

\*4 Fujitsu Technology Solutions

COBOL85-BC, COBOL85 und COBOL85-R (BS2000/OSD)

\*5 Version 2.3A  
Juni 2012

Freigabemitteilung

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

Copyright © Fujitsu Technology Solutions 2012

<b>1 Allgemeines</b>	<b>3</b>
1.1 Bestellung	4
1.2 Auslieferung	4
1.3 Dokumentation	7
<b>2 Software-Erweiterungen</b>	<b>8</b>
2.1 Unterstützung bei der Umstellung auf den nächsten COBOL-Standard	8
2.2 Behandlung von Segmentierung	8
2.3 Unterstützung beim Jahrhundertwechsel	8
2.4 Weitere Spracherweiterungen	9
2.5 Unterstützung von RISC-Codegenerierung	9
2.6 Unterstützung des EURO-Zeichens	10
2.7 Produktinterne SW-Anpassung	10
<b>3 Technische Hinweise</b>	<b>11</b>
3.1 Ressourcenbedarf	11
3.2 SW-Konfiguration	11
3.3 Produkt-Installation	13
3.3.1 Allgemeines	13
3.3.2 Installation zum Einsatz mit COMOPT-Steuerung	13
3.3.3 Installation zum Einsatz mit SDF-Steuerung	13
3.3.4 Installation zum Einsatz im POSIX	14
3.3.5 Installation für den Einsatz der Strukturiererfunktionen	15
3.4 Produkt-Einsatz	15
3.5 Entfallene (und gekündigte) Funktionen	15
3.6 Inkompatibilitäten	15
3.7 Einschränkungen	16
3.8 Verhalten im Fehlerfall	19
<b>4 Hardware-Anforderungen</b>	<b>20</b>
<b>5 Anhang</b>	<b>21</b>
5.1 Konvertierung relativer Dateien mit FCBTYP= PAM	21

# 1 Allgemeines

\*5 Diese Freigabemitteilung enthält in gedrängter Form die wesentlichen Erweiterungen, Abhängigkeiten und Betriebshinweise zu COBOL85 V2.3A unter dem Betriebssystem BS2000.

\*5 Der Inhalt entspricht dem Freigabestand: Juni 2012.

\*5 Änderungen gegenüber Freigabestand Oktober 2000 sind mit \*4 gekennzeichnet.

\*5 Änderungen gegenüber Freigabestand März 2009 sind mit \*5 gekennzeichnet.

\*5 Diese und andere aktuelle Freigabemitteilungen sind auf der SoftBooks-DVD enthalten und online verfügbar unter <http://manuals.ts.fujitsu.com/>.

COBOL85, der COBOL-Compiler im BS2000/OSD \*), ermöglicht Programmierung nach dem COBOL-Standard ANSI X3.23-1985 (= ISO/IEC 1989-1985 = EN 21989 = DIN 66028-1986).

Ab COBOL85 (BS2000) V2.1A wird der Zusatz zur Norm ANSI X3.23a-1989 bzw. ISO/IEC 1989 Amendment1, Intrinsic Function Module, unterstützt.

Wie bereits die Version 2.0A, unterstützt auch COBOL85 V2.3A vollständig den COBOL-Standard X3.23-1985 High Level.

Außer den vom Standard verlangten Pflichtmodulen werden die optionalen Module "Report Writer" und "Segmentation" im vollen Funktionsumfang zur Verfügung gestellt.

Ebenso enthält COBOL85 die "Data Manipulation Language DML" zur Bedienung des Datenbanksystems UDS.

Mit COBOL85 V2.3A ist es möglich, sich neue Schlüsselwörter des zukünftigen Standards anzeigen zu lassen.

Neben den Spracherweiterungen werden in COBOL85 V2.3A zusätzlich POSIX- und XPG4-Programmierschnittstellen unterstützt.

Ab COBOL85 (BS2000) V2.1A wird mit COBOL85 nur noch der COBOL-Compiler ohne Laufzeitsystem ausgeliefert. Das gemeinsame Laufzeitsystem für COBOL85, C und C++ im BS2000 ist das Common Run-Time Environment CRTE.

COBOL85 V2.3A wird in drei Varianten angeboten:

- Basisvariante (BC - Basic Configuration)
- Vollvariante
- RISC-Variante (R - RISC-Code erzeugend)

Letztere wird nicht allgemein, sondern nur auf speziellen Wunsch (Sonderfreigabe) freigegeben.

\*4 \*) BS2000/OSD (R) ist ein eingetragenes Warenzeichen von  
\*4 Fujitsu Technology Solutions

Die gegenüber der Vollvariante in der Basisvariante nicht unterstützten Funktionen sind nachfolgend aufgeführt:

- REPORT WRITER
- CODASYL-DML (Native UDS-Unterstützung)
- Symbolisches Testen mit AID
- COBOL85-Strukturierer
- Ausgabe einer Liste aller Fehlermeldungen
- Ausgabe von Objektlisten
- POSIX-Schnittstellen

Die RISC-Variante enthält gegenüber der Vollvariante die zusätzliche Funktion:

- Erzeugung von RISC-Code

Diese Freigabemitteilung beschreibt, wenn nichts anderes angegeben, die Vollvariante.

Werden bei der Produktinstallation eine oder mehrere Vorgängerversionen übersprungen, so sind bei der Installation der aktuellen Produktversion auch die Installationshinweise aus den Freigabemitteilungen der Vorgängerversionen zu berücksichtigen.

