Jedes unter EBCDIC.DF.03 definierte Zeichen hat ein Äquivalent im ISO 8859. Die mit Akzent versehenen Buchstaben im ISO 8859-1 haben dagegen kein Gegenstück im EBCDIC.DF.03-IRV sondern in dessen Code-Erweiterung EBCDIC.DF.04-1.

i Einige Produkte benutzen die Kennzeichnung von begrenzt kompatiblen Codes zur Feststellung, ob die Daten an einem 7-bit-Terminal dargestellt werden können. Es wird deshalb geraten, 8-bit-Codes nicht als begrenzt kompatible Codes zu definieren.

5.3 Tabellenstruktur

Jeder in einem System benutzte 8-bit-Code wird durch sechs Tabellen beschrieben. Dabei haben die ersten fünf Tabellen eine Größe von je 256 Bytes und die sechste Tabelle eine Größe von 512 Bytes. Die Tabellen sind in einer Datenstruktur zusammengefasst, wobei die Tabellen des Standard-System-Codes als Erstes in der Datenstruktur definiert sind. Zur Unterstützung von 8-bit-Datenstationen ist der Standard-Code ein erweiterter Zeichensatz.

Es gibt:

– eine Tabelle zur Umwandlung in den Referenzcode der Gruppe
– eine Tabelle zur Umwandlung von Klein- in Großbuchstaben
– eine Sortiertabelle
– eine Tabelle mit Eigenschaften (in der 8 verschiedene Eigenschaften gespeichert sind)
– eine Tabelle zur Umwandlung vom Referenzcode
– eine Tabelle zur Umwandlung des 8-bit-Codes nach Unicode

Die Tabellen müssen den nachfolgend beschriebenen Regeln entsprechen. XHCS überprüft nicht, ob die Tabellen diesen Regeln entsprechen.

5.3.1 Umwandlungstabelle in den Referenzcode

Diese Tabelle ermöglicht die Umwandlung eines Codes in den entsprechenden Referenzcode der Gruppe. Für jedes Zeichen des umzuwandelnden Codes enthält die Tabelle den Code des entsprechenden Zeichens im Gruppenreferenzcode. Enthält der Code ein undefiniertes Zeichen, muss dieses in der Umwandlungstabelle auf NIL (X'00') abgebildet werden. Für den Referenzcode muss die abgebildete Position dem Zeichenwert entsprechen (d.h. Byte Nr. i hat den Zeichenwert i).

Den Konventionen entsprechend verwendet das BS2000 eine Reihe von Basis-Symbolen, die in allen definierten EBCDIC-Varianten denselben Zeichenwert haben und somit nicht redefiniert werden können. Bei diesen Symbolen handelt es sich um den EBCDIC-Kern und um die folgenden Zeichen:

#	Nummernzeichen
@	kommerzielles At
$	Dollar
!	Ausrufezeichen
"	Doppelapostroph

XHCS überprüft nicht, ob das zu redefinierende Zeichen ein Basis-Symbol ist.

Zeichenwerte kleiner X'40' sind in EBCDIC-Zeichensätzen für Steuerzeichen reserviert und sollten deshalb unverändert bleiben.

5.3.2 Tabelle zur Umwandlung von Klein- in Großbuchstaben

Diese Tabelle bildet jeden Kleinbuchstaben auf der Position des entsprechenden Großbuchstaben ab. Die übrigen Zeichen werden auf sich selbst abgebildet.

5.3.3 Sortiertabelle

Diese Tabelle beschreibt die Sortierfolge der einzelnen Zeichen. Die Umwandlung muss eindeutig sein, d.h., zwei Zeichen dürfen nicht in denselben Zeichenwert umgewandelt werden. Nur so ist auch eine umgekehrte Umwandlung möglich.

5.3.4 Tabelle der Zeicheneigenschaften

In der Tabelle der Zeicheneigenschaften werden pro Zeichen acht Eigenschaften gespeichert. Jede Eigenschaft ist als ein Bit codiert. Die Eigenschaft trifft zu, wenn das Bit auf ’1’ gesetzt ist. Die Eigenschaft trifft nicht zu, wenn das Bit auf ’0’ gesetzt ist.

Jedes Byte wird in der Tabelle wie folgt dargestellt, wobei Bit 0 das niederwertigste Bit ist:

Bit 0: Das Zeichen ist ein Buchstabe.

Bit 1: Das Zeichen ist eine Ziffer.

Bit 2: Das Zeichen ist ein arithmetisches Zeichen.

Bit 3: Das Zeichen ist ein Sonderzeichen.

Bit 4: Das Zeichen ist ein Kleinbuchstabe.

Bit 5: Das Zeichen gehört zum EBCDIC-Kern.

Bit 6: Das Zeichen ist im Code definiert.

Bit 7: Das Zeichen ist darstellbar/druckbar.

5.3.5 Tabelle zur Umwandlung vom Referenzcode

Diese Tabelle ermöglicht die Umwandlung des Referenzcodes in den entsprechenden Code, der durch die sechs Tabellen beschrieben wird. Für jedes Zeichen des Gruppenreferenzcodes enthält die Tabelle den Code des entsprechenden Zeichens im Zielcode.

In jedem standardmäßig gelieferten Makro, der einen 8-bit-Code definiert, ist die Tabelle zur Umwandlung vom Referenzcode leer, das heißt, sie enthält 256 Zeichen mit dem Wert X’00’. Der Systembetreuer kann diese Tabelle umdefinieren, indem er GNLMTAB modifiziert und neu übersetzt.

Solange die fünfte Tabelle leer ist, wird sie von XHCS beim Laden des Subsystems automatisch erstellt. Dies garantiert die Konsistenz mit der Tabelle zur Umwandlung in den Referenzcode der Gruppe (erste Tabelle).

Wenn der Systembetreuer die Tabelle zur Umwandlung vom Referenzcode ändert, benutzt XHCS-SYS diese ohne Konsistenzcheck gegenüber der ersten Tabelle. Normalerweise sollten die erste und die fünfte Tabelle invers sein.

Beispiel

EDF041 ist der Referenzcode der Code-Gruppe, die die Codes ISO88591, EDF041 und EDF04DRV enthält.

Für ISO88591 definiert die erste Tabelle die Umwandlung in den Referenzcode EDF041. An der Adresse X'31' (Codierung des Zeichens '1' in ASCII) steht der Wert X'F1' (Codierung des Zeichens '1' in EBCDIC).

Die fünfte Tabelle definiert die Umwandlung von EDF041 nach ISO88591. An der Adresse X'F8' (Codierung des Zeichens '8' in EBCDIC) steht der Wert X'38' (Codierung des Zeichens '8' in ASCII).

5.3.6 Tabelle zur Unicode-Abbildung

Diese Tabelle ordnet jedem definierten Punkt einer 8-bit-Code-Tabelle die entsprechende Unicode-Position zu. Sie ist derzeit für ISO8859-1/-2/-3/-4/-5/-7/-9/-15 und für den Pseudo-ISO-Code ISOEXT1 definiert, kann aber für weitere ISO-8-bit-Codes erweitert werden. Für kompatible 8-bit-Codes, also insbesondere für EDF04-1/-2/-3/-4/-5/-7/-9/-15, wird die Tabelle bei Bedarf aus den entsprechenden
ISO-8-bit-Tabellen berechnet.

Die Tabelle hat eine Größe von 2*256 Bytes: Jedem definierten Wert des 8-bit-Codes wird das entsprechende Unicode-Zeichen mit einer Größe von 2 Bytes zugeordnet.

5.4 Erstellung und Modifikation von Code-Tabellen

Alle XHCS-Tabellen sind in dem Modul GNLMTAB gespeichert. Dieses Modul wird mithilfe des Makros NLSCTAB aus einem Assemblerquellcode erzeugt und muss in die Modulbibliothek SYSLNK.XHCS-SYS.020 oder in eine benutzereigene Bibliothek aufgenommen werden (siehe Abschnitt „Installation benutzerdefinierter Tabellen“ auf Seite 130). Der Makro NLSCTAB generiert für maximal 99 Codes alle Tabellen, durch die jeder einzelne Code unterstützt wird. Der Quellcode von GNLMTAB ist in der Bibliothek SYSSRC.XHCS-SYS.020.GNLMTAB enthalten. Darüber hinaus kann der Systembetreuer mithilfe des Makros NLSCTAB eigene Code-Tabellen erzeugen, indem er Standard-Code-Tabellen modifiziert. Die Modifizierung der Standard-Code-Tabellen ist vergleichbar mit der der Erstellung von benutzerspezifischen FHS-Code-Tabellen (Makro MGCTS). Das modifizierte Modul GNLMTAB kann mit dem Kommando /ADD-CODE-TABLES ins System eingebunden werden (siehe Abschnitt „Neue Code-Tabellen dynamisch hinzufügen: ADD-CODE-TABLES“ auf Seite 134). Ansonsten kann es erst nach erneutem Laden des Subsystems genutzt werden, d.h. nach dem nächsten Systemstart.

Wenn Sie Tabellen modifizieren, beachten Sie, dass es bei einer Dateiübertragung mit anderen BS2000-Rechnern zu Fehlern kommt, wenn diese nicht die gleichen Tabellen benutzen. Ebenso müssen Sie die Regeln und Konventionen, die auf Seite 131 beschrieben sind, beachten, unter anderem, dass Sie keine Zeichen ändern, die für das System von Bedeutung sind (z.B. Zeichen der Kommandosyntax). Um Fehler dieser Art zu vermeiden, sollten Sie die standardmäßig gelieferten erweiterten Zeichensätze nutzen und die Tabellenmodifikation als eine zusätzliche Eigenschaft von XHCS betrachten, die nur in speziellen Fällen (z.B. aus Kompatibilitätsgründen) genutzt wird.

Aufgrund der Unicode-Unterstützung muss in einem solchen Makro NLSCTAB zusätzlich zu den bisher schon bekannten Tabellen eine Tabelle zur Abbildung vom 8-bit-Code nach Unicode definiert werden. Diese Tabelle ist 2 * 256 Bytes groß und befindet sich hinter der inversen Umwandlungstabelle. In ihr wird jedem Zeichen des 8-bit-Codes die entsprechende Unicode-Position zugeordnet.

Der Makro NLSCTAB sollte nur mit dem H-Assembler assembliert werden, da seine Parameterliste für den F-Assembler zu groß werden kann (die maximale Parameterlänge beträgt 127 Zeichen). Als Alternative für den F-Assembler werden die Makros NLSHEAD und NLSCCS angeboten. Beim H-Assembler ist die Parameterliste auf 1020 Zeichen limitiert. Bei der Generierung sehr großer Tabellenmodule reicht dies u.U. nicht aus.

5.4.1 Generierung von Tabellensätzen

Im folgenden Abschnitt wird die Generierung von Tabellensätzen dargestellt.

Makro NLSCTAB

Aufruf:

NLSCTAB ((CCSN,isovar,refcode,fullcode,type,deftab,macroname),...)

Parameterbeschreibung

CCSN	Name des zu definierenden Codes.	
isovar	(pseudo) ISO-Code-Variantennummer des Codes (hexadezimal).	
refcode	Anzeige des Referenzcodes.	
	*REF	für den Referenzcode der Gruppe
	*NORMAL	für die anderen Codes
fullcode	Anzeige der Kompatibilität (vollständig kompatibel/ begrenzt kompatibel) einzelner Codes (siehe Abschnitt „Gruppierung kompatibler Codes“ auf Seite 116).	
	*FULL	Code ist vollständig kompatibel mit dem Referenzcode
	*RESTR	Code ist begrenzt kompatibel mit dem Referenzcode
type	Anzeige der Code-Art.	
	*EBCDIC	Der Code ist ein EBCDIC-Code
	*ISO	Der Code ist ein ISO-Code
deftab	Name des Ausgangs-Tabellensatzes, der vom Anwender nach Bedarf modifiziert werden kann. Um vom Makro NLSCTAB unabhängig zu sein, stehen diese Tabellen als einzelne Makros zur Verfügung. Sie befinden sich zusammen mit dem Makro NLSCTAB in der Bibliothek SYSLIB.XHCS-SYS.020.	
macroname	Name des Redefinitionsmakros	

5.4.2 Die Makroaufrufe NLSHEAD und NLSCCS

Die Makros NLSHEAD und NLSCCS ermöglichen für maximal 99 Codes Tabellendefinitionen, sowohl mit dem F-Assembler, wie auch mit dem H-Assembler. Sie arbeiten ähnlich wie der Makro NLSCTAB.

Makro NLSHEAD

Der Makro NLSHEAD baut den Header des Moduls GNLMTAB auf. Sein einziger Parameter ist die Anzahl der Codes, die deklariert werden soll.

Aufruf:

NLSHEAD NCCS=Code-Anzahl

Beispiel

```
NLSHEAD   NCCS = 3
```

Makro NLSCCS

Der Makro NLSCCS muss für jede Definition eines Codes einmal aufgerufen werden. Als Argument nimmt der Makro NLSCCS eine Parameterliste, die mit einer Unterliste des Makros NLSCTAB identisch ist.

Redefinitionsmakro (von Systembetreuer zu definieren)

```
         MACRO
&MACNAM  makroname ------------------------------------------------(1)
         GBLC   &GVAR1
*
&MACNAM  EQU    *
*
         ORG    DEFTB&GVAR1.n+X'dist' ---------------------------------(2)
         DC     X'Hexcode' --------------------------------------------(3)
         .
         .
         .
         MEND
```

(1) Identisch mit dem Makronamen im NLSCTAB

(2) DEFTB&GVAR1.n Adressen der Basistabellen

n=0 Umwandlungstabelle
n=1 Tabelle zur Umwandlung von Klein- in Großbuchstaben
n=2 Sortiertabelle
n=3 Eigenschaftstabelle
dist Abstand vom Tabellenanfang

(3) Neuer Wert der/des angegebenen Bytes

Die Makrovariable GVAR1 wird von NLSCTAB (oder NLSCCS) dazu verwendet, den Code zu indizieren, der redefiniert werden soll.

5.4.3 Beispiel zur Generierung von Tabellensätzen

Dieses Beispiel zeigt die Generierung von Tabellen für ISO 8859-1, EBCDIC.DF.04-1 und EBCDIC.DF.03-DRV. Referenzcode und Systemstandardwert für den erweiterten Rechner-Code ist der EBCDIC.DF.04-1. Dieser Standardwert muss in BS2000 V10 in der Datenstruktur an erster Stelle deklariert sein.

Das Vorhandensein vordefinierter Makros wird vorausgesetzt. Diese Makros beschreiben den EBCDIC.DF.04-1, EBCDIC.DF.03-IRV und ISO8859-1.

Folgendes Redefinitionsmakro muss vom Systembetreuer nicht erzeugt werden, da der EBCDIC.DF.03-DRV einer der Standard-Codes ist, die mit XHCS als Makro zusammen ausgeliefert werden.

Redefinitionsmakro

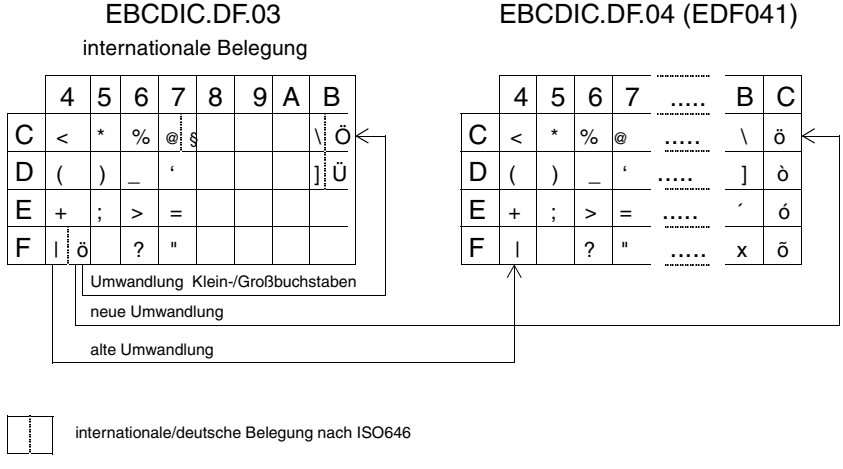
```
         MACRO
&MACNAM  ITOD
         GBLC   &GVAR1
*
*
*  Erstellung von Tabellen für den EBCDIC.DF.03-DRV durch
*  Modifikation der Tabellen des EBCDIC.DF.03-IRV
*
*
&MACNAM  EQU    *
*
*  Redefinition der Umwandlungstabelle:
*  (Referenzcode ist der EBCDIC.DF.04-1)
         ORG    DEFTB&GVAR1.0+X'4F'     * ö
         DC     X'CC'
         ORG    DEFTB&GVAR1.0+X'7C'     * Paragraphen-Zeichen
         DC     X'B5'
         ORG    DEFTB&GVAR1.0+X'BB'     * Ä Ö Ü
         DC     X'63ECFC'
         ORG    DEFTB&GVAR1.0+X'FB'     * ä
         DC     X'43'
         ORG    DEFTB&GVAR1.0+X'FD'     * ü
         DC     X'DC'
         ORG    DEFTB&GVAR1.0+X'FF'     * ß
         DC     X'59'
*
*  Redefinition der Tabelle zur Umwandlung von Klein- in Großbuchstaben:
*
         ORG    DEFTB&GVAR1.1+X'4F'
         DC     X'BC'
```

```
         ORG    DEFTB&GVAR1.1+X'FB'
         DC     X'BB'
         ORG    DEFTB&GVAR1.1+X'FD'
         DC     X'BD'
*
*
*  Redefinition der Sortiertabelle
*  Die Sortiertabelle wird reorganisiert, damit Umlaute auf die
*  entsprechenden Vokale ohne Umlautzeichen und 'ß' auf 'S' folgen.
*  Bei einigen Zeichen wird das Sortiergewicht vertauscht, dadurch bleibt
*  die Tabelle eindeutig.
*
*
         ORG    DEFTB&GVAR1.2+X'43'
         DC     X'76'
         ORG    DEFTB&GVAR1.2+X'4F'
         DC     X'D5'
         ORG    DEFTB&GVAR1.2+X'59'
         DC     X'FF'
         ORG    DEFTB&GVAR1.2+X'5F'
         DC     X'42'
         ORG    DEFTB&GVAR1.2+X'63'
         DC     X'77'
         ORG    DEFTB&GVAR1.2+X'7C'
         DC     X'5F'
         ORG    DEFTB&GVAR1.2+X'A1'
         DC     X'52'
         ORG    DEFTB&GVAR1.2+X'B5'
         DC     X'63'
         ORG    DEFTB&GVAR1.2+X'BB'
         DC     X'97D6F1'
         ORG    DEFTB&GVAR1.2+X'CO'
         DC     X'5D'
         ORG    DEFTB&GVAR1.2+X'CC'
         DC     X'5B'
         ORG    DEFTB&GVAR1.2+X'DO'
         DC     X'5E'
         ORG    DEFTB&GVAR1.2+X'DC'
         DC     X'5CEC'
         ORG    DEFTB&GVAR1.2+X'EO'
         DC     X'6A'
         ORG    DEFTB&GVAR1.2+X'EC'
         DC     X'EF'
         ORG    DEFTB&GVAR1.2+X'FB'
         DC     X'96EDFO'
         ORG    DEFTB&GVAR1.2+X'FF'
         DC     X'E3'
```

```
*
*  Redefinition der Eigenschaftstabelle
*  X'C1': definiert-darstellbar-Großbuchstabe-Buchstabe
*  X'D1': definiert-darstellbar-Kleinbuchstabe-Buchstabe
*
         ORG    DEFTB&GVAR1.3+X'4F'     * ö
         DC     X'D1'
         ORG    DEFTB&GVAR1.3+X'BB'     * Ä Ö Ü
         DC     X'C1C1C1'
         ORG    DEFTB&GVAR1.3+X'FB'     * ä
         DC     X'D1'
         ORG    DEFTB&GVAR1.3+X'FD'     * ü
         DC     X'D1'
         ORG    DEFTB&GVAR1.3+X'FF'     * ß
         DC     X'D1'
*
         MEND
```

Beispiel

Die Redefinition des Buchstabens ö für Umwandlungstabellen und für Tabellen zur Umwandlung von Klein- in Großbuchstabentabellen wird im folgenden Bild dargestellt.



Quellcode für GNLMTAB mit NLSCTAB

```
GNLMTAB CSECT
*
        AMODE   ANY
        RMODE   ANY
*
        MCALL   EDF041
        MCALL   ISO88591
        MCALL   EDF03IRV
        MCALL   ITOD
        NLSCTAB ((EDF041,1,*REF,*FULL,*EBCDIC,EDF041,),             ---(1)
                 (ISO88591,1,*NORMAL,*FULL,*ISO,ISO88591,),         ---(2)
                 (E03GERM,1,*NORMAL,*RESTR,*EBCDIC,EDF03IRV,ITOD))  ---(3)
        END
```

(1) Definition des Standardzeichensatzes EBCDIC.DF.04-1;
Die ISO-Code-Variante hat die Nr.1. Der Code ist Gruppenreferenzcode und enthält einen vollständigen Satz aller, innerhalb der Gruppe verwendbaren Zeichen (notwendige Voraussetzung für den Gruppenreferenzcode!). Der Code ist ein EBCDIC-Code und wird aus dem Standard-Code EDF041 ohne Modifikation erstellt.

(2) Definition des Standardzeichensatzes ISO 8859-1;
Die ISO-Code-Variante hat die Nr.1. Der Code ist kein Gruppenreferenzcode, enthält aber den vollständigen Zeichensatz des Gruppenreferenzcodes. Der Code ist ein ISO-Code und wird aus dem mitgelieferten ISO 8859-1-Code ohne Modifikation erstellt.

(3) Definition einer deutschen Version vom EBCDIC.DF.03, genannt E03GERM;
Die ISO-Code-Variante hat die Nr.1. Der Code ist kein Gruppenreferenzcode und enthält nicht alle Zeichen des Gruppenreferenzcodes. Der Code ist ein EBCDIC-Code und wird aus dem Standard-Code EDF03IRV erstellt, der durch Anwendung des Makros ITOD redefiniert wird.

Quellcode für GNLMTAB mit NLSHEAD und NLSCCS

```
GNLMTAB CSECT
*
        AMODE    ANY
        RMODE    ANY
*
        MCALL    EDF041
        MCALL    ISO88591
        MCALL    EDF03IRV
        MCALL    ITOD
*
        NLSHEAD  NCCS=3
*
        NLSCCS   (EDF041,1,*REF,*FULL,*EBCDIC,EDF041,)              ----(1)
        NLSCCS   (ISO88591,1,*NORMAL,*FULL,*ISO,ISO88591,)          ----(2)
        NLSCCS   (E03GERM,1,*NORMAL,*RESTR,*EBCDIC,EDF03IRV,ITOD)   ----(3)
*
        END
```

(1) Definition des Standardzeichensatzes EBCDIC.DF.04-1;
Die ISO-Code-Variante hat die Nr.1. Der Code ist Gruppenreferenzcode und enthält einen vollständigen Satz aller, innerhalb der Gruppe verwendbaren Zeichen (notwendige Voraussetzung für den Gruppenreferenzcode!). Der Code ist ein EBCDIC-Code und wird aus dem Standard-Code EDF041 ohne Modifikation erstellt.

(2) Definition des Standardzeichensatzes ISO 8859-1;
Die ISO-Code-Variante hat die Nr.1. Der Code ist kein Gruppenreferenzcode, enthält aber den vollständigen Zeichensatz des Gruppenreferenzcodes. Der Code ist ein ISO-Code und wird aus dem mitgelieferten ISO 88591-Code ohne Modifikation erstellt.

(3) Definition einer deutschen Version vom EBCDIC.DF.03, genannt E03GERM;
Die ISO-Code-Variante hat die Nr.1. Der Code ist kein Gruppenreferenzcode und enthält nicht alle Zeichen des Gruppenreferenzcodes. Der Code ist ein EBCDIC-Code und wird aus dem Standard-Code EDF03IRV erstellt, der durch Anwendung des Makros ITOD redefiniert wird.

5.4.4 Installation benutzerdefinierter Tabellen

Der Systembetreuer kann ein benutzerdefiniertes Modul GNLMTAB in einer eigenen Bibliothek oder in der Modulbiliothek SYSLNK.XHCS-SYS.020 installieren, indem er die standardmäßig ausgelieferte Bibliothek überschreibt. Durch die Installation einer eigenen Bibliothek ist der Systembetreuer sicher, dass sein individuell angepasstes Modul nicht durch eine Neuinstallation des Produkts XHCS überschrieben wird. Diese Funktionalität ist zusammen mit dem Produkt IMON verfügbar.

Die benutzereigene Bibliothek muss mit folgendem IMON-Kommando mit XHCS verknüpft werden:

```
SET-INSTALLATION-PATH PATH-NAME=eigene-bibliothek,
       LOGICAL-IDENTIFIER=SYSLNK.USER,
       INSTALLATION-UNIT=XHCS-SYS(VERSION=V02.0A00)
```

Wenn der Systembetreuer verhindern will, dass XHCS eine benutzereigene Bibliothek benutzt, kann er das folgende IMON-Kommando verwenden:

```
SET-INSTALLATION-PATH PATH-NAME=*NONE,
       LOGICAL-IDENTIFIER=SYSLNK.USER,
       INSTALLATION-UNIT=XHCS-SYS(VERSION=V02.0A00)
```

Weitere Informationen zur Produktinstallation finden Sie im Handbuch „IMON (BS2000/OSD)“ [9].

Wenn keine benutzereigene Bibliothek existiert, benutzt XHCS-SYS zum Laden des Tabellenmoduls GNLMTAB die Modulbiliothek SYSLNK.XHCS-SYS.020, wie sie von IMON mit dem logischen Namen SYSLNK installiert wurde.

Wenn XHCS-SYS nicht mit IMON installiert wurde, muss das Modul GNLMTAB in die unter TSOS installierte Modulbibliothek übertragen werden, das heißt in $TSOS.SYSLNK.XHCS-SYS.020.

Wird GNLMTAB nicht in der ausgewählten Bibliothek gefunden, gibt XHCS die Meldung GLN0001 zusammen mit dem Bibliotheksnamen an die Konsole aus.

Änderungen (zum Beispiel an der Bibliothek, in der die Tabellen enthalten sind) werden erst nach dem Neuladen des Systems wirksam.

5.5 Zusammenfassung von Regeln und Konventionen

Die verschiedenen Regeln und Konventionen bezogen auf den Aufbau der Tabellenmodule sind nachfolgend zusammengefasst:

- In jeder Code-Familie gibt es nur einen einzigen Referenzcode. Als Referenzcode müssen Sie den EBCDIC.DF.04-n-Code der Code-Gruppe wählen, wobei -n die (pseudo) ISO-Code-Variantennummer (hexadezimal) ist.
- Der Referenzcode muss alle Zeichen enthalten, die in den anderen Codes dieser Familie definiert sind. Andernfalls erfolgt keine Umwandlung.
- Bei der Umwandlungstabelle des Referenzcodes muss die Position des Zeichens gleich dem Zeichenwert sein (d.h., Byte Nr.i hat den Zeichenwert i).
- Zeichen, die in einem 'Nicht-Referenzcode' undefiniert sind, werden in der Umwandlungstabelle auf NIL (X'00') abgebildet. Wenn der Referenzcode undefinierte Zeichen enthält (z.B. Code-Gruppe mit der ISO-Code-Variantennummer 7), muss jedes undefinierte Zeichen im 'Nicht-Referenzcode' auf einem undefinierten Zeichen im Referenzcode abgebildet werden.
- Einige Zeichen können nicht redefiniert werden, da sie unabhängig vom verwendeten erweiterten Zeichensatz erkannt und verarbeitet werden müssen. Dies sind die Zeichen des EBCDIC-Kerns und die Zeichen # @ $! " und die Kontrollzeichen.
- In der Umwandlungstabelle von Klein- in Großbuchstaben werden Zeichen, die nicht in Großbuchstaben umgewandelt werden, auf sich selbst abgebildet.
- Die Sortiertabelle muss eindeutig sein, das heißt, jeder Wert innerhalb des Bereichs von X'00' bis X'FF' tritt genau nur einmal in dieser Tabelle auf.
- Für einige Softwareprodukte sind vollständige Codes immer 8-bit-Codes und begrenzte Codes immer 7-bit-Codes. Auch wenn das aus der Sicht von XHCS nicht immer stimmt, sollten Sie sich bei der Definition neuer Codes danach richten.
- Normierte Codes haben Standardnamen und sollten nicht redefiniert werden.
- Wenn Sie Tabellen modifizieren, beachten Sie, dass es bei einer Dateiübertragung mit anderen BS2000-Rechnern zu Fehlern kommt, wenn diese nicht die gleichen Tabellen benutzen. Ebenso müssen Sie die Regeln und Konventionen, die in diesem Abschnitt beschrieben sind, beachten. So dürfen Sie z.B. keine Zeichen ändern, die für das System von Bedeutung sind (z.B. Zeichen der Kommandosyntax). Um Fehler dieser Art zu vermeiden, sollten Sie die standardmäßig gelieferten erweiterten Zeichensätze nutzen und die Tabellenmodifikation als eine zusätzliche Eigenschaft von XHCS betrachten, die nur in speziellen Fällen (z.B. aus Kompatibilitätsgründen) genutzt wird.