## 1.1 Bestellung

\*4

Die Software können Sie bei der für Sie zuständigen Region von Fujitsu Technology Solutions bestellen. Für dieses Produkt gelten die allgemeinen Bedingungen zum Vertrag über die Nutzung und Betreuung von Softwareprodukten.

## 1.2 Auslieferung

Die Auslieferung der Dateien zu COBOL85 V2.3A erfolgt mit dem Lieferverfahren SOLIS. Im SOLIS2-Lieferanschreiben sind die jeweils gültigen Datei- und Datenträgermerkmale aufgeführt.

Produktbestandteile der Basisvariante:

SINPRC.COBOL85-BC.023.DO-INST	Aufruf Installationsprozedur
SINPRC.COBOL85-BC.023.INSTAL	interne Installationsprozedur
SYSFGM.COBOL85-FGM.023.D	Freigabemitteilung (deutsch)
SYSFGM.COBOL85-FGM.023.E	Freigabemitteilung (englisch)
SYSMES.COBOL85-BC.023	Meldungsdatei (MSGMAKER-Format)
SYMSV.COBOL85-BC.023	Meldungs-Primärdatei
SYSPRG.COBOL85-BC.023	Compiler

	SYSSDF.COBOL85-BC.023	System-Syntaxdatei für BS2000/OSD
	SYSSDF.COBOL85-BC.023.INSTAL	User-Syntaxdatei für Installation
	SYSSDF.COBOL85-BC.023.SYSTEM	System-Syntaxdatei für BS2000 V10
	SYSSDF.COBOL85-BC.023.USER	User-Syntaxdatei
	SYSSII.COBOL85-BC.023	IMON Informationsdatei
	SYSSPR.COBOL85-BC.023.COMPILE	SDF-Startprozedur für Compiler
	Produktbestandteile der Vollvariante:	
	SINLIB.COBOL85.023	Bibliothek zur Installation im POSIX
	SINPRC.COBOL85.023.DO-INST	Aufruf Installationsprozedur
	SINPRC.COBOL85.023.INSTAL	interne Installationsprozedur
*5	SYFSGM.COBOL85.023.D	Freigabemitteilung (deutsch)
*5	SYFSGM.COBOL85.023.E	Freigabemitteilung (englisch)
	SYSMES.COBOL85.023	Meldungsdatei (MSGMAKER- Format)
	SYSMSV.COBOL85.023	Meldungs-Primärdatei
	SYSPRG.COBOL85.023	Compiler
	SYSPRG.COBOL85.023.IND	Strukturierer Beautify
	SYSPRG.COBOL85.023.VERDI	Strukturierer Pretty-Print
	SYSSDF.COBOL85.023	System-Syntaxdatei für BS2000/OSD
	SYSSDF.COBOL85.023.INSTAL	User-Syntaxdatei für Installation
	SYSSDF.COBOL85.023.SYSTEM	System-Syntaxdatei für BS2000 V10
	SYSSDF.COBOL85.023.USER	User-Syntaxdatei
	SYSSII.COBOL85.023	IMON Informationsdatei
	SYSSPR.COBOL85.023.COMPILE	SDF-Startprozedur für Compiler
	SYSSPR.COBOL85.023.STRUCT	SDF-Startprozedur für Strukturierer

## Produktbestandteile der RISC-Variante:

SINLIB.COBOL85-R.023	Bibliothek zur Installation im POSIX
SINPRC.COBOL85-R.023.DO-INST	Aufruf Installationsprozedur
SINPRC.COBOL85-R.023.INSTAL	interne Installationsprozdur
SYFSGM.COBOL85-FGM.023.D	Freigabemitteilung (deutsch)
SYFSGM.COBOL85-FGM.023.E	Freigabemitteilung (enlisch)
SYSLNK.COBOL85-R.023.ASS	Nachladebibliothek mit ASSTRAN
SYSTEMS.COBOL85-R.023	Meldungsdatei (MSGMAKER- Format)
SYMSV.COBOL85-R.023	Meldungs-Primärdatei
SYSPRG.COBOL85-R.023	Compiler
SYSPRG.COBOL85-R.023.IND	Strukturierer Beautify
SYSPRG.COBOL85-R.023.VERDI	Strukturierer Pretty-Print
SYSSDF.COBOL85-R.023	System-Syntaxdatei für BS2000/OSD
SYSSDF.COBOL85-R.023.INSTAL	User-Syntaxdatei für Installation
SYSSDF.COBOL85-R.023.USER	User-Syntaxdatei
SYSSII.COBOL85-R.023	IMON Informationsdatei
SYSSPR.COBOL85-R.023.COMPILE	SDF-Startprozedur für Compiler
SYSSPR.COBOL85-R.023.STRUCT	SDF-Startprozedur für Strukturierer

### 1.3 Dokumentation

Folgende Handbücher gelten für COBOL85 V2.3A (sowohl für die Basisvariante, den Vollausbau, als auch für die RISC-Variante):

COBOL85 (BS2000/OSD) V2.3  
Sprachbeschreibung

COBOL85 (BS2000/OSD) V2.3  
Language Reference Manual

COBOL85 (BS2000/OSD) V2.3  
Benutzerhandbuch

COBOL85 (BS2000/OSD) V2.3  
User Guide

COBOL85 (BS2000/OSD) V2.3  
Tabellenheft

COBOL85 (BS2000/OSD) V2.3  
Ready Reference

Daneben sind das Handbuch zu CRTE (BS2000/OSD) V2.1 sowie die Handbücher der BS2000-Standardkonfiguration für den Betrieb von COBOL85 empfehlenswert.