5.6 Zusätzliche Unicode-Zeichen definieren

Mit XHCS V2.0 haben Sie die Möglichkeit, zusätzliche Unicode-Zeichen zu definieren. Wichtig ist dies, da von XHCS zunächst nur ein begrenzter Code-Umfang ausgeliefert wird. Es handelt sich dabei um die ISO8859-Varianten 1, 2, 3, 4, 5, 7, 9 und 15, sowie um Zeichen, die zusätzlich von unseren Kunden verwendet werden. Wenn Sie darüber hinaus weitere Zeichen für Unicode unterstützen wollen, gehen Sie vor wie nachfolgend beschrieben:

Prüfen Sie vorab, ob in der Erweiterung Zeichen sind, die in einer der schon existierenden, aber noch nicht unterstützten ISO-8859-Varianten vorhanden sind. Wenn dies der Fall ist, kann diese Variante mit einem eigenen Makro im Modul GNLMTAB unterstützt werden, wie in Abschnitt „Erstellung und Modifikation von Code-Tabellen“ auf Seite 120 beschrieben. Beachten Sie dabei, dass die 8-bit-Sortiertabelle keine Bedeutung für Unicode hat. Die entsprechende Sortierinformation für Unicode muss separat festgelegt werden, siehe Seite 133.

5.6.1 Pseudo-8-bit-Code definieren

Wenn Sie Zeichen hinzufügen wollen, die in keiner ISO8859-Variante vorkommen, müssen Sie zunächst einen Pseudo-8-bit-Code definieren.

- Verwenden Sie dazu einen der ausgelieferten Makros ISOEXT2 oder ISOEXT3. ISOEXT1 wurde schon zur Definition zusätzlicher von unseren Kunden benötigten Zeichen verwendet.
- Sortieren Sie zu Beginn die gewünschten Unicode-Zeichen binär.
- Ordnen Sie jedem Zeichen eine 8-bit-Position zu, beginnend mit 0xA0. Die Positionen 0x00 bis 0x9F sind bei allen ISO8859-Varianten gleich, und können daher nicht verwendet werden.
- Übernehmen Sie diese Zuordnung in die Tabelle 8-bit-Code -> Unicode-Position, welches die letzte Tabelle des ISOEXT2(3)-Makros ist.
- Legen Sie mithilfe dieser Tabelle die Klein-/Großschreibungstabelle und die Eigenschaftstabelle fest. Die Sortiertabelle hat auf Unicode-Ebene keine Bedeutung (siehe Seite 133).
- Versorgen Sie die Umwandlungstabelle und deren inverse Tabelle an den Positionen, die besetzt wurden, mit der identischen Abbildung. Alles andere ist 0x00.
- Verankern Sie den Pseudocode im Modul GNLMTAB:

  NLSCCS (ISOEXT2|3 , 0 , *NORMAL , *FULL ,*ISO , ISOEXT2|3,)
- Erhöhen Sie den Counter von NLSHEAD.

Der so definierte Pseudocode wird nur als Hilfsmittel zur Erweiterung des unterstützten Zeichenvorrats auf Unicode-Ebene benutzt. Er hat auf 8-bit-Ebene keine Bedeutung. Die Klein-/Großschreibungstabelle, deren Umkehrung sowie die Eigenschaftstabelle für Unicode werden dynamisch aus den vorhandenen 8-bit-Codes berechnet.

5.6.2 Sortierinformation festlegen

Die Sortierinformation für Unicode ist in den Modul GNLMTAB eingelagert, unmittelbar hinter den Makros zur Definition der 8-bit-Code-Tabellen. Sie ist so aufgebaut, wie unter NLSCOD TABLE=SORT auf Seite 69 beschrieben und besteht zunächst aus X'3000' = 12288 Einträgen (Collation Entries) je 8 Bytes, für mögliche Unicode-Positionen zwischen 0 und 2FFF. Die Nummer eines Eintrags entspricht seiner Unicode-Position. Derzeit sind ca. 700 Positionen belegt. Die Einträge sind belegt mit dem ersten Collation Element eines Collation Entries und evtl. einem Verweis auf ein zweites Collation Element. Hinter diesen X'3000'*8 = X'18000' Bytes beginnt der Bereich der zweiten und dritten Collation Elemente. Dieser hat die Größe von X'2000' Bytes und somit Platz für 1024 (= X'2000' / 8 = X'400') Collation-Elemente. Detaillierte Information dazu finden Sie unter NLSCOD, TABLE=SORT auf Seite 69.

5.6.3 GNLMTAB assemblieren und erweiterte Code-Basis einbinden

- Nachdem Sie die Sortierinformation festgelegt haben, assemblieren Sie den Modul GNLMTAB.
- Binden Sie die erweiterte Code-Basis beim Start des Subsystems XHCS-SYS oder dynamisch mit dem Kommando /ADD-CODE-TABLES (siehe auch Seite 134).

5.7 Neue Code-Tabellen dynamisch hinzufügen: ADD-CODE-TABLES

Das Kommando ADD-CODE-TABLES erlaubt es, Code-Tabellen dynamisch hinzuzufügen, ohne das System neu zu laden.

ADD-CODE-TABLES
FROM-LIBRARY = filename_1..54_without-generation

Beschreibung der Operanden

FROM-LIBRARY = filename_1..54_without-generation
XHCS-SYS verwendet diese Datei, um dynamisch neue Varianten hinzuzufügen. Dazu muss das Modul GNLMTAB (Typ R oder Typ L) in der angegebenen Bibliothek vorhanden sein.

Programmierhinweise

XHCS-SYS fügt automatisch neue Varianten hinzu und gibt diese aus, um den Systemadministrator zu informieren. Diese neuen Varianten stehen dann allen laufenden Tasks und Anwendungen zur Verfügung.

Falls die im hinzugefügten Modul definierten Varianten schon bekannt sind, (seit dem Start oder durch ein vorangegangenes /ADD-CODE-TABLES Kommando), so werden diese nicht in die Liste aufgenommen, um deren Konsistenz zu erhalten. D.h. die vorhandenen Definitionen werden beibehalten.

Das Kommando kann nicht dazu verwendet werden, bereits vorhandene Tabellen zu verändern.

Rechte, die zum Ausführen des Kommandos nötig sind

Die Rechte des XHCS-SYS Kommandos sind in der folgenden Tabelle aufgeführt:

Kommando	Rechte
ADD-CODE-TABLES	TSOS, OPERATING

Kurzbeschreibung der verschiedenen Kommando-Returncodes

	Returncode	Bedeutung	Liste der möglichen XHCS-SYS Meldungen
1	00 00 CMD0001	Kommando akzeptiert	Informationsmeldungen: GNL0006 GNL0007
2	02 00 GNL0100	Warnungen während der Kommando-Ausführung	Informationsmeldungen: GNL0003
3	00 40 CMD0216	Der Benutzer ist nicht berechtigt das Kommando auszuführen. Kommando abgelehnt	Fehlermeldungen: GNL0011
4	00 20 GNL0101	Fehler während eines externen Systemaufrufs. Kommando nicht ausgeführt	Fehlermeldungen: GNL0001 GNL0004
5	00 40 GNL0102	Nicht behebbarer Fehler bei der Ausführung. Kommando nicht ausgeführt	Fehlermeldungen: GNL0008
6	00 80 GNL0005	Kommando zur Zeit nicht möglich. Später erneut versuchen	Retry-Meldungen GNL0005

Mögliche Fehler und Warnungen

Fehler

– Da das Modul durch einen Aufruf der BLS Schnittstelle ($PBBND1) eingebunden wird, werden alle eventuellen Fehler, die beim Binden auftreten, vom BLS selbst ausgegeben.

– Nach dem Einbinden des Moduls GNLMTAB kann es vorkommen, dass XHCS ein ungültiges Format oder eine ungültige Version feststellt. In diesem Fall wird eine Ausführung des Kommandos verweigert.

Warnungen

– Das Modul GNLMTAB, das aus der angegebenen Bibliothek eingebunden wurde, enthält keine neuen Varianten.

5.8 Arabische und persische Codes

BS2000/OSD-BC unterstützt mit VTSU ab V11.0 und speziellen Routinen bidirektionale Schriften.

5.8.1 Unterstützte Codes

XHCS unterstützt die folgenden Codes für bidirektionale Schriften:

ISO-Code-Variante	XHCS-Name (CCSN)	vollständiger Name	Darstellung
6	EDF046 EEHCLAA	EBCDIC.DF.04-6.ARA EBCDIC.EHC.LA.ARA	Latein/Arabisch mit arabischen Ziffern
A	EDF04A EEHCLAI	EBCDIC.DF.04-6.IND EBCDIC.EHC.LA.IND	Latein/Arabisch mit indischen Ziffern
C	EDF04C EEHCLFI	EBCDIC.DF.04-FAR.INT EBCDIC.EHC.LF.INT	Latein/Farsi mit internationalen Ziffern
D	EDF04D EEHCLFF	EBCDIC.DF.04-FAR.FAR EBCDIC.EHC.LF.FAR	Latein/Farsi mit persischen Ziffern
E	EDF04E EEHCNAA	EBCDIC.DF.04-NAF.ARA EBCDIC.EHC.NA.ARA	Franz./Arabisch mit arabischen Ziffern
F	EDF04F EEHCNAI	EBCDIC.DF.04-NAF.IND EBCDIC.EHC.NA.IND	Franz./Arabisch mit indischen Ziffern

Im Gegensatz zu anderen Zeichensätzen ist der Terminal-Code (line code) für bidirektionale Schriften nicht voll kompatibel mit dem verwendeten Rechner-Code. Einige spezielle Zeichen erscheinen nur einmal im Terminal-Code, ob sie aus der lateinischen oder der arabischen Schrift kommen, wird nicht berücksichtigt.

Auf der Rechnerseite (EBCDIC-Code) kommen diese speziellen Zeichen doppelt vor, so gibt es z.B. zwei „+“-Zeichen, eines ist der lateinischen, das andere der arabischen Schrift zugeordnet. Die lateinische oder arabische Zugehörigkeit kann so eindeutig festgestellt werden.

Bedingt durch den Unterschied in der Struktur des Terminal- und Rechner-Codes müssen diese Codes zusätzlich auf der Basis von VTSU ab V11.0 bearbeitet werden.

Nur der CCS für Latein/Arabisch ist in ISO 8859 (Variante 6) definiert.

Im Gegensatz zu anderen Zeichensätzen sind die Zeichensätze, die bidirektionale Schriften unterstützen, durch sechs „Pseudo-ISO-Variantennummern“ gekennzeichnet. Die Nummer, die diese Code-Familie kennzeichnet, ist keine vorhandene ISO-Variantennummer.

5.8.2 Regeln für arabische Codes

Zwei Terminal-Codes, die das gleiche Alphabet abdecken, sind aus Sicht von XHCS gleichwertig, wenn sie sich nur in der Darstellung der Ziffern unterscheiden. Daraus ergibt sich, dass der Inhalt der Code-Familien 6 und A, C und D, E und F paarweise identisch sein sollte, wenn sie definiert werden. Dies kann dadurch erfolgen, dass für äquivalente Codes gleiche Default-Codes und Redefinitionsmakros erstellt werden („deftab-“ und „macroname-“Operanden der NLSCTAB- und NLSCCS-Makros). Damit ist ein identischer Inhalt der Tabellen sichergestellt. Dieses paarweise Zuordnen von Codes zeigt sich auch im Namen der arabischen Standard-Codes, die sich nur im letzten Buchstaben unterscheiden.

5.9 Unterstützung des EURO-Zeichens in XHCS

Zur Unterstützung des EURO-Zeichens € in XHCS gibt es zwei Wege. Sie können
– weiter die ISO-Code-Varianten 8859-1, 8859-2 und 8859-9 verwenden oder
– die neue ISO-Code-Variante 8859-15 einführen.

Beide Wege sind in diesem Abschnitt beschrieben.

– Die Einführung der neuen Variante ISO 8859-15 führt zu Inkompatibilitäten bei der Unterstützung arabischer Codes. Diese Einschränkung ist im Abschnitt „Kompatibilitätsbeschränkungen zur arabischen Code-Variante F“ auf Seite 139 beschrieben.

– In kyrillischen und ISO-7-bit-Codes wird das EURO-Zeichen nicht unterstützt.

5.9.1 Verwendung der ISO-Code-Varianten ISO 8859-1/-2/-7/-9

Zur Unterstützung des EURO-Zeichens € können die ISO-Code-Varianten 8859-1 (ISO88591), 8859-2 (ISO88592), 8859-7 (ISO88597) und 8859-9 (ISO88599) und die zugehörigen EBCDIC-Codes (siehe Tabelle auf Seite 146) weiter verwendet werden.

Emulationen können unabhängig vom eingestellten lateinischen 8-bit-Zeichensatz das Zeichen an Position X’A4’ (ASCII) bzw. X’9F’ (EBCDIC) wie bisher als Währungszeichen ¤ oder neu als EURO-Zeichen interpretieren. Die Auswahl der Ausgabeform erfolgt über eine Option im Emulations-Menü oder durch eine Eintrag in der Konfigurationsdatei (.INI) der Emulation. Die gleichzeitige Ausgabe des Währungszeichens ¤ und des EURO-Zeichens ist nicht möglich.

Die Einstellung der Option („Währungszeichen“ oder „EURO-Zeichen“) gilt für alle drei Code-Varianten. Das heißt zum Beispiel, dass es nicht möglich ist, die Option für 8859-1 auf „EURO-Zeichen“ zu stellen und für die beiden anderen Varianten auf „Währungszeichen“.

Durch dieses Vorgehen werden für das EURO-Zeichen keine neuen, anders benannten Zeichensätze benötigt. Dateien dürfen das EURO-Zeichen enthalten, ohne dass ihnen ein neues Code-Attribut (Coded Character Set Name) zugewiesen wird. Bestehende Anwendungen brauchen nicht modifiziert zu werden. Die geänderte Interpretation der bisherigen Code-Tabellen ist zur bisherigen kompatibel. Zeichen-Eigenschaften, Konvertierungen und Sortierungen werden nicht berührt.

5.9.2 Einführung der ISO-Code-Variante ISO 8859-15

XHCS unterstützt die neue Norm-Variante ISO 8859-15 (Latin-9) durch eine neue Code-Gruppe mit den ISO-Tabellen und dem entsprechenden EBCDIC-Äquivalent. Das EURO-Zeichen € steht darin an Position X'A4' und ersetzt das Währungszeichen ¤, das in den anderen lateinischen ISO-Varianten diese Position belegt.

Außerdem sind in dieser Code-Variante gegenüber ISO 8859-1 spezielle Zeichen durch europäische Spezial-Buchstaben ersetzt worden. Dadurch unterstützt Latin-9 alle EURO-relevanten Sprachen. Allerdings haben sich Zeichen-Eigenschaften, die über Programm-Schnittstellen abfragbar sind, inkompatibel geändert.