Zusätzlich werden beim Einsatz der Vollvariante des Compilers  
- zum Testen mit AID die entsprechenden AID-Manuale,  
- zum Arbeiten mit POSIX die entsprechenden POSIX-Manuale  
empfohlen.

\*5 Die Dokumentation zu BS2000/OSD ist unter dem Titel BS2000/OSD SoftBooks in  
\*5 deutscher und englischer Sprache auf DVD erhältlich.

\*5 Die Dokumentation ist auch als Online-Manuale unter <http://manuals.ts.fujitsu.com>  
\*5 verfügbar oder gegen gesondertes Entgelt unter <http://manualshop.ts.fujitsu.com>  
\*5 zu bestellen. Im Manualshop sind ausschließlich Manuale zu den im Vertrieb be-  
\*5 findlichen Produktversionen bestellbar.

\*5 Zu diesen Handbüchern kann es zusätzlich README-Dateien geben. Sie enthal-  
\*5 ten Änderungen und Erweiterungen zum Handbuch des jeweiligen Produktes. Die  
\*5 Dateien haben folgenden Namensaufbau:

\*5 SYSRME.<produkt>.<version>.D (Datei mit deutschem Text)

\*5 SYSRME.<produkt>.<version>.E (Datei mit englischem Text)

\*5 Beim Drucken der Dateien sollte im PRINT-FILE-Kommando der Operand  
\*5 CONTR-CHAR=EBCDIC angegeben werden.

## 2 Software-Erweiterungen

Im Folgenden werden die wichtigsten Erweiterungen und Verbesserungen gegenüber der Vorgängerversion COBOL85 V2.2C beschrieben.

### 2.1 Unterstützung bei der Umstellung auf den nächsten COBOL-Standard

Der Compiler weist den Anwender bei Bedarf auf die neuen Schlüsselwörter des nächsten COBOL-Standards hin. Um diese Funktionalität zu nutzen, ist die COMOPT MARK-NEW-KEYWORDS bzw. die gleichnamige SDF-Option zu verwenden.

Die neue Compiler-Option dient der Erleichterung des Umstiegs auf den nächsten COBOL-Standard. Bei gesetzter Option werden Schlüsselwörter aus dem künftigen Standard mit Fehlermeldungen der Severity I versehen. Hierdurch kann bei Neuprogrammierung unnötiger Umstellungsaufwand vermieden werden, bzw. bei bestehenden Programmen schon jetzt mit den Umstellungsarbeiten begonnen werden.

Syntax und Semantik der Compiler-Option sind im Benutzerhandbuch dokumentiert.

### 2.2 Behandlung von Segmentierung

Mit dem COBOL85-Standard wurden die Sprachmittel zur Segmentierung obsolet. Sie werden im nächsten Standard nicht mehr enthalten sein.

Aus diesem Grunde wurde eine Compiler-Option zur Steuerung der Behandlung von Segmentierung geschaffen. Der Name der COMOPT und ihre Werte lauten ELABORATE-SEGMENTATION=YES/NO. Die entsprechende SDF-Option lautet SEGMENTATION=ELABORATE/IGNORE.

Hat ELABORATE-SEGMENTATION den Wert NO bzw. wurde bei der entsprechenden SDF-Option SEGMENTATION=IGNORE spezifiziert, werden segmentierungsbezogene Sprachmittel ignoriert und mit einer Warnung quittiert. Dieses Verhalten entspricht der DEFAULT-Einstellung des jetzt ausgelieferten Compilers.

Hat ELABORATE-SEGMENTATION den Wert YES bzw. wurde bei der entsprechenden SDF-Option SEGMENTATION=ELABORATE spezifiziert, wird wie bisher Segmentierung unterstützt, sofern das Programm keine geschachtelten Programme enthält.

Syntax und Semantik der Compiler-Option sind im Benutzerhandbuch dokumentiert.

### 2.3 Unterstützung beim Jahrhundertwechsel

Die Funktionen DATE-TO-YYYYMMDD, DAY-TO-YYYYDDD und YEAR-TO-YYYY dienen der Konvertierung zweistelliger Jahreszahlen in vierstellige Jahreszahlen. Sie sind eine Erweiterung zum COBOL85-Standard und werden Teil des nächsten COBOL-Standards.



Ergänzend hierzu wurde bei der SORT/MERGE-Anweisung eine Spracherweiterung implementiert, die es möglich macht, zweistellige Jahreszahlen in Abhängigkeit von einem festgelegten Jahrhundertfenster als SORT-Schlüssel zu benutzen. Diese Erweiterung der Schnittstelle zum SORT dient der Erleichterung der Umstellung zum Jahr 2000. Die entsprechende Spracherweiterung ist kein Bestandteil eines COBOL-Standards.

Syntax und Semantik der Funktionen und Anweisungen sind in der Sprachbeschreibung dokumentiert.

## 2.4 Weitere Spracherweiterungen

Bei der STRING-Anweisung darf die DELIMITED BY SIZE Angabe unmittelbar vor der INTO Angabe fehlen. In diesem Fall wird DELIMITED BY SIZE angenommen. Diese Spracherweiterung wird Teil des nächsten COBOL-Standards werden.

Syntax und Semantik der STRING-Anweisung sind in der Sprachbeschreibung dokumentiert.