Die Code-Gruppe ISO 8859-15 enthält die Tabellen der Codes

- EDF04F (EBCDIC, Referenzcode der Gruppe)
- ISO8859F (ASCII)
- WCP1252P (beschränkt kompatible Variante des Windows-Zeichensatzes 1252, der das EURO-Symbol enthält)

Durch Angabe dieser Code-Gruppe beim Dateiattribut CODED-CHARACTER-SET-NAME wird für Textdateien explizit festgelegt, dass alle enthaltenen X'9F'-EBCDIC-Codes als EURO-Zeichen interpretiert werden müssen.

5.9.2.1 Kompatibilitätsbeschränkungen zur arabischen Code-Variante F

XHCS unterstützt bereits eine Code-Variante mit der Bezeichnung 'F' für arabische Codes, die keine Standard-ISO-Variante ist (siehe Abschnitt „Arabische und persische Codes“ auf Seite 136). Wegen derselben Bezeichnung kommt die Standard-ISO-Variante 8859-15 mit dieser Variante in Konflikt. Die gleichzeitige Nutzung der arabischen Variante 'F' und der ISO-Variante 'F' mit dem EURO-Zeichen € ist ausgeschlossen

In der Standardinstallation ist die europäische Variante ISO 8859-15 aktiviert.

Die Unterstützung der arabischen Variante erfordert die Installation spezieller VTSU-Sonderroutinen (siehe Abschnitt „Unterstützung spezieller Datenstationen durch VTSU-Sonderroutinen“ auf Seite 36).

- Wenn diese Routinen installiert sind, betrachtet VTSU die Variante 'F' als arabische Variante.
- Wenn diese Routinen nicht installiert sind, nimmt VTSU die Variante 'F' als europäische Variante.

Der Systembetreuer ist für die konfliktfreie Installation verantwortlich.

6 Einsatzvorbereitung

Dieses Kapitel beschreibt die Installationsschritte bis zur einsatzfähigen 8-bit-Datenstation bzw. nationalen 7-bit-Datenstation.

6.1 Einsatzvorbereitung für die XHCS-Unterstützung

6.1.1 PDN-Generierung

Terminals vom Typ 9758, 9759 und 9763 müssen mit STATTYP= DSS-9763 und Terminals vom Typ 9756 müssen mit STATTYP= DSS-9755 im PDN generiert werden. 8-bit-Drucker müssen im PDN mit ihren entsprechenden Stationstypen definiert werden. Zusätzlich müssen die Drucker über PDN-Freitextparameter (PDN ab V11) oder in der VTSU-Betriebsparameterdatei als 8-bit-Drucker definiert werden. VTSU geht davon aus, dass der richtige Zeichensatz für den Drucker eingestellt ist.

ACHTUNG!

Der PDN-Freitextparameter hat eine höhere Priorität als die entsprechenden VTSU-Betriebsparameter.

6.1.2 XHCS

Das optionale Produkt XHCS ist ein dynamisches Subsystem, das über DSSM geladen wird und im laufenden Betrieb nicht entladen werden kann. XHCS unterstützt standardmäßig die ISO-Code-Varianten 1, 2, 3, 4, 5, 7, 9 und F. Die entsprechenden Code-Tabellen sind im Modul GNLMTAB definiert (siehe Abschnitt „Komponenten und Programmschnittstellen“ auf Seite 28).

Zusätzlich unterstützt XHCS für alle anderen Code-Varianten entsprechende Standard-EBCDI-Codes (siehe Seite 145). Diese Codes sind im Modul GNLMTAB vordefiniert, aber durch Kommentarzeichen deaktiviert. Durch Entfernen der entsprechenden Kommentarzeichen können die jeweils benötigten Code-Tabellen aktiviert werden. Das Modul GNLMTAB muss assembliert werden und das übersetzte Modul muss in die Modulbibliothek SYSLNK.XHCS-SYS.020 aufgenommen werden. Das modifizierte Modul GNLMTAB kann mit dem Kommando /ADD-CODE-TABLES ins System eingebunden werden (siehe Abschnitt „Neue Code-Tabellen dynamisch hinzufügen: ADD-CODE-TABLES“ auf Seite 134). Ansonsten wird die Modifikation erst nach dem nächsten Laden des Subsystems XHCS-SYS wirksam, d.h. nach dem nächsten Systemstart.

Zusätzliche Tabellen können durch Modifikation der im Modul GNLMTAB enthaltenen Tabellen definiert werden. Damit können Sie benutzereigene EBCDIC-Varianten definieren.

6.1.3 Aktivierung der 8-bit-Umgebung für 8-bit-Terminals

Der 8-bit-Modus kann entweder fest eingestellt (permanenter 8-Bit-Modus) oder nur für ein bestimmtes Programm aktiviert werden.

Permanenter 8-bit-Modus

Der permanente 8-bit-Modus kann für TIAM-, DCAM- und openUTM-Anwendungen durch Verwendung der VTSU-Betriebsparameterdatei systemweit eingestellt werden, indem der Name des Standard-Anwenderzeichensatzes (CCSN) automatisch für die 8-bit-DSS aktiviert wird. Zusätzlich kann für TIAM-Anwendungen dieser CCSN mithilfe des TIAM-Kommandos

```
/MODIFY-TERMINAL-OPTIONS CODED-CHARACTER-SET=8-BIT-DEFAULT
```

vom Anwender selbst aktiviert werden.

Programmbezogener 8-bit-Modus

Der 8-bit-Modus kann von Programmen (TIAM- und DCAM-Anwendungen) über den VTSUCB eingestellt werden.

Der 8-bit-Modus wird auch automatisch für bestimmte Dienstprogramme (z.B. EDT, IFG/FHS...) aktiviert, falls z.B. mit Dateien bzw. Formaten gearbeitet wird, die das Dateikennzeichen CCSN besitzen, welches von der Datenstation unterstützt wird.

Beispiel

Die Datenstation 9759-M2 ist in der SIDATA-Keyboard-Einstellung auf griechischen Zeichensatz eingestellt. Der 8-bit-Modus ist nicht aktiviert worden. Der 8-bit-Modus wird z.B. automatisch für den Editor EDT aktiviert, indem der EDT eine Datei mit griechischem CCSN einliest.

6.2 Einsatzvorbereitung für VTSU-Sonderroutinen

6.2.1 PDN-Generierung

Terminals vom Typ 9758 und 9763 müssen mit STATTYP= DSS-9763, Terminals vom Typ 9756 müssen mit STATTYP=DSS-9755 und nationale 7-bit-Terminals müssen mit STATTYP = DSS-9750 im PDN generiert werden. ESC-Drucker müssen im PDN mit ihren entsprechenden Stationstypen definiert werden. Zusätzlich müssen die Drucker über PDN-Freitextparameter (PDN ab V11) oder in der VTSU-Betriebsparameterdatei (siehe Handbuch „VTSU“ [1]) als 8-bit-Drucker definiert werden. VTSU geht davon aus, dass der richtige Zeichensatz für den Drucker eingestellt ist.

ACHTUNG!

Der PDN-Freitextparameter hat eine höhere Priorität als die entsprechenden VTSU-Betriebsparameter.

6.2.2 VTSU

Nach Änderung der Konfigurationsdatei und/oder nach Installation der „VTSU-Sonderroutinen“ ist das Entladen des Subsystems VTSU mit anschließendem Systemstart erforderlich.

6.2.3 Aktivierung der 8-bit-Umgebung für arabische/persische Datenstationen

Die arabischen/persischen 8-bit-Terminals können Sie nur nutzen, wenn Sie die Unterstützung dieser Terminals in der Installationsprozedur angefordert haben.
Die Installationsprozedur ist im Handbuch „VTSU“ [1] beschrieben. Die Aktivierung des 8-bit-Modus ist auf Seite 142 beschrieben. Eine Liste der gültigen CCSN für ARA/FAR ist auf Seite 146 aufgeführt.

6.2.4 Aktivierung der europäischen 7-bit-Terminals

Die europäischen 7-bit-Terminals können Sie nur nutzen, wenn Sie die Unterstützung dieser Terminals in der Installationsprozedur angefordert haben. Die Installationsprozedur ist im Handbuch „VTSU“ [1] beschrieben. Europäische 7-bit-Terminals müssen zusätzlich in einer so genannten Konfigurationsdatei definiert werden. Das Erstellen der Konfigurationsdatei ist ebenfalls im Handbuch „VTSU“ [1] beschrieben.

6.2.5 Aktivierung der ESC-Drucker

ESC-Drucker müssen in einer sog. Konfigurationsdatei definiert werden. Das Erstellen der Konfigurationsdatei obliegt dem Anwender und ist im Handbuch „VTSU“ [1] beschrieben. Zusätzlich müssen die Drucker über PDN-Freitextparameter (PDN ab V11) oder in der VTSU-Betriebsparameterdatei (siehe Handbuch „VTSU“ [1]) als 8-bit-Drucker definiert werden. VTSU geht davon aus, dass der richtige Zeichensatz für den Drucker eingestellt ist.

7 Tabellen

7.1 Tabellen der Standard-CCSN

7-bit-Leitungscodes (ISO-Codes) und zugehörige EBCDI-Codes

ISO-Code-Variante	XHCS-Name	Vollständiger Name	Darstellung
1	ISO646 EDF03IRV EDF03DRV	ISO646-IRV EBCDIC.DF.03-IRV *) EBCDIC.DF.03-DRV	Internationaler 7 bit Code

8-bit-Leitungscodes (ISO-Codes) und zugehörige EBCDI-Codes

ISO-Code-Variante	XHCS-Name (CCSN)	Vollständiger Name	Darstellung
1	ISO88591 EDF041 EDF04DRV	ISO8859-1 EBCDIC.DF.04(-1) *) EBCDIC.DF.04(-1) DRV	Lateinisches Alphabet Nr. 1 Erweiterung des EDF03DRV
2	ISO88592 EDF042 EEHCL2	ISO8859-2 EBCDIC.DF.04-2 EBCDIC.EHC.L2 *)	Lateinisches Alphabet Nr. 2
3	ISO88593 EDF043	ISO8859-3 EBCDIC.DF.04-3	Lateinisches Alphabet Nr. 3
4	ISO88594 EDF044	ISO8859-4 EBCDIC.DF.04-4	Lateinisches Alphabet Nr. 4
5	ISO88595 EDF045 EEHCLC EEHCLC1	ISO8859-5 EBCDIC.DF.04-5 EBCDIC.EHC.LC *) EBCDIC.EHC-LC.1 **)	Lateinisch/kyrillisches Alphabet
6 A B C D E F	EDF046 EEHCLAA EDF04A EEHCLAI EDF04B EDF04C EEHCLFI EDF04D EEHCLFF EDF04E EEHCNAA EDF04F EEHCNAI	EBCDIC.DF.04-6.ARA EBCDIC.EHC.LA.ARA*) EBCDIC.DF.04-6.IND EBCDIC.EHC.LA.IND*) EBCDIC.DF.04.BIB.9756 EBCDIC.DF.04-FAR.INT EBCDIC.EHC.LF.INT*) EBCDIC.DF.04-FAR.FAR EBCDIC.EHC.LF.FAR EBCDIC.DF.04-NAF.ARA EBCDIC.EHC.NA.ARA*) EBCDIC.DF.04-NAF.IND EBCDIC.EHC.NA.IND*)	Lateinisch/arabisches Alphabet mit arabischen Ziffern Lateinisch/arabisches Alphabet mit indischen Ziffern Zeichensatz für 9756-Bibliotheks- terminal nach DIN 66003/ DIN 31624 Lateinisch/Farsi Alphabet mit internationalen Ziffern Lateinisch/Farsi Alphabet mit persischen Ziffern Franz./arabisches Alphabet mit arabischen Ziffern Franz./arabisches Alphabet mit indischen Ziffern

ISO-Code-Variante	XHCS-Name (CCSN)	Vollständiger Name	Darstellung
7	ISO88597 EDF047 EEHCLG	ISO8859-7(ELOT928) EBCDIC.DF.04-7 EBCDIC.EHC.LG *)	Lateinisch/griechisches Alphabet
9	ISO88599 EDF049 oder EDF04BE	ISO8859-9 EBCDIC.DF.04-9 *) EBCDIC.DF.04.BIB	Lateinisches Alphabet Nr. 5 Bibliothekszeichensatz nach DIN 31628/2 (Emulation)
F	ISO8859F EDF04F WCP1252P	ISO8859-15 EBCDIC.DF.04-F Windows Code Page 1252 Partial	Lateinisches Alphabet Nr. 9

*) empfohlener Standard-EBCDIC

**) empfohlener Standard-EBCDIC für Neukunden

Die Namen der unterschiedlichen ISO 8859- und EBCDIC.DF.04-Codes werden immer in Form von ’ISO 8859x’ und ’EDF04x’ dargestellt. Diese Regel wird für alle ISO-Code-Variantennummern angewendet.

7.2 Überblick der XHCS-VTSUCB-Rückmeldungen

In der nachfolgenden Tabelle, finden Sie einen Überblick über die unterschiedlichen VTSUCB-Rückmeldungen die XHCS betreffen. Welche Rückmeldungen geliefert werden, hängt davon ab, ob XHCS geladen ist, welchen Anwender-/System-Standard-Code oder welchen Datenstationstyp Sie benutzen und welche Parameter im VTSUCB festgelegt wurden.

XHCS	8-bit-Code*	8-bit-Datenst.	VTSUCB-Parameter CODETR	CCSNAME	Returncode x	Bemerkung
N	—	—	Y	—	00000000	5)
N	—	—	—	Y	60010004	
Y	—	N	Y	—	00000000	5)
Y	—	N	—	Y	61010004	
Y	N	Y	Y	—	00000000	5)
Y	N	Y	—	Y	61010004	
Y	Y	Y	Y	—	00000000	1)
Y	Y	Y	—	*EXTEND	00000000	3)
Y	Y	Y	—	EDF03IRV	00000000	2)
Y	Y	Y	—	andere	00000000	4)
Y	Y	Y	—	andere	1E010004	4)
Y	Y	Y	—	andere	86010004	4)

— nicht relevant

Y vorhanden

N nicht vorhanden

* N bedeutet entweder, dass der Anwender-/System-Standard-Code kein vollständiger 8-bit-Code ist, oder, dass dieser Code nicht durch die aktuellen Datenstationen unterstützt wird.

x Returncodes ohne Bemerkung sind im Handbuch „VTSU“ [1] beschrieben

1) Der Parameter CODETR wird solange ignoriert, bis eine Nachricht im 8-bit-Modus gesendet wird.

2) Die Nachricht wird im 7-bit-Modus gesendet. Die Nachricht wird dabei so verarbeitet, als ob kein VTSUCB benutzt wird, bzw. kein Code-Name festgelegt wurde.

3) Die gesendete Nachricht benutzt den Standard-Anwender-Code.