## 2.5 Unterstützung von RISC-Codegenerierung

Hinweis: Dieser Punkt gilt nur für die RISC-Variante.

Die RISC-Variante des COBOL85-Compiler kann sowohl /390- als auch RISC-Code erzeugen. CROSS-Compilierung wird unterstützt.

\*2 Bei RISC-Codegenerierung wird spezifischer Objektcode für den RISC-Prozessor generiert. Dieser ist nur auf SR2000/MIPS ablauffähig. Der erzeugte Objektcode ist effizienter, da er native und nicht über die Firmware auf dem MIPS-Prozessor abläuft.

\*2 Die Effizienzsteigerung ist nur aus der Realzeit, nicht aus der auf SR2000 Anlagen in den meisten Fällen standardmäßig eingestellten "normierten CPU-Zeit" ersichtlich. Die normierte CPU-Zeit gewichtet nämlich den CPU-Verbrauch von /390-Code und RISC-Code unterschiedlich, um eine normierte Abrechnung der benötigten Elementaroperationen, die sich immer auf /390-Operationen beziehen, zu ermöglichen.

Um diese Funktionalität zu nutzen, ist die COMOPT GENERATE-RISC-CODE bzw. die SDF-Option DESTINATION-CODE=RISC-4000 anzugeben.

Syntax der entsprechenden COMOPT lautet:

COMOPT GENERATE-RISC-CODE=YES/NO  
bzw.  
COMOPT GEN-RISC=YES/NO

Es wird der zu generierende Zielcode vereinbart:  
- Bei Angabe von NO wird /390-Code erzeugt.  
- Bei Angabe von YES wird RISC-Code erzeugt.

Die Syntax der SDF-Option wird im folgenden dargestellt:

```
COMPILER-ACTION=.../MODULE-GENERATION(...)
```

```
MODULE-GENERATION(...)
```

```
| ...  
| ,DESTINATION-CODE=STD/RISC-4000
```

```
DESTINATION-CODE=STD/RISC-4000
```

Es wird der zu generierende Zielcode vereinbart:

- Bei Angabe von STD wird /390-Code erzeugt.
- Bei Angabe von RISC-4000 wird RISC-Code erzeugt.

Bei der Generierung von RISC-Code werden segmentierungsbezogene Sprachmittel ignoriert (siehe auch Abschnitt 2.2).

Die Erzeugung von RISC-Code wird nicht im POSIX unterstützt. RISC-Objekte können auch nicht im POSIX abgelegt werden.

Beim Bearbeiten von Dateien ist darauf zu achten, dass das Öffnen und Schließen sowie die Dateizugriffe im gleichen Prozessormodus (/390-Modus bzw. RISC-Modus) erfolgen. Die Einhaltung dieser Vorschrift ist immer gewährleistet, wenn keine EXTERNAL-Klausel in der Dateierklärung verwendet wurde.

## \*2 2.6 Unterstützung des EURO-Zeichens

\*2  
\*2  
\*2  
\*2  
\*2  
\*2

Ab der Version V02.3A10 wird das EURO-Zeichen in der CURRENCY SIGN-Klausel unterstützt und als Währungssymbol in der PICTURE-Klausel zugelassen. Die so aufbereiteten Felder können zur weiteren Verarbeitung oder zur Ausgabe benutzt werden.

## \*5 2.7 Produktinterne SW-Anpassung

\*5  
\*5  
\*5  
\*5

In der Version 2.3A23 wurde produktinterne SW-Anpassung zur Verbesserung der Wartbarkeit realisiert.

## 3 Technische Hinweise

### 3.1 Ressourcenbedarf

Die mit dem Produkt ausgelieferten Dateien belegen die in der folgenden Tabelle angegebene Anzahl von PAM-Seiten.

	Variante:	Ressourcenbedarf:
*5	Basisvariante	ca. 2500 PAM-Seiten
	Vollvariante	ca. 9900 PAM-Seiten
	RISC-Variante	ca. 13600 PAM-Seiten

Wird bei der Vollvariante das POSIX-Dateisystem nicht genutzt, so kann die Bibliothek SINLIB.COBOL85.023 gelöscht werden.

Der Compiler benötigt bei Übersetzungen mindestens 11 MB virtuellen Adressraum.

Bei der Erzeugung von RISC-Code mit der RISC-Variante wird deutlich mehr virtueller Adressraum bei der Übersetzung benötigt. Der Bedarf lässt sich durch  $32 \text{ MByte} + \text{DATA} + \text{NLOC} * k \text{ KByte}$  in etwa abschätzen. Dabei bezeichnen NLOC die Anzahl Sourcezeilen ohne Kommentare und DATA die Größe der benutzerspezifischen Daten im Modul. Die Konstante k liegt zwischen 0.5 und 1. Die Erzeugung von RISC-Code benötigt auch deutlich mehr Zeit.

Bei Nutzung der POSIX-Funktionalität (ENABLE-UFS-ACCESS=YES) vergrößert sich der Platzbedarf der gebundenen Programme. Durch das Binden mit der CRTE-Bibliothek SYSLNK.CRTE.PARTIAL-BIND und dem Vorladen des Subsystems CRTESIS, einer Komponente des CRTE, lässt sich der Platzbedarf wieder verringern.