4) Die gesendete Nachricht benutzt den festgelegten Code. Dieser Code muss in XHCS bekannt sein und von den verwendeten Datenstationen unterstützt werden. Ansonsten wird die Meldung mit dem Returncode X’1E’ (Name unbekannt in XHCS) oder X’86’ (Name wird von der Datenstation nicht unterstützt) abgelehnt.

5) Der Parameter CODETR wird ignoriert. Umgebungsfehler werden nur über den Parameter CCSNAME entdeckt.

7.3 Unterstützte Leitungscodes und BS2000-EBCDI-Codes

7.3.1 Unterstützte Leitungscodes und BS2000-EBCDI-Codes für europäische Schriften

Erläuterung der Raster:

Steuerzeichen nicht belegbar

Position nicht belegbar

EBCDIC.DF03-Kernel

7.3.1.1 8-bit-Leitungscodes und zugehörige EBCDI-Codes für europäische Schriften

Latin Alphabet No. 1 **ISO 8859-1** (CCSN: **ISO88591**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	À	Đ	à	ð
01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
06			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
08			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
0A			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
0B			+	;	K	[	k	{			«	»	Ë	Û	ë	û
0C			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
0D			-	=	M	]	m	}			SHY	½	Í	Ý	í	ý
0E			.	>	N	^	n	~			®	¾	Î	Þ	î	þ
0F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Erweiterter BS2000-Code **EBCDIC.DF.04-1** (CCSN: **EDF041**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ø	Ø	°	µ	¢	ù	¦	Ù	0
01					NBSP	é	/	É	a	j	¯	£	A	J	÷	1
02					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
04					à	è	À	È	d	m	u	©	D	M	U	4
05					á	í	Á	Í	e	n	v	§	E	N	V	5
06					ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07					å	ï	Å	Ï	g	p	x	¼	G	P	X	7
08					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
09					ñ	ß	Ñ	¨	i	r	z	¾	I	R	Z	9
0A					`	!	^	:	«	ª	¡	¬	SHY	¹	²	³
0B					.	$	,	#	»	º	¿	[	ô	û	Ô	{
0C					<	*	%	@	ð	æ	Ð	\	ö	ü	Ö	Ü
0D					(	)	_	'	ý	¸	Ý	]	ò	Û	Ò	}
0E					+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
0F					|		?	"	±	¤	®	×	õ	ÿ	Õ	~

Erweiterter BS2000-Code **EBCDIC.DF.04-DRV** (CCSN: **EDF04DRV**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ø	Ø	°	µ	¢	{	}	\	0
01					NBSP	é	/	É	a	j	~	£	A	J	÷	1
02					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03					|	ë	¦	Ë	c	l	t	·	C	L	T	3
04					à	è	À	È	d	m	u	©	D	M	U	4
05					á	í	Á	Í	e	n	v	@	E	N	V	5
06					ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07					å	ï	Å	Ï	g	p	x	¼	G	P	X	7
08					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
09					ñ		Ñ	¨	i	r	z	¾	I	R	Z	9
0A					`	!	^	:	«	ª	¡	¬	SHY	¹	²	³
0B					.	$	,	#	»	º	¿	Ä	ô	û	Ô	ä
0C					<	*	%	§	ð	æ	Ð	Ö	[	]	Û	Ù
0D					(	)	_	'	ý	¸	Ý	Ü	ò	ù	Ò	ü
0E					+	;	>	=	þ	Æ	Þ	´	ó	ú	Ó	Ú
0F					ö	¯	?	"	±	¤	®	×	õ	ÿ	Õ	ß

Latin Alphabet No. 9 **ISO 8859-15** (CCSN: **ISO885915**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	À	Đ	à	ð
01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
04			$	4	D	T	d	t			€	Ž	Ä	Ô	ä	ô
05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
06			&	6	F	V	f	v			Š	¶	Æ	Ö	æ	ö
07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
08			(	8	H	X	h	x			š	ž	È	Ø	è	ø
09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
0A			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
0B			+	;	K	[	k	{			«	»	Ë	Û	ë	û
0C			,	<	L	\	l	|			¬	Œ	Ì	Ü	ì	ü
0D			-	=	M	]	m	}			SHY	œ	Í	Ý	í	ý
0E			.	>	N	^	n	~			®	Ÿ	Î	Þ	î	þ
0F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Erweiterter BS2000-Code **EBCDIC.DF.04-F** (CCSN: **EDF04F**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ø	Ø	°	µ	¢	ù	Š	Ù	0
01					NBSP	é	/	É	a	j	¯	£	A	J	÷	1
02					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
04					à	è	À	È	d	m	u	©	D	M	U	4
05					á	í	Á	Í	e	n	v	§	E	N	V	5
06					ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07					å	ï	Å	Ï	g	p	x	Œ	G	P	X	7
08					ç	ì	Ç	Ì	h	q	y	œ	H	Q	Y	8
09					ñ	ß	Ñ	š	i	r	z	Ÿ	I	R	Z	9
0A					`	!	^	:	«	ª	¡	¬	SHY	¹	²	³
0B					.	$	,	#	»	º	¿	[	ô	û	Ô	{
0C					<	*	%	@	ð	æ	Ð	\	ö	ü	Ö	Ü
0D					(	)	_	'	ý	ž	Ý	]	ò	Û	Ò	}
0E					+	;	>	=	þ	Æ	Þ	Ž	ó	ú	Ó	Ú
0F					|		?	"	±	€	®	×	õ	ÿ	Õ	~

Windows Code Page 1252 Partial (CCSN: **WCP1252P**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p	€		NBSP	°	À	Đ	à	ð
01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
04			$	4	D	T	d	t					Ä	Ô	ä	ô
05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
06			&	6	F	V	f	v				¶	Æ	Ö	æ	ö
07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
08			(	8	H	X	h	x					È	Ø	è	ø
09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
0A			*	:	J	Z	j	z	Š	š	ª	º	Ê	Ú	ê	ú
0B			+	;	K	[	k	{			«	»	Ë	Û	ë	û
0C			,	<	L	\	l	|	Œ	œ	¬		Ì	Ü	ì	ü
0D			-	=	M	]	m	}			SHY		Í	Ý	í	ý
0E			.	>	N	^	n	~			®		Î	Þ	î	þ
0F			/	?	O	_	o	DEL		Ÿ	¯	¿	Ï	ß	ï	ÿ

Latin Alphabet No. 5 **ISO 8859-9** (CCSN: **ISO88599**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	À	Ğ	à	ğ
01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ
02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò
03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ
06			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö
07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
08			(	8	H	X	h	x			¨	¸	È	Ø	è	ø
09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù
0A			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú
0B			+	;	K	[	k	{			«	»	Ë	Û	ë	û
0C			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü
0D			-	=	M	]	m	}			SHY	½	Í	İ	í	ı
0E			.	>	N	^	n	~			®	¾	Î	Ş	î	ş
0F			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ

Erweiterter BS2000-Code **EBCDIC.DF.04-9** (CCSN: **EDF049**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ø	Ø	°	µ	¢	ù	¦	Ù	0
01					NBSP	é	/	É	a	j	¯	£	A	J	÷	1
02					â	ê	Â	Ê	b	k	s	¥	B	K	S	2
03					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
04					à	è	À	È	d	m	u	©	D	M	U	4
05					á	í	Á	Í	e	n	v	§	E	N	V	5
06					ã	î	Ã	Î	f	o	w	¶	F	O	W	6
07					å	ï	Å	Ï	g	p	x	¼	G	P	X	7
08					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
09					ñ	ß	Ñ	¨	i	r	z	¾	I	R	Z	9
0A					`	!	^	:	«	ª	¡	¬	SHY	¹	²	³
0B					.	$	,	#	»	º	¿	[	ô	û	Ô	{
0C					<	*	%	@	ğ	æ	Ğ	\	ö	ü	Ö	Ü
0D					(	)	_	'	ı	¸	İ	]	ò	Û	Ò	}
0E					+	;	>	=	ş	Æ	Ş	´	ó	ú	Ó	Ú
0F					|		?	"	±	¤	®	×	õ	ÿ	Õ	~

Latin Alphabet No. 2 **ISO 8859-2** (CCSN: **ISO88592**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	Ŕ	Đ	ŕ	đ
01			!	1	A	Q	a	q			Ą	ą	Á	Ń	á	ń
02			"	2	B	R	b	r			˘	˛	Â	Ň	â	ň
03			#	3	C	S	c	s			Ł	ł	Ă	Ó	ă	ó
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u			Ľ	ľ	Ĺ	Ő	Í	ő
06			&	6	F	V	f	v			Ś	ś	Ć	Ö	ć	ö
07			'	7	G	W	g	w			§	ˇ	Ç	×	ç	÷
08			(	8	H	X	h	x			¨	¸	Č	Ř	č	ř
09			)	9	I	Y	i	y			Š	š	É	Ů	é	ů
0A			*	:	J	Z	j	z			Ş	ş	Ę	Ú	ę	ú
0B			+	;	K	[	k	{			Ť	ť	Ë	Ű	ë	ű
0C			,	<	L	\	l	|			Ź	ź	Ě	Ü	ě	ü
0D			-	=	M	]	m	}			SHY	˝	Í	Ý	í	ý
0E			.	>	N	^	n	~			Ž	ž	Î	Ţ	î	ţ
0F			/	?	O	_	o				Ż	ż	Ď	ß	ď	˙

Erweiterter BS2000-Code **EBCDIC.DF.04-2** (CCSN: **EDF042**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ř	Ř	°	ľ	˘	ů	Ś	Ů	0
01					NBSP	é	/	É	a	j	Ż	Ł	A	J	÷	1
02					â	ę	Â	Ę	b	k	s	Ľ	B	K	S	2
03					ä	ë	Ä	Ë	c	l	t	ˇ	C	L	T	3
04					ŕ	č	Ŕ	Č	d	m	u	Š	D	M	U	4
05					á	í	Á	Í	e	n	v	§	E	N	V	5
06					ă	î	Ă	Î	f	o	w	ś	F	O	W	6
07					Í	ď	Ĺ	Ď	g	p	x	ź	G	P	X	7
08					ç	ě	Ç	Ě	h	q	y	˝	H	Q	Y	8
09					ń	ß	Ń	¨	i	r	z	ž	I	R	Z	9
0A					`	!	^	:	Ť	Ş	Ą	Ź	SHY	š	˛	ł
0B					.	$	,	#	ť	ş	ż	[	ô	ű	Ô	{
0C					<	*	%	@	đ	ć	Đ	\	ö	ü	Ö	Ü
0D					(	)	_	'	ý	¸	Ý	]	ň	Ű	Ň	}
0E					+	;	>	=	ţ	Ć	Ţ	´	ó	ú	Ó	Ú
0F					|		?	"	ą	¤	Ž	×	ő	˙	Ő	~

Erweiterter BS2000-Code **EBCDIC.EHC.L2** (CCSN: **EEHCL2**)

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00					SP	&	-	ř	Ř	°	ˇ	˘	ů	˝	Ů	0
01					NBSP	ę	/	á	a	j	ß	Á	A	J	÷	1
02					â	Ę	Â	é	b	k	s	É	B	K	S	2
03					ä	ë	Ä	í	c	l	t	Í	C	L	T	3
04					ă	Ë	Ă	ó	d	m	u	Ó	D	M	U	4
05					ą	ě	Ą	ö	e	n	v	Ö	E	N	V	5
06					ć	î	Ć	ő	f	o	w	Ő	F	O	W	6
07					č	Ě	Č	ú	g	p	x	Ú	G	P	X	7
08					ç	Î	Ç	ü	h	q	y	Ü	H	Q	Y	8
09					×	§	¨	ű	i	r	z	Ű	I	R	Z	9
0A					`	!	^	:	ł	ľ	Ł	¸	SHY	Í	·	Ĺ
0B					.	$	,	#	ď	Ľ	Ď	[	ô	ş	Ô	{
0C					<	*	%	@	đ	ŕ	Đ	\	ń	ź	Ń	Ź
0D					(	)	_	'	ý	¸	Ý	]	ň	Ş	Ň	}
0E					+	;	>	=	ţ	Ŕ	Ţ	´	ś	ż	Ś	Ż
0F					|		?	"	ť	¤	Ť	ž	š	Ž	Š	~

Latin Alphabet No. 3 **ISO 8859-3** (CCSN: **ISO88593**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	À		à	
01			!	1	A	Q	a	q			Ħ	ħ	Á	Ñ	á	ñ
02			"	2	B	R	b	r			˘	²	Â	Ò	â	ò
03			#	3	C	S	c	s			£	³		Ó		ó
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u				µ	Ċ	Ġ	ċ	ġ
06			&	6	F	V	f	v			Ĥ	ĥ	Ĉ	Ö	ĉ	ö
07			'	7	G	W	g	w			§	·	Ç	×	ç	÷
08			(	8	H	X	h	x			¨	¸	È	Ĝ	è	ĝ
09			)	9	I	Y	i	y			İ	ı	É	Ù	é	ù
0A			*	:	J	Z	j	z			Ş	ş	Ê	Ú	ê	ú
0B			+	;	K	[	k	{			Ğ	ğ	Ë	Û	ë	û
0C			,	<	L	\	l	|			Ĵ	ĵ	Ì	Ü	ì	ü
0D			-	=	M	]	m	}			SHY	½	Í	Ŭ	í	ŭ
0E			.	>	N	^	n	~					Î	Ŝ	î	ŝ
0F			/	?	O	_	o				Ż	ż	Ï	ß	ï	˙

Erweiterter BS2000-Code **EBCDIC.DF.04-3** (CCSN: **EDF043**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ĝ	Ĝ	°	µ	˘	ù	Ĥ	Ù	0
01					NBSP	é	/	É	a	j	Ż	£	A	J	÷	1
02					â	ê	Â	Ê	b	k	s		B	K	S	2
03					ä	ë	Ä	Ë	c	l	t	·	C	L	T	3
04					à	è	À	È	d	m	u	İ	D	M	U	4
05					á	í	Á	Í	e	n	v	§	E	N	V	5
06						î		Î	f	o	w	ĥ	F	O	W	6
07					ċ	ï	Ċ	Ï	g	p	x	ĵ	G	P	X	7
08					ç	ì	Ç	Ì	h	q	y	½	H	Q	Y	8
09					ñ	ß	Ñ	¨	i	r	z		I	R	Z	9
0A					`	!	^	:	Ğ	Ş	Ħ	Ĵ	SHY	ı	²	³
0B					.	$	,	#	ğ	ş	ż	[	ô	û	Ô	{
0C					<	*	%	@		ĉ		\	ö	ü	Ö	Ü
0D					(	)	_	'	ŭ	¸	Ŭ	]	ò	Û	Ò	}
0E					+	;	>	=	ŝ	Ĉ	Ŝ	´	ó	ú	Ó	Ú
0F					|		?	"	ħ	¤		×	ġ	·	Ġ	~