### 3.2 SW-Konfiguration

Die Freigabe von COBOL85 V2.3A erfolgt für BS2000/OSD ab V1.0A bzw. OSD-SVP ab V2.0A. Sie benötigen das Korrekturpaket

BS2000/OSD-BC V1.0A (ab SOLIS-Änderungsstand 30), bzw.  
 BS2GA.BS2OSD V2.0A (ab SOLIS-Änderungsstand 10), bzw.  
 BS2GA.BS2OSD V3.0A (ab SOLIS-Änderungsstand 03).

In diesen Korrekturpaketen ist CRTE-MSG V1.1 enthalten.

Zum Ablauf von COBOL85 V2.3A-Objekten ist die Installation des Produkts "Common Run-Time Environment" CRTE ab V2.1A erforderlich. CRTE ist eine eigene Liefereinheit und muss für die Modelle mit /390-Systemarchitektur getrennt bestellt werden.

CRTE V2.1A wird für OSD ab V1.0 freigegeben und ist in OSD-SVP ab V2.0A für die SR2000 Modelle enthalten.

Die Nutzung des POSIX-Dateisystems durch den Compiler bzw. durch die erzeugten Objekte (ENABLE-UFS-ACCESS=YES) ist erst ab OSD V2.0 möglich. Dafür muss das POSIX-Subsystem geladen sein. Mit CRTE V2.1A produzierte POSIX-Anwendungen setzen unter OSD V2.0 POSIX-BC V1.1, unter OSD V3.0 POSIX-BC V3.0 und unter OSD V3.1 POSIX-BC V3.1 voraus.

Zur Übersetzung werden folgende Produkte vorausgesetzt:

Produkt	ab Version
CRTE	V2.1A
SDF	V3.0
SDF-P-BASYS	V1.0B
SDF-P (1)	V1.0
ILAM/PLAM (2)	V3.0A
LLMAM (5)	V2.4
SDF-A (3)	V3.0
UDS/SQL (4)	V1.2A
BINDER	V1.2

- (1) nur beim Übersetzen im BS2000 mit Zugriff auf COPY-Elemente im POSIX-Dateisystem oder zum Schreiben von Listings ins POSIX-Dateisystem
- (2) bei Nutzung der POSIX-Funktionalität ab V3.0B
- (3) nur bei Verwendung der Installationsprozedur; alternativ kann eine passende kostenfreie SDF-U Version (siehe Abschnitt 3.3.3) eingesetzt werden
- (4) nur, wenn entsprechende COBOL-Sprachmittel/Systemdienste verwendet werden
- (5) bei RISC-Codegenerierung mit der RISC-Variante ab V3.1A10

Zum Ablauf von COBOL85-Objekten werden folgende Produkte vorausgesetzt:

Produkt	ab Version
CRTE	V2.1A
BINDER (1)(2)	V1.0
SDF-P-BASYS (1)	V1.0B
AID (1)(3)	V2.1A
LMS (1)	V3.0
JV (1)	(6)
SORT (1)(7)	V7.4
UDS/SQL (1)(4)	V1.2A
UTM (5)	V3.3

- (1) nur, wenn entsprechende COBOL-Sprachmittel/Systemdienste verwendet werden

- (2) nur wenn LLMs erzeugt wurden/verarbeitet werden sollen, sonst reicht TSOSLNK bzw. DBL; für das Kommando cobol in der POSIX-Shell ist mindestens BINDER V1.2 notwendig
- (3) falls AID-FE verwendet wird, ist V2.1B erforderlich
- (4) spezielle FIND-7 Features erfordern ggf. eine entsprechend neuere UDS/SQL-Version
- (5) nur wenn COBOL-Objekte als UTM-Teilprogramme ablaufen sollen
- (6) ablauffähig ab BS2000 Version 10
- (7) Spracherweiterung der SORT/MERGE-Anweisung (siehe Abschnitt 2.3) setzt V7.7 voraus

### 3.3 Produkt-Installation

#### 3.3.1 Allgemeines

Die Standard-Installation erfolgt durch das Verfahren SOLIS bzw. ab BS2000/OSD V3 durch das Produkt IMON. Ist in Ihrem System das Produkt IMON (Installation MONitor) gestartet, kann die Standardinstallation über IMON erfolgen.

Dabei werden alle Produktdateien in das BS2000-System eingebracht.

Wollen Sie COBOL85 bzw. COBOL85-R auch im POSIX installieren, so rufen Sie nach erfolgreicher SOLIS/IMON-Installation unter TSOS das POSIX-Installationsprogramm (Beschreibung siehe Handbuch "POSIX Grundlagen") auf mit:

```
/CALL-PROCEDURE(SINPRC.POSIX-BC.<version>,POSINST)
```

Wählen Sie den Punkt 'Install packages on POSIX' und geben Sie im weiteren als Produkt COBOL85 bzw. COBOL85-R und als Produktversion 023 an.

#### 3.3.2 Installation zum Einsatz mit COMOPT-Steuerung

Der Compiler COBOL85 kann unter einer beliebigen Kennung installiert werden.

#### 3.3.3 Installation zum Einsatz mit SDF-Steuerung

Standardmäßig werden die Lieferbestandteile

- SYSPRG.COBOL85.023 (Compiler)
- SYSSDF.COBOL85.023.USER (User-Syntaxdatei)
- SYSSDF.COBOL85.023 (System-Syntaxdatei)
- SYSSPR.COBOL85.023.COMPILE (Compiler-Prozedur)

unter der Default-Userid installiert.

Das Einmischen der COBOL85-Syntaxdatei SYSSDF.COBOL85.023 in die System-Syntaxdatei sowie die Installation der Meldungsdateien erfolgt bei Standardinstallation mit SOLIS/IMON automatisch.