Latin Alphabet No. 4 **ISO 8859-4** (CCSN: **ISO88594**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	Ā	Đ	ā	đ
01			!	1	A	Q	a	q			Ą	ą	Á	Ņ	á	ņ
02			"	2	B	R	b	r			ĸ	˛	Â	Ō	â	ō
03			#	3	C	S	c	s			Ŗ	ŗ	Ã	Ķ	ã	ķ
04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô
05			%	5	E	U	e	u			Ĩ	ĩ	Å	Õ	å	õ
06			&	6	F	V	f	v			Ļ	ļ	Æ	Ö	æ	ö
07			'	7	G	W	g	w			§	ˇ	Ļ	×	į	÷
08			(	8	H	X	h	x			¨	¸	Č	Ø	č	ø
09			)	9	I	Y	i	y			Š	š	É	Ų	é	ų
0A			*	:	J	Z	j	z			Ē	ē	Ę	Ú	ę	ú
0B			+	;	K	[	k	{			Ģ	ģ	Ë	Û	ë	û
0C			,	<	L	\	l	|			Ŧ	ŧ	Ė	Ü	ė	ü
0D			-	=	M	]	m	}			SHY	Ð	Í	Ũ	í	ũ
0E			.	>	N	^	n	~			Ž	ž	Î	Ū	î	ū
0F			/	?	O	_	o				¯	ŋ	Ī	ß	ī	˙

Erweiterter BS2000-Code **EBCDIC.DF.04-4** (CCSN: **EDF044**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ø	Ø	°	ĩ	ĸ	ų	Ļ	Ų	0
01					NBSP	é	/	É	a	j	¯	Ŗ	A	J	÷	1
02					â	ę	Â	Ę	b	k	s	Ĩ	B	K	S	2
03					ä	ë	Ä	Ë	c	l	t	ˇ	C	L	T	3
04					ā	č	Ā	Č	d	m	u	Š	D	M	U	4
05					á	í	Á	Í	e	n	v	§	E	N	V	5
06					ã	î	Ã	Î	f	o	w	ļ	F	O	W	6
07					å	ī	Å	Ī	g	p	x	ŧ	G	P	X	7
08					į	ė	ļ	Ė	h	q	y	Ŋ	H	Q	Y	8
09					ŋ	ß	Ņ	¨	i	r	z	ž	I	R	Z	9
0A					`	!	^	:	Ģ	Ē	Ą	Ŧ	SHY	š	˛	ģ
0B					.	$	,	#	ģ	ē	ŋ	[	ô	û	Ô	{
0C					<	*	%	@	đ	æ	Đ	\	ö	ü	Ö	Ü
0D					(	)	_	'	ũ	¸	Ũ	]	ō	Û	Ō	}
0E					+	;	>	=	ū	Æ	Ū	´	ķ	ú	Ķ	Ú
0F					|		?	"	ą	¤	Ž	×	õ	·	Õ	~

Latin/Cyrillic Alphabet **ISO 8859-5** (CCSN: **ISO88595**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	А	Р	а	р	№
01			!	1	A	Q	a	q			Ё	Б	С	б	с	ё
02			"	2	B	R	b	r			Ђ	В	Т	в	т	ђ
03			#	3	C	S	c	s			Ѓ	Г	У	г	у	ѓ
04			$	4	D	T	d	t			Є	Д	Ф	д	ф	є
05			%	5	E	U	e	u			S	Е	Х	е	х	s
06			&	6	F	V	f	v			І	Ж	Ц	ж	ц	і
07			'	7	G	W	g	w			Ї	З	Ч	з	ч	ї
08			(	8	H	X	h	x			J	И	Ш	и	ш	j
09			)	9	I	Y	i	y			Љ	Й	Щ	й	щ	љ
0A			*	:	J	Z	j	z			Њ	К	Ъ	к	ъ	њ
0B			+	;	K	[	k	{			Ћ	Л	Ы	л	ы	ћ
0C			,	<	L	\	l	|			Ќ	М	Ь	м	ь	ќ
0D			-	=	M	]	m	}			SHY	Н	Э	н	э	§
0E			.	>	N	^	n	~			Ў	О	Ю	о	ю	ў
0F			/	?	O	_	o				Џ	П	Я	п	я	џ

Erweiterter BS2000-Code **EBCDIC.DF.04-5** (CCSN: **EDF045**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	j	и	А	Е	Ђ	љ	І	й	0
01					NBSP	щ	/	Щ	a	j	Ц	Ґ	A	J	ї	1
02					т	ъ	Т	Ъ	b	k	s	S	B	K	S	2
03					ф	ы	Ф	Ы	c	l	t	З	C	L	T	3
04					р	ш	Р	Ш	d	m	u	Љ	D	M	U	4
05					с	э	С	Э	e	n	v	Ї	E	N	V	5
06					у	ю	У	Ю	f	o	w	Ж	F	O	W	6
07					х	я	Х	Я	g	p	x	М	G	P	X	7
08					ч	ь	Ч	Ь	h	q	y	Н	H	Q	Y	8
09					ё	п	б	J	i	r	z	О	I	R	Z	9
0A					`	!	^	:	Ћ	Њ	Ё	Ќ	SHY	Й	В	Г
0B					.	$	,	#	Л	К	П	[	є	ћ	д	{
0C					<	*	%	@	№	ц	а	\	і	ќ	ж	м
0D					(	)	_	'	§	И	н	]	ђ	л	в	}
0E					+	;	>	=	ў	Ц	о	Д	ґ	њ	г	к
0F					|		?	"	Б	Є	Ў	з	s	ц	е	~

Erweiterter BS2000-Code **EBCDIC.EHC.LC** (CCSN: **EEHCLC**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ђ	ѓ	є	s	Ъ	Ѓ	Є	S	0
01					NBSP	№	/	ё	a	j	~	Ё	A	J	§	1
02					а	и	р	ш	b	k	s	Т	B	K	S	2
03					б	й	с	щ	c	l	t	У	C	L	T	3
04					в	к	т	ъ	d	m	u	Ф	D	M	U	4
05					г	л	у	ы	e	n	v	Х	E	N	V	5
06					д	м	ф	ь	f	o	w	Ц	F	O	W	6
07					е	н	х	э	g	p	x	Ч	G	P	X	7
08					ж	о	ц	ю	h	q	y	Ш	H	Q	Y	8
09					з	п	ч	я	i	r	z	Щ	I	R	Z	9
0A					`	!	^	:	А	Ж	М	Ъ	SHY	ћ	J	Ц
0B					.	$	,	#	Б	З	Н	Ы	і	ќ	Љ	[
0C					<	*	%	@	В	И	О	Ь	ї	ў	Њ	{
0D					(	)	_	'	Г	Й	П	Э	j	ц	Ћ	]
0E					+	;	>	=	Д	К	Р	Ю	љ	І	Ќ	}
0F					|		?	"	Е	Л	С	Я	њ	Ї	Ў	\

Erweiterter BS2000-Code **EBCDIC.EHC.LC.1** (CCSN: **EEHCLC1**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ђ	s	j	љ	ћ	Њ	Ћ	Ќ	0
01					NBSP	№	/	y	a	j	њ	ќ	A	J	Џ	1
02					a	ё	л	ў	b	k	s	§	B	K	S	2
03					б	ж	м	ф	c	l	t	џ	C	L	T	3
04					в	з	н	x	d	m	u	Ђ	D	M	U	4
05					г	и	о	ц	e	n	v	S	E	N	V	5
06					ѓ	і	п	ч	f	o	w	J	F	O	W	6
07					д	ї	р	ш	g	p	x	Љ	G	P	X	7
08					е	й	с	щ	h	q	y	I	H	Q	Y	8
09					є	к	т	ъ	i	r	z	Ї	I	R	Z	9
0A					`	!	^	:	ы	A	E	Й	M	T	Ч	Э
0B					.	$	,	#	ь	Б	Є	[	H	У	Ш	{
0C					<	*	%	@	э	В	Ё	\	O	Ў	Щ	Ю
0D					(	)	_	'	ю	Г	Ж	]	П	Ф	Ъ	}
0E					+	;	>	=	я	Ґ	З	К	Р	Х	Ы	Я
0F					|		?	"	SHY	Д	И	Л	С	Ц	Ь	~

Latin/Greek Alphabet **ISO 8859-7/ELOT 928** (CCSN: **ISO88597**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	@	P	`	p			NBSP	°	Ï	Π	ü	π
01			!	1	A	Q	a	q			‘	±	A	P	α	ρ
02			"	2	B	R	b	r			’	²	B		β	ς
03			#	3	C	S	c	s			£	³	Γ	Σ	γ	σ
04			$	4	D	T	d	t			€	΄	Δ	T	δ	τ
05			%	5	E	U	e	u			₯	¨	E	Y	ε	υ
06			&	6	F	V	f	v			¦	Ά	Z	Φ	ζ	φ
07			'	7	G	W	g	w			§	·	H	X	η	χ
08			(	8	H	X	h	x			¨	Έ	Θ	Ψ	θ	ψ
09			)	9	I	Y	i	y			©	Ή	I	Ω	ι	ω
0A			*	:	J	Z	j	z			ͺ	Ί	K	Ϊ	κ	ϊ
0B			+	;	K	[	k	{			«	»	Λ	Ÿ	λ	ü
0C			,	<	L	\	l	|			¬	Ό	M	ά	μ	ό
0D			-	=	M	]	m	}			SHY	½	N	έ	ν	ύ
0E			.	>	N	^	n	~				Ύ	Ξ	ή	ξ	ώ
0F			/	?	O	_	o				–	Ώ	O	ί	ο	

Erweiterter BS2000-Code **EBCDIC.DF.04-7** (CCSN: **EDF047**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ψ	Ψ	°	̋	’	ω	¦	Ω	0
01					NBSP	ι	/	Ι	a	j	–	£	A	J	χ	1
02					β	κ	Β	Κ	b	k	s	𝒟ₚ	Β	Κ	S	2
03					δ	λ	Δ	Λ	c	l	t	·	C	L	T	3
04					ϋ	θ	ï	Θ	d	m	u	©	D	M	U	4
05					α	ν	Α	Ν	e	n	v	§	E	N	V	5
06					γ	ξ	Γ	Ξ	f	o	w	Ά	F	O	W	6
07					ε	ο	Ε	Ο	g	p	x	Ό	G	P	X	7
08					η	μ	Η	Μ	h	q	y	½	H	Q	Y	8
09					ρ	ί	Ρ	¨	i	r	z	Ύ	I	R	Z	9
0A					`	!	^	:	«	¸	‘	¬	SHY	Ή	²	³
0B					.	$	,	#	»	Ί	Ώ	[	τ	ü	Τ	{
0C					<	*	%	@	π	ζ	Π	\	φ	ό	Φ	ά
0D					(	)	_	'	ύ	Έ	έ	]	ς	Ÿ		}
0E					+	;	>	=	ώ	Ζ	ή	΄	σ	ï	Σ	Ϊ
0F					|		?	"	±	€		Χ	υ		Υ	~

Erweiterter BS2000-Code **EBCDIC.EHC.LG** (CCSN: **EEHCLG**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	NBSP	–	°		±		¦		0
01					A	Ή	/		a	j		½	A	J	©	1
02					Ά	Θ	Ξ	Υ	b	k	s	²	B	K	S	2
03					B	Ι	Ο	Ύ	c	l	t	³	C	L	T	3
04					Γ	Ί	Ό	Ϋ	d	m	u	‘	D	M	U	4
05					Δ	Ϊ	Π	Φ	e	n	v	’	E	N	V	5
06					E	K	P	X	f	o	w	§	F	O	W	6
07					Έ	Λ		Ψ	g	p	x	¸	G	P	X	7
08					Z	M	Σ	Ω	h	q	y	£	H	Q	Y	8
09					H	N	T	Ώ	i	r	z	´	I	R	Z	9
0A					`	!	^	:	α	έ	ί	¬	v	ς	ΰ	·
0B					.	$	,	#	ά	ζ	ï	[	ξ	σ	φ	{
0C					<	*	%	@	β	η	ΐ	\	o	τ	χ	¨
0D					(	)	_	'	γ	ή	κ	]	ό	υ	ψ	}
0E					+	;	>	=	δ	θ	λ	«	π	ύ	ω	΅
0F					|	SHY	?	"	ε	ι	μ	»	ρ	ϋ	ώ	~

7.3.1.2 7-bit-Leitungscodes und zugehörige EBCDI-Codes für europäische Schriften

Swedish 7-bit Code
ISO 646, Swedish version for names / SEN 850200

	00	10	20	30	40	50	60	70
00			SP	0	É	P	é	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			¤	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
0A			*	:	J	Z	j	z
0B			+	;	K	Ä	k	ä
0C			,	<	L	Ö	l	ö
0D			-	=	M	Å	m	å
0E			.	>	N	Ü	m	ü
0F			/	?	O	_	o	

BS2000-Code **EBCDIC.DF.03.SWE**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/		a	j			A	J		1
02									b	k	s		B	K	S	2
03									c	l	t		C	L	T	3
04									d	m	u		D	M	U	4
05									e	n	v		E	N	V	5
06									f	o	w		F	O	W	6
07									g	p	x		G	P	X	7
08									h	q	y		H	Q	Y	8
09									i	r	z		I	R	Z	9
0A					é	!	Ü	:								
0B					.	$	,	#				Ä				ä
0C					<	*	%	É				Ö				
0D					(	)	_	'				Å				å
0E					+	;	>	=								
0F					ö		?	"								ü

BS2000-Code **EBCDIC.NHC.SWE**

Characters comply with ISO 646-SWE

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-						ä	å	É	0
01							/		a	j	ü		A	J		1
02									b	k	s		B	K	S	2
03									c	l	t		C	L	T	3
04									d	m	u		D	M	U	4
05									e	n	v		E	N	V	5
06									f	o	w		F	O	W	6
07									g	p	x		G	P	X	7
08									h	q	y		H	Q	Y	8
09								é	i	r	z		I	R	Z	9
0A					#	¤	ö	:								
0B					.	Å	,	Ä								
0C					<	*	%	Ö								
0D					(	)	_	'								
0E					+	;	>	=								
0F					!	Ü	?	"								

Hungarian 7-bit code Siemens version **TD7.HUN**

	00	10	20	30	40	50	60	70
00			SP	0	@	P	`	
01			!	1	A	Q	Ft	
02			"	2	B	R	Á	
03			#	3	C	S	É	
04			¤	4	D	T	Í	
05			%	5	E	U	Ó	
06			&	6	F	V	Ö	
07			'	7	G	W	Ő	
08			(	8	H	X	Ú	
09			)	9	I	Y	Ü	
0A			*	:	J	Z	Ű	
0B			+	;	K	[		{
0C			,	<	L	\		|
0D			-	=	M	]		}
0E			.	>	N	^		~
0F			/	?	O	_		