Verwendung der Prozedur SINPRC.COBOL85.023.DO-INST:

Der Name des Compilers lautet nicht mehr COBOL85, wie in früheren Versionen, sondern er wurde zu SYSPRG.COBOL85.023 modifiziert. Hierdurch ist die Installation mehrerer COBOL85-Compiler unter einer Kennung möglich. Falls andere Dateinamen bzw. eine andere Installationskennung gewünscht werden, können folgende Werte mit der Installationsprozedur SINPRC.COBOL85.023.DO-INST individuell angepasst werden:

- Name des Compilers
- Name der Syntaxdatei
- Benutzerkennung für Compiler und Syntaxdatei

Alle gewünschten Änderungen sind durch den einmaligen Aufruf der Installationsprozedur vorzunehmen. Für den ordnungsgemäßen Ablauf der Installationsprozedur ist die Datei SYSSDF.COBOL85.023.SYSTEM erforderlich. Die Datei kann bei Bedarf nach erfolgter Installation gelöscht werden.

Die Installationsprozedur kann auch von einer anderen Kennung aus gerufen werden, als derjenigen unter der die Produktdateien ins BS2000-System eingebracht wurden. Wurden die Produktdateien unter der Kennung \$<cobuid> eingebracht, so lautet der Aufruf der Prozedur:

```
/CALL-PROCEDURE $<cobuid>.SINPRC.COBOL85.023.DO-INST,-
/          PROC-PAR=(CURRENT-FILE-USERID=<cobuid>)
```

Achtung:

Diese Prozedur arbeitet standardmäßig mit dem Softwareprodukt SDF-A.

Unter nachfolgend beschriebener Voraussetzung kann statt SDF-A das kostenfreie SDF-U in der Installationsprozedur verwendet werden.

Es wird die SDF-U Version V4.0A mit Rep-Korrektur aus Korrekturpaket BS2GA.SDF V11.0B (SOLIS-Änderungsstand 04), bzw. BS2GA.SDF V2.0A (SOLIS-Änderungsstand 03) vorausgesetzt. Ist die passende SDF-U Version vorhanden, so ist die Installationsprozedur wie folgt aufzurufen:

```
/CALL-PROCEDURE SINPRC.COBOL85.023.DO-INST,-
/          PROC-PAR=(SDFA='START-SDF-U')
```

Nach Ablauf der Installationsprozedur kann das Kommando /START-COBOL85-COMPILER durch Zuweisung der COBOL85-User-Syntaxdatei (temporär) oder durch Einmischen der COBOL85-System-Syntaxdatei (dauerhaft) aktiviert werden.

Anmerkung:

Für COBOL85-BC ist in allen Dateinamen statt COBOL85 COBOL85-BC einzusetzen.

Analog ist für COBOL85-R in allen Dateinamen statt COBOL85 COBOL85-R einzusetzen.

### 3.3.4 Installation zum Einsatz im POSIX

Hinweis: Dieser Punkt gilt nicht für COBOL85-BC.

Zur Installation des Compilers im POSIX verwenden Sie bitte das POSIX-Installationstool. Beachten Sie bitte, dass dieses Tool die SINLIB-Datei auf der Default-User-Id erwartet. Haben Sie den Compiler nicht auf der Default-User-Id installiert, so müssen Sie, zum Einsatz in POSIX, die SINLIB-Datei dorthin kopieren. (Beschreibung siehe Handbuch "POSIX Grundlagen")

### 3.3.5 Installation für den Einsatz der Strukturierfunktionen

Hinweis: Dieser Punkt gilt nicht für COBOL85-BC.

Die folgenden Lieferbestandteile

- SYSPRG.COBOL85.023.IND (Strukturierer Beautify)
- SYSPRG.COBOL85.023.VERDI (Strukturierer Pretty-Print)
- SYSSPR.COBOL85.023.STRUCT (Strukturierer-Prozedur)

müssen unter der gleichen Kennung wie die unter 3.3.3 genannten Komponenten vorhanden sein.

Das Kommando /START-COBOL85-STRUCTURIZER wird zusammen mit dem Kommando /START-COBOL85-COMPILER durch Zuweisung der COBOL85-User-Syntaxdatei oder durch Einmischen der COBOL85-System-Syntaxdatei aktiviert.

### 3.4 Produkt-Einsatz

Das Compilersystem COBOL85 V2.3A ist ab BS2000/OSD V1.0 ablauffähig.

Die POSIX-Funktionalität des Compilers kann ab BS2000/OSD V2.0 genutzt werden.

### 3.5 Entfallene (und gekündigte) Funktionen

keine

### 3.6 Inkompatibilitäten

Die Compiler-Optionen USE-APOSTROPHE, RESET-PERFORM-EXITS und REPLACE-PSEUDOTEXT werden durch das Setzen einer der folgenden Compiler-Optionen FLAG-ABOVE-MINIMUM, FLAG-ABOVE-INTERMEDIATE, FLAG-ALL-SEGMENTATION, FLAG-INTRINSIC-FUNCTIONS, FLAG-NONSTANDARD, FLAG-OBSOLETE, FLAG-REPORT-WRITER oder FLAG-SEGMENTATION-ABOVE1 auf ihren Defaultwert zurückgesetzt. Dies wurde bisher durch eine Severity I Fehlermeldung angezeigt. Wegen gravierender semantischer Änderungen durch die Rücksetzung der Compiler-Optionen wird zur Sicherheit des Nutzers der Severity-Code von I auf 2 erhöht!

Werden mehrere Programme in einer Source gehalten (sequence of programs), wird bei der Übersetzung dieser Source nunmehr nur noch eine Optionenliste erstellt und nicht mehr wie früher eine Optionenliste für jedes Programm. Kommentarzeilen zwischen den einzelnen Programmen wurden früher ignoriert. Sie werden jetzt dem jeweils nachfolgenden Programm zugeordnet. Kommentarzeilen nach dem Ende des letzten Programms werden ignoriert.

COPY- und REPLACE-Anweisungen werden auch ausgeführt, wenn sie in Kommentareinträgen angegeben wurden. Kommentareinträge sind im COBOL85-Standard obsolet und entfallen im nächsten Standard. Durch diese Änderung eventuell entstehende Probleme lassen sich durch einen Stern in Spalte 7 der betreffenden Zeilen leicht umgehen.

In der Sourceliste wird im DATE-COMPILED Paragraph nicht mehr das Übersetzungsdatum ausgewiesen. Dieses steht bereits im Listenkopf.

### 3.7 Einschränkungen

Alias Catalog System (ACS):

Standard-Dateinamen (xxxLST.COB85.programmname) dürfen nicht als Alias-Dateinamen eingetragen werden, da ACS Aliasnamen mit und ohne vorangestellter Benutzerkennung nicht als gleich erkennt.

\*3 COBOL85-Strukturierer:

\*3

\*3 - Die Wirkung der Optionen LINE-SIZE und LOWER-CASE-KEYWORDS des  
\*3 LAYOUT-PARAMETERS beim PRETTY-PRINTER ist für vom Standard abwei-  
\*3 chende Werte undefiniert. Die Option BOLD-FACE=YES des LAYOUT-  
\*3 PARAMETERS wird für die Ausgabe auf SYSLIST nicht unterstützt

\*3

\*3 - Nach Kommentarzeilen werden von den Strukturierern in seltenen Fällen zusätz-  
\*3 liche Leerzeilen eingefügt. Die Verwendung des Zeichens X'15' führt bei den  
\*3 Strukturierern zum Abschneiden der betreffenden Zeile.

\*3

\*3 - Die Strukturierer unterstützen keine geschachtelten Programme

\*3

\*3 - Die Strukturierer unterstützen die Verarbeitung von Sourcen bis zu einer maxi-  
\*3 malen Größe von ca. 30000 Zeilen. Dabei sind die expandierten COPY-  
\*3 Elemente mitzuzählen.

\*3

\*3 - Liest der Beautifier eine Source von SYSDTA ein, wird bei Ausgabe auf STD-  
\*3 FILES statt dem Namen der Quellprogrammdatei SYSCMD zur Dateinamensbil-  
\*3 dung verwendet.

\*3

\*3 - Der Strukturierer löst bei Fehlern der Fehlerklasse 2 den SPINOFF-  
\*3 Mechanismus aus.

\*3

- Bei der "Pretty Print"-Funktion des Strukturierers werden die Anweisungen  
COPY textname REPLACING ==pseudotext== BY ==pseudotext==REPLACE  
==pseudotext== BY ==pseudotext== nur syntaktisch akzeptiert. Die beabsichtigte  
Textersetzung wird im Programmtext jedoch nicht dargestellt. Die nicht ersetzten  
Texte können Syntaxfehler verursachen. Wird bei einer COPY-Anweisung mit  
Pseudotextanweisung im kopierten Bereich ein Syntaxfehler gefunden, wird eine  
Warnung ausgegeben und in diesem kopierten Bereich kein weiterer Syntaxfehler  
mehr gemeldet. (Manualhinweis: Benutzerhandbuch Kapitel 5)

Wegen der systemspezifischen Unterschiede zwischen POSIX und BS2000 sind  
bei der Nutzung von COBOL-Programmen, die POSIX-Schnittstellen verwenden  
(ENABLE-UFS-ACCESS=YES), einige Besonderheiten hinsichtlich Sprachum-  
fang und Ablaufverhalten zu beachten. (Manualhinweis: Benutzerhandbuch Kapi-  
tel 13.5)

Bei Verwendung der Option ENABLE-UFS-ACCESS=YES wird der Gebrauch  
des in der ALPHABET-Klausel spezifizierten Zeichensatzes STANDARD-2 (In-  
ternational Reference Version of the ISO7-Bit Code) in der CODE-SET-Klausel  
nicht unterstützt. Ein derartiger OPEN wird zur Laufzeit mit FILE STATUS 30 ab-  
gewiesen.

Falls über die Option ENABLE-UFS-ACCESS der Zugriff auf das POSIX-  
Dateisystem ermöglicht wurde, werden in den Meldungen COB9151 und  
COB9175 bei Fehlern beim POSIX-Dateizugriff statt DMS-Codes die entspre-  
chenden SIS-Meldungsnummern eingesetzt.



Das gleiche gilt auch für den an das COBOL-Objekt ggf. zurückgegebenen extended File Status. Auch der zurückgegebene File Status kann vom bisher erwarteten Wert abweichen (Manualhinweis: Benutzerhandbuch Kapitel 13.6.3).

Falls die Option ENABLE-UFS-ACCESS=YES verwendet wird, erfordern die dann intern verwendeten Komponenten auf jeden Fall, dass das Programm unter ILCS läuft. Dies ist bei einem COBOL85-Hauptprogramm ab Version 1.1 gewährleistet. Anderssprachige Hauptprogramme müssten in diesem Fall für die entsprechende ILCS-Initialisierung sorgen.

Vorgebundene Großmodule bzw. Bindelademodule, die ältere COBOL85-Laufzeitmodule enthalten, werden durch das aktuelle Laufzeitsystem nicht unterstützt. Entsprechende Großmodule müssen neu gebunden werden. Dies bedeutet aber keine Einschränkung der Aufwärtskompatibilität zu alten COBOL85-Objektmodulen.

Im POSIX werden bei Indexed-Dateien Duplikate von Alternate Keys nicht korrekt erkannt (File Status 02 kommt nicht).

Im POSIX wird das im SINIX übliche LOCKING-Verhalten umgesetzt. So wird zum Beispiel das mehrfache Öffnen der gleichen Datei zur Ausgabe nicht unterbunden.

Simultanverarbeitung von Dateien (SHARED-UPDATE) wird nur bei Übersetzung mit COMOPT bzw. RUNTIME-OPTION ENABLE-UFS-ACCESS=NO unterstützt.

Für die Unterstützung von UDS/SQL V2.0A gilt folgende Einschränkung: Die auf FETCH1 und ACCEPT2 folgenden Anweisungen können, wenn

- das Eingabefeld für den FIND oder ACCEPT vom Typ DATABASE-KEY-LONG ist, und
- das Subschema im alten Format (UDS/SQL V1.2) vorliegt, nur dann richtig bearbeitet werden, wenn
- der Wert im Eingabefeld sich in einem Feld vom Typ DATABASE-KEY darstellen lässt.

Lässt sich nicht sicherstellen, dass der Wert im Eingabefeld diesen Voraussetzungen genügt, kann das Problem wie folgt umgangen werden:

- Subschema ohne die Option SUBSCHEMA FORM IS OLD mit dem DDL-Compiler übersetzen und
- anschließend das betreffende COBOL-Programm übersetzen.

\*2 Hinweis zu Dateien mit COBOL Dateiformat RELATIVE:  
 \*2  
 \*2 Relative Dateien können sowohl im DMS Dateiformat mit FCBTYPE=ISAM als  
 \*2 auch aus Kompatibilitätsgründen im DMS Dateiformat mit FCBTYPE=PAM ange-  
 \*2 legt sein. Im letzteren Fall ist der vom System zur Verfügung gestellte PAMKEY  
 \*2 für die korrekte Verarbeitung dieser relativen Datei unbedingt erforderlich!  
 \*2 Solche Dateien dürfen nicht auf NONKEY-Platten angelegt werden. Eine mecha-  
 \*2 nische Umsetzung mit Konvertierungsprogrammen, wie PAMCONV ist nicht  
 \*2 sinnvoll. Diese Tools kennen die COBOL85-spezifische Dateistruktur nicht und  
 \*2 zerstören sie. Der einzige Weg, solche relative Dateien auf NONKEY-Platten zu  
 \*2 bringen, besteht in einer Konvertierung mittels eines COBOL-Programms, das  
 \*2 die existierende relative Datei mit FCBTYPE=PAM liest und eine Kopie mit  
 \*2 FCBTYPE=ISAM erzeugt. Ein solches Programm ist im Abschnitt 5.1 bereitge-  
 \*2 stellt.  
 \*2  
 \*2 Eine Anpassung der die Datei verarbeitenden COBOL-Programme ist in der Re-  
 \*2 gel nicht erforderlich \*). Eine solche Konvertierung gibt Anlass, die besseren  
 \*2 Möglichkeiten bei relativen Dateien auf ISAM zu nutzen.

\*1 Der folgende Hinweis betrifft nur die RISC-Variante.  
 \*1  
 \*1 In Ausnahmefällen wird die RISC-Codegenerierung mit der folgenden Severity 3  
 \*1 Fehlermeldung abgebrochen  
 \*1 S.E. (FEHLER '!!!0000' BEI DER GENERIERUNG DES AUSGEWAELHTEN  
 \*1 DESTINATION CODE)  
 \*1 bzw.  
 \*1 S.E. (ERROR '!!!0000' DURING GENERATION OF SELECTED DESTINA-  
 \*1 TION CODE)  
 \*1 S.E. steht für System Error. Die RISC-Codegenerierung konnte nicht durchgeführt  
 \*1 werden. Bitte erzeugen Sie sich in diesem Fall ein /390-Objekt (siehe Abschnitt  
 \*1 2.5). Dieses kann mit den übrigen RISC-Objekten gebunden werden.

---

\*2 \*) Nur bei Satzgrößen nahe 2K kann es zu ungünstiger Blockung bzw. bei Angabe  
 \*2 einer BLOCK CONTAINS-Klausel unter Umständen zu Übersetzungsfehlern  
 \*2 kommen.

### 3.8 Verhalten im Fehlerfall

Im Fehlerfall sind die in der nachfolgenden Tabelle aufgelisteten Unterlagen an die betreffende regionale Dienststelle zu senden:

	Übersetzungs-Fehler	Ablaufzeit-Fehler
SOURCE-, DIAG- u. COMOPT-Liste	X	X
Ablaufprotokoll (MSG=FH)	X	X
Source (incl. COPY-Elemente und ggf. COSSD (COBOL-Subschema-Directory))	X	X
Binderliste		X
Ein-/Ausgabedateien		X
Erwartetes Ergebnis		X
Kurzbeschreibung