BS2000-Code **EBCDIC.DF.03.HUN**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/		Ft	Ű			A	J		1
02									Á				B	K	S	2
03									É				C	L	T	3
04									Í				D	M	U	4
05									Ó				E	N	V	5
06									Ö				F	O	W	6
07									Ő				G	P	X	7
08									Ú				H	Q	Y	8
09									Ü				I	R	Z	9
0A					`	!	^	:								
0B					.	¤	,	#				[				{
0C					<	*	%	@				\				
0D					(	)	_	'				]				}
0E					+	;	>	=								
0F					|		?	"								~

BS2000-Code **EBCDIC.NHC.HUN**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/					Á	A	J		1
02												É	B	K	S	2
03												Í	C	L	T	3
04												Ó	D	M	U	4
05												Ö	E	N	V	5
06												Ő	F	O	W	6
07												Ú	G	P	X	7
08												Ü	H	Q	Y	8
09					Ft							Ű	I	R	Z	9
0A					`	!	^	:								
0B					.	¤	,	#				[				{
0C					<	*	%	@				\				
0D					(	)	_	'				]				}
0E					+	;	>	=								
0F					|		?	"								~

Cyrillic 7-bit code Siemens version **TD7.CYR**

	00	10	20	30	40	50	60	70
00			SP	0	@	P	Ю	П
01			!	1	A	Q	А	Я
02			"	2	B	R	Б	Р
03			#	3	C	S	Ц	С
04			¤	4	D	T	Д	Т
05			%	5	E	U	Е	У
06			&	6	F	V	Ф	Ж
07			'	7	G	W	Г	В
08			(	8	H	X	Х	Ш
09			)	9	I	Y	И	З
0A			*	:	J	Z	Й	Щ
0B			+	;	K	Ы	К	[
0C			,	<	L	Ь	Л	|
0D			-	=	M	Э	М	]
0E			.	>	N	^	Н	Ч
0F			/	?	O	_	О	

BS2000-Code **EBCDIC.DF.03.CYR**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/		А	Й			А	J		1
02									Б	К	С		В	K	S	2
03									Ц	Л	Т		C	L	T	3
04									Д	М	У		D	M	U	4
05									Е	Н	Ж		E	N	V	5
06									Ф	О	В		F	O	W	6
07									Г	П	Ш		G	P	X	7
08									Х	Я	З		H	Q	Y	8
09									И	Р	Щ		I	R	Z	9
0A					Ю	!	^	:								
0B					.	¤	,	#				Ы				[
0C					<	*	%	@				Ь				
0D					(	)	_	'				Э				]
0E					+	;	>	=								
0F					|		?	"								Ч

BS2000-Code **EBCDIC.NHC.CYR**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/						A	J		1
02												Т	B	K	S	2
03												У	C	L	T	3
04												Ф	D	M	U	4
05												Х	E	N	V	5
06												Ц	F	O	W	6
07												Ч	G	P	X	7
08												Ш	H	Q	Y	8
09												Щ	I	R	Z	9
0A						!	^	:	А	Ж	М					
0B					.	¤	,	#	Б	З	Н	Ы				[
0C					<	*	%	@	В	И	О	Ь				
0D					(	)	_	'	Г	Й	П	Э				]
0E					+	;	>	=	Д	К	Р	Ю				
0F					|		?	"	Е	Л	С	Я				

Greek 7-bit code Siemens version **TD7.GRE**

	00	10	20	30	40	50	60	70
00			SP	0	@	P	`	Π
01			!	1	A	Q	A	P
02			"	2	B	R	B	C
03			#	3	C	S	Γ	Σ
04			$	4	D	T	Δ	T
05			%	5	E	U	E	Y
06			&	6	F	V	Z	Φ
07			'	7	G	W	H	X
08			(	8	H	X	Θ	Ψ
09			)	9	I	Y	I	Ω
0A			*	:	J	Z	K	§
0B			+	;	K	[	Λ	{
0C			,	<	L	\	M	|
0D			-	=	M	]	N	}
0E			.	>	N	^	Ξ	~
0F			/	?	O	_	O	

BS2000-Code **EBCDIC.DF.03.GRE**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/		A	K			A	J		1
02									B	Λ	Σ		B	K	S	2
03									Γ	M	T		C	L	T	3
04									Δ	N	Y		D	M	U	4
05									E	Ξ	Φ		E	N	V	5
06									Z	O	X		F	O	W	6
07									H	Π	Ψ		G	P	X	7
08									Θ	P	Ω		H	Q	Y	8
09									I		§		I	R	Z	9
0A					`	!	^	:								
0B					.	$	,	#				[				{
0C					<	*	%	@				\				
0D					(	)	_	'				]				}
0E					+	;	>	=								
0F					|		?	"								~

BS2000-Code **EBCDIC.NHC.GRE**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01					A		/						A	J		1
02						Θ	Ξ	Y					B	K	S	2
03					B	I	O						C	L	T	3
04					Γ								D	M	U	4
05					Δ		Π	Φ					E	N	V	5
06					E	K	P	X				§	F	O	W	6
07						Λ		Ψ					G	P	X	7
08					Z	M	Σ	Ω					H	Q	Y	8
09					H	N	T						I	R	Z	9
0A					`	!	^	:								
0B					.	$	,	#				[				{
0C					<	*	%	@				\				
0D					(	)	_	'				]				}
0E					+	;	>	=								
0F					|		?	"								~

International 7-bit code **ISO 646-IRV** (CCSN: **ISO646**)

	00	10	20	30	40	50	60	70
00			SP	0	@	P	`	p
01			!	1	A	Q	a	q
02			"	2	B	R	b	r
03			#	3	C	S	c	s
04			¤	4	D	T	d	t
05			%	5	E	U	e	u
06			&	6	F	V	f	v
07			'	7	G	W	g	w
08			(	8	H	X	h	x
09			)	9	I	Y	i	y
0A			*	:	J	Z	j	z
0B			+	;	K	[	k	{
0C			,	<	L	\	l	|
0D			-	=	M	]	m	}
0E			.	>	N	^	n	~
0F			/	?	O	_	o	

BS2000-Code **EBCDIC.DF.03** International (CCSN: **EDF03IRV**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-									0
01							/		a	j			A	J		1
02									b	k	s		B	K	S	2
03									c	l	t		C	L	T	3
04									d	m	u		D	M	U	4
05									e	n	v		E	N	V	5
06									f	o	w		F	O	W	6
07									g	p	x		G	P	X	7
08									h	q	y		H	Q	Y	8
09									i	r	z		I	R	Z	9
0A					`	!	^	:								
0B					.	$	,	#				[				{
0C					<	*	%	@				\				
0D					(	)	_	'				]				}
0E					+	;	>	=								
0F					|		?	"								~

7.3.2 Unterstützte Leitungscodes und BS2000-EBCDI-Codes für arabische Schriften

Erläuterung der Raster

Steuerzeichen nicht belegbar

Position nicht belegbar

EBCDIC.DF03-Kernel

dieses Sonderzeichen wird im arabischen Script nicht unterstützt

wählbare Darstellung im arabischen Script

kompatibilitätsrelevante Siemens-Erweiterung

lateinisches Zeichen, das im arabischen Script verwendet wird

Latin/Arabic Alphabet **ISO 8859-6 / TD8.LA** (CCSN: **ISO88596**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0 ٠	@	P	`	p						ذ	-	ـِ
01			!	1 ١	A	Q	a	q					ء	ر	ف	ّ
02			"	2 ٢	B	R	b	r					آ	ز	ق	ٌ
03			#	3 ٣	C	S	c	s					أ	س	ك	لآ
04			$	4 ٤	D	T	d	t			¤		ؤ	ش	ل	لأ
05			%	5 ٥	E	U	e	u					إ	ص	م	لإ
06			&	6 ٦	F	V	f	v					ئ	ض	ن	لا
07			'	7 ٧	G	W	g	w					ا	ط	ه	
08			(	8 ٨	H	X	h	x					ب	ظ	و	
09			)	9 ٩	I	Y	i	y					ة	ع	ى	
0A			*	:	J	Z	j	z					ت	غ	ي	
0B			+	;	K	[	k	{				؛	ث		ً	
0C			,	<	L	\	l	|			،		ج		ٌ	
0D			-	=	M	]	m	}					ح		ً	
0E			.	>	N	^	n	~					خ		َ	
0F			/	?	O	_	o					؟	د		ُ	

Erweiterter BS2000 Code **EBCDIC.DF.04-6** (CCSN: **EDF046/EDF04A**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-		ظ	0 ٠	5 ٥	"		&	ع	0
01					SP	ى	/	ة	a	j	/	#	A	J		1
02					ق	ي	آ	ت	b	k	s	%	B	K	S	2
03					ل	ً	ؤ	ث	c	l	t	7 ٧	C	L	T	3
04					ـ	و	@	ب	d	m	u	)	D	M	U	4
05					ف	ٍ	ئ	ح	e	n	v	´	E	N	V	5
06					ك	َ	أ	خ	f	o	w	6 ٦	F	O	W	6
07					م	ُ	إ	د	g	p	x	<	G	P	X	7
08					ه	ٌ	ا	ج	h	q	y	=	H	Q	Y	8
09					ّ	_	ر	(	i	r	z	>	I	R	Z	9
0A					`	!	^	:	+	*	!	،	-	9 ٩	2 ٢	3 ٣
0B					.	$	,	#	؛	:	؟	[	لأ	{	ش	{
0C					<	*	%	@	ً	ن	ذ	\	لا	|	ض	\
0D					(	)	_	'	}	8 ٨	]	]	ْ	[	ز	}
0E					+	;	>	=	¯	ئ ى	^	4 ٤	لآ		س	غ
0F					|		?	"	1 ١	¤	.	ط	لإ		ص	~

Erweiterter BS2000 Code **EBCDIC.EHC.LA** (CCSN: **EEHCLAA/EEHCLAI**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ف				0 ٠	-	&	SP	0
01					ء	ت	/	ق	a	j		1 ١	A	J	/	1
02					آ	ث	س	ك	b	k	s	2 ٢	B	K	S	2
03					أ	ج	ش	ل	c	l	t	3 ٣	C	L	T	3
04					ؤ	ح	ص	م	d	m	u	4 ٤	D	M	U	4
05					إ	خ	ض	ن	e	n	v	5 ٥	E	N	V	5
06					ئ	د	ط	ه	f	o	w	6 ٦	F	O	W	6
07					ا	ذ	ظ	و	g	p	x	7 ٧	G	P	X	7
08					ب	ر	ع	ى	h	q	y	8 ٨	H	Q	Y	8
09					ة	ز	غ	ي	i	r	z	9 ٩	I	R	Z	9
0A					`	!	^	:	}	ً	ّ	.	:	^	!	
0B					.	$	,	#	]	ٌ	�°	[	#	،	¤	{
0C					<	*	%	@	\	ٍ	لآ	\	@	%	*	>
0D					(	)	_	'	[	َ	لأ	]	'	_	(	}
0E					+	;	>	=	{	ُ	لإ	)	=	<	؛	+
0F					|	-	?	"	|	ِ	لا	¯	"	؟		~

French/Arabic Alphabet **TD8.NA**

	00	10	20	30		40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0	٠	@	P	`	p						ذ	ـ	ِ
01			!	1	١	A	Q	a	q					ء	ر	ف	ّ
02			"	2	٢	B	R	b	r					آ	ز	ق	ٔ
03			#	3	٣	C	S	c	s					أ	س	ك	Ç
04			$	4	٤	D	T	d	t			¤		ؤ	ش	ل	à
05			%	5	٥	E	U	e	u					إ	ص	م	â
06			&	6	٦	F	V	f	v					ئ	ض	ن	ç
07			'	7	٧	G	W	g	w					ا	ط	ه	è
08			(	8	٨	H	X	h	x					ب	ظ	و	é
09			)	9	٩	I	Y	i	y					ة	ع	ى	ê
0A			*	:		J	Z	j	z					ت	غ	ي	ë
0B			+	;		K	[	k	{				؛	ث	¨	ً	î
0C			,	<		L	\	l	|		،		ج	°	ٌ	ï
0D			-	=		M	]	m	}					ح	¶	ٍ	ô
0E			.	>		N	^	n	~					خ	§	َ	ù
0F			/	?		O	_	o					؟	د		ُ	û

Erweiterter BS2000 Code **EBCDIC.DF.04-NAF** (CCSN: **EDF04E/EDF04F**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	é	ظ	0 ٠	5 ٥	"	ê	&	ع	0
01					SP	ى	/	ة	a	j	/	#	A	J	è	1
02					ق	ي	آ	ت	b	k	s	%	B	K	S	2
03					ل	ً	ؤ	ث	c	l	t	7 ٧	C	L	T	3
04					ـ	و	@	ب	d	m	u	)	D	M	U	4
05					ف	ٍ	ء	ح	e	n	v	´	E	N	V	5
06					ك	َ	أ	خ	f	o	w	6 ٦	F	O	W	6
07					م	ُ	إ	د	g	p	x	<	G	P	X	7
08					٥	ّ	ا	ج	h	q	y	=	H	Q	Y	8
09					ّ	_	ر	(	i	r	z	>	I	R	Z	9
0A					`	!	^	:	+	*	!	،	-	9 ٩	2 ٢	3 ٣
0B					.	$	،	#	؛	:	؟	[	à	î	ش	{
0C					<	*	%	@	ِ	ن	ذ	\	ç	ï	ض	ه
0D					(	)	_	'	ô	8 ٨	¶	]	°	¨	ز	}
0E					+	;	>	=	ù	ئ	§	4 ٤	Ç	ë	س	غ
0F					|		?	"	1 ١	¤	.	ط	â	û	ص	~

Erweiterter BS2000 Code **EBCDIC.EHC.NA** (CCSN: **EEHCNAA/EEHCNAI**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	ف	§		ë	0 ٠	-	&	SP	0
01					ء	ت	/	ق	a	j	î	1 ١	A	J	/	1
02					آ	ث	س	ك	b	k	s	2 ٢	B	K	S	2
03					أ	ج	ش	ل	c	l	t	3 ٣	C	L	T	3
04					ؤ	ح	ص	م	d	m	u	4 ٤	D	M	U	4
05					إ	خ	ض	ن	e	n	v	5 ٥	E	N	V	5
06					ئ	د	ط	ه	f	o	w	6 ٦	F	O	W	6
07					ا	ذ	ظ	و	g	p	x	7 ٧	G	P	X	7
08					ب	ر	ع	ى	h	q	y	8 ٨	H	Q	Y	8
09					ة	ز	غ	ي	i	r	z	9 ٩	I	R	Z	9
0A					`	!	^	:	à	ً	ّ	.	:	Ç	!	¨
0B					.	$	،	#	â	ٌ	°	[	#	،	¤	{
0C					<	*	%	@	ç	ٍ	ï	\	@	%	*	>
0D					(	)	_	'	è	َ	ô	]	'	_	(	}
0E					+	;	>	=	é	ُ	ù	)	=	<	؛	+
0F					|	-	?	"	ê	ِ	û	¶	"	؟	°	~

7.3.3 Unterstützte Leitungscodes und BS2000-EBCDI-Codes für persische Schriften

Erläuterung der Raster:

Steuerzeichen nicht belegbar

Position nicht belegbar

EBCDIC.DF03-Kernel

dieses Sonderzeichen wird im Farsi-Script nicht unterstützt

wählbare Darstellung im Farsi-Script

wird für zusammengesetzte Farsi-Wörter benötigt

lateinisches Zeichen, das im Farsi-Script verwendet wird

Latin/Farsi Alphabet for Data Interchange **TD8.LF**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00			SP	0 ۰	@	P	`	p					ﻉ ﻊ	چ	ض	گ
01			!	1 ۱	A	Q	a	q					آ	ح	ض	گ
02			"	2 ۲	B	R	b	r					اً	ح	ط	ل
03			#	3 ۳	C	S	c	s					ا	خ	ط	ل
04			$	4 ٤	D	T	d	t			﷼		اً	خ	ظ	م
05			%	5 ٥	E	U	e	u					ب	د	ظ	م
06			&	6 ٦	F	V	f	v			÷		ب	ذ	ع	ن
07			'	7 ٧	G	W	g	w					پ	ر	ع	ن
08			(	8 ٨	H	X	h	x					پ	ز	غ	و
09			)	9 ٩	I	Y	i	y					ت	ژ	غ	ؤ
0A			*	:	J	Z	j	z			×		ت	س	ف	ه
0B			+	;	K	[	k	{				؛	ث	س	ف	ه
0C			,	<	L	\	l	|			،		ث	ش	ق	ی
0D			-	=	M	]	m	}					ج	ش	ق	ی
0E			.	>	N	^	n	~					ج	ص	ک	ئ
0F			/	?	O	_	o					؟	چ	ص	ک	_

Erweiterter BS2000 Code **EBCDIC.DF.04-FAR** (CCSN: **EDF04C/EDF04D)**

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	و	ز	0 ۰	5 ٥	"	وُ	÷	ژ	0
01					SP	غ	/	ت	a	j	/	#	A	J	ن	1
02					ط	ف	اً	ت	b	k	s	%	B	K	S	2
03					ظ	ف	أ	ث	c	l	t	7 v	C	L	T	3
04					ض	غ	ءٍ	پ	d	m	u	)	D	M	U	4
05					ض	ق	آ	ج	e	n	v	´	E	N	V	5
06					ط	ک	ا	چ	f	o	w	6 ٦	F	O	W	6
07					ظ	ک	ب	چ	g	p	x	<	G	P	X	7
08					ع	ق	پ	ث	h	q	y	=	H	Q	Y	8
09					گ	ص	ح	(	i	r	z	>	I	R	Z	9
0A					`	!	^	:	+	×	!	،	-	9 ٩	2 ٢	3 ٣
0B					.	$	,	#	؛	:	؟	[	م	ه	خ	{
0C					<	*	%	@	گ	ع	چ	\	ن	ی	ذ	ش
0D					(	)	_	'	ی	8 ٨	ش	]	ل	س	ح	}
0E					+	;	>	=	ئ	ب	ص	4 ٤	ل	ه	خ	س
0F					|		?	"	1 ١	﷼	.	ر	م	_	د	~

Erweiterter BS2000 Code **EBCDIC.EHC.LF** (CCSN: **EEHCLFI/EEHCLFF**)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
00					SP	&	-	6 ٦	ب		خ	س	غ	ک	SP	0
01					#	¤	/	7 ٧	a	j	خ	ش	A	J	ن	1
02					'	%	/	8 ٨	b	k	s	ش	B	K	S	2
03					÷	.	؟	9 ٩	c	l	t	ص	C	L	T	3
04					<	:	0 ٠	ءٍ	d	m	u	ص	D	M	U	4
05					؛	)	1 ١	آ	e	n	v	ض	E	N	V	5
06					>	(	2 ٢	اً	f	o	w	ض	F	O	W	6
07					!	×	3 ٣	ا	g	p	x	ط	G	P	X	7
08					"	+	4 ٤	أ	h	q	y	ط	H	Q	Y	8
09					،	-	5 ٥	ب	i	r	z	ظ	I	R	Z	9
0A					`	!	^	:	پ	ج	د	ظ	غ	گ	ن	ی
0B					.	$	,	#	پ	چ	ذ	[	ف	گ	و	{
0C					<	*	%	@	ت	چ	ر	\	ف	ل	ؤ	ئ
0D					(	)	_	'	ت	چ	ز	]	ق	ل	ه	}
0E					+	;	>	=	ث	ح	ژ	ع	ق	م	ه	_
0F					|	=	?	"	ث	ح	س	ع	ک	م	ی	~

Fachwörter

Abdruckbare Zeichen
Printable Character
siehe darstellbare Zeichen

ASCII
US-Version der 7-bit-Code-Tabelle gemäß ISO646.

Code
siehe Codierter Zeichensatz

Code-Tabellen
Code Tables
Tabellen, die einen Code beschreiben. Jedem zu beschreibenden Code sind mehrere Tabellen zugeordnet. Diese Tabellen werden durch Makros aufgebaut und modifiziert.

Codierter Zeichensatz
Coded Character Set
Regeln, die die eindeutige Zuordnung von Zeichen eines Zeichensatzes mit ihrer Darstellung in Bits festlegen.

Darstellbares Zeichen
Displayable Character
Grafisch darstellbares Zeichen (können auch Leerzeichen sein).

Definiertes Zeichen
Defined Character
Ein bestimmter Wert eines Codes gilt als definiert, wenn diesem Wert ein (abdruckbares oder nicht abdruckbares) Zeichen zugeordnet ist.

EBCDIC Kern
EBCDIC Kernel
Normierter Grundzeichensatz des BS2000/OSD. Bestehend aus den Zeichen A-Z, a-z, 0-9, Leerzeichen und den Zeichen & - / : . , < * % ( ) _ ’ + ; > = ?

Erweiterung
Extension

Ein codierter Zeichensatz B ist eine Erweiterung des codierten Zeichensatzes A, wenn er alle in A definierten sowie zusätzliche Zeichen enthält. In beiden Codes müssen die gemeinsamen Zeichen gleich codiert sein.

Format-Datenstation
Format terminal

Betriebsart einer logischen Datenstation, bei der die Nachricht aus einem Format (= Formular, Maske) besteht.

ISO-Code

Normierter Standard-Code für den Nachrichtenaustausch, wie er z.B. in UNIX, MS-DOS und DEC-Systemen vorkommt. Der Ursprung des ISO-Codes liegt im ASCII-Code.

ISO-Code-Variantennummer
ISO code variant number

Nummer, die eine normierte Variante des ISO 8859-Codes und des entsprechenden kompatiblen EBCDI-Codes bezeichnet.

Kompatible Codes
Compatible coded character sets

Zwei Codes werden kompatibel genannt, wenn sie den gleichen Zeichensatz enthalten. Codierung bzw. Wert der einzelnen Zeichen können jedoch unterschiedlich sein.

Logische Datenstation
Virtual terminal

Modellvorstellung einer Datenstation, deren Funktionen auf die physikalischen Eigenschaften unterschiedlicher Datenstationen abgebildet werden.

Nachricht
Message

Eine logisch zusammengehörige Datenmenge, die an einen Kommunikationspartner gesendet werden soll bzw. von einem Kommunikationspartner empfangen werden soll.

Protokoll
Protocol

Regeln, nach denen die Kommunikation in Rechnernetzen abläuft.

Prozess
Process
Instanz zur Ausführung eines Programms innerhalb einer Task.

Referenzcode
Reference Code
Innerhalb einer Gruppe kompatibler Codes beziehen sich die Umwandlungstabellen auf einen gemeinsamen Zielcode, den Gruppenreferenzcode. Der Referenzcode enthält alle Zeichen der anderen Codes.

Sonderzeichen
Special character
Ein grafisch darstellbares Zeichen, das weder ein Buchstabe, noch eine Ziffer oder ein Leerzeichen ist.

Statusabfrage
Status inquiry
Über eine Statusabfrage erhalten Sie Grundinformationen über das Terminal (z.B. Stationstyp, Art und Anzahl der logisch ansprechbaren Zeichensätze, Anzahl der Farben) und eine Beschreibung der angeschlossenen Peripherie (z.B. Ausweisleser, Chipkartenterminal).

Zeichensatz
Character set
Satz von Buchstaben, Ziffern und Sonderzeichen, aus denen Wörter und andere elementare Bestandteile einer Sprache (auch Computersprache) aufgebaut sind.

Zeilen-Datenstation
Line terminal
Betriebsart einer logischen Datenstation, bei der die Nachricht in Form von Zeilen strukturiert ist.

Zugriffsmethode
Communication access method
Software, die den Anwendungen Schnittstellen zur Kommunikation bietet.

7-bit-Code
Ein ISO/ASCII-Code, der nur 7 Bits benutzt und der äquivalente EBCDI-Code.

8-bit-Code
Ein ISO-Code, der 8 Bits benutzt und der äquivalente EBCDI-Code.

Abkürzungen

BCAM	Basic Communication Access Method
CCS	Codierter Zeichensatz (Code) (Coded Character Set)
CCSN	Name des codierten Zeichensatzes (Coded Character Set Name)
DCAM	Data Communication Access Method
DRV	Deutsche Referenz-Version
DSSM	Verwaltung des dynamischen Subsystems (Dynamic Subsystem Manager)
EBCDIC	Binärcode für die stellenweise Verschlüsselung von Dezimalziffern (Extended Binary-Coded Decimal Interchange Code)
EMDS	Emulation Terminal
IRV	Internationale Referenz-Version
ISO	International Organization of Standardization
NLS	Unterstützung der Landessprache (Native Language Support)
PDN	Programmsystem für Datenfernverarbeitung und Netzsteuerung
TIAM	Terminal Interactive Access Method
openUTM	Universeller Transaktionsmonitor
UCS	Universal Character Set
UTF-8	Universal Character Set Transformation Format 8 Bit
UTF-16	Universal Character Set Transformation Format 16 Bit
UTFE	Universal Character Set Transformation Format EBCDIC

VTSU	Programmsystem zur Entkopplung von Anwendungsprogrammen (Virtual Terminal Support)
VTSUCB	VTSU-Control-Block

Literatur

Die Handbücher sind online unter *http://manuals.fujitsu-siemens.com* zu finden oder in gedruckter Form gegen gesondertes Entgelt unter *http://FSC-manualshop.com* zu bestellen.

[1] **VTSU**
Virtual Terminal Support
Benutzerhandbuch

[2] **BS2000/OSD-BC**
Einführung in die Systembetreuung
Benutzerhandbuch

[3] **BS2000/OSD-BC**
Systeminstallation
Benutzerhandbuch

[4] **BS2000/OSD-BC**
Kommandos Band 1 - 5
Benutzerhandbuch

[5] **BS2000/OSD-BC**
Kommandos Band 6, Ausgabe in S-Variablen und SDF-P-BASYS
Benutzerhandbuch

[6] **BS2000/OSD-BC**
Dienstprogramme
Benutzerhandbuch

[7] **BS2000/OSD**
Makroaufrufe an den Ablaufteil
Benutzerhandbuch

[8] **BS2000/OSD-BC**
System Exits
Benutzerhandbuch

[9] **IMON** (BS2000/OSD)
Installationsmonitor
Benutzerhandbuch

[10] **EMDS** (SINIX)
Benutzerhandbuch

[11] ***open*Net Server** (BS2000/OSD)
BCAM
Benutzerhandbuch

[12] **Generierung eines Datenkommunikationssystems** (TRANSDATA)
Benutzerhandbuch

[13] **TransView-NMA/-NMAE V1.2A, TransView-NTAC2 V7.1A, NTAC2E V5.1A**
(TRANSDATA, BS2000)
Netzmanagement im BS2000
Benutzerhandbuch

[14] **TransView-NMA** (PDN)
TransView-NMAE (PDN)
Netzmanagement und Messdatenerfassung im PDN (TRANSDATA, PDN)
Kommandos
Benutzerhandbuch

[15] **Netzzugang für Datenstationen** (TRANSDATA)
Benutzerhandbuch

[16] **SDF** (BS2000/OSD)
SDF-Verwaltung
Benutzerhandbuch

[17] **DCAM** (BS2000/OSD, TRANSDATA)
Makroaufrufe
Benutzerhandbuch

[18] **DCAM** (BS2000/OSD, TRANSDATA)
COBOL-Aufrufe
Benutzerhandbuch

[19] **DCAM** (BS2000/OSD, TRANSDATA)
Programmschnittstellen
Beschreibung

[20] **TIAM** (TRANSDATA, BS2000/OSD)
Benutzerhandbuch

[21] **FHS** (TRANSDATA)
Benutzerhandbuch

[22] **IFG für FHS** (TRANSDATA)
Benutzerhandbuch

[23] **openUTM** (BS2000/OSD, UNIX, Windows)
Anwendungen generieren
Benutzerhandbuch

[24] **openUTM** (BS2000/OSD, UNIX, Windows)
Anwendungen programmieren mit KDCS für COBOL, C und C++
Benutzerhandbuch

[25] **openUTM**
Konzepte und Funktionen
Benutzerhandbuch

[26] **EDT** (BS2000/OSD)
Anweisungen
Benutzerhandbuch

[27] **PERCON V2.9A** (BS2000/OSD)
Benutzerhandbuch

[28] **SORT V7.9A** (BS2000/OSD)
Benutzerhandbuch

[29] **RSO** (BS2000/OSD)
Remote SPOOL Output
Benutzerhandbuch

[30] **LMS** (BS2000/OSD)
SDF-Format
Benutzerhandbuch

[31] **BS2000/OSD**
Softbooks Deutsch
CD-ROM

[32] **Unicode im BS2000/OSD**
Übersichtshandbuch

Literatur zu Unicode im Internet

http://www.unicode.org
Homepage des Unicode-Konsortiums

http://www.unicode.org/reports/tr10/
Unicode Collation Algorithm

http://www.unicode.org/reports/tr15/
Unicode Normalization Forms

http://www.unicode.org/reports/tr16/
UTF-EBCDIC

http://www.unicode.org/charts
Unicode Code Charts

http://www.unicode.org/Public/UCA/4.0.0/allkeys-4.0.0.txt
Unicode Default Collation Table

http://www.unicode.org/Public/UNIDATA/UnicodeData.txt
Unicode Character Database

http://developer.mimer.com/collations/charts/index.tml
u.a. nationale Sortier-Sonderregeln

http://unicode.e-workers.de/
Allgemeine Einführung in Unicode

Stichwörter

Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format *...@ts.fujitsu.com*.

The Internet pages of Fujitsu Technology Solutions are available at *http://ts.fujitsu.com/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form *...@ts.fujitsu.com*.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter *http://de.ts.fujitsu.com/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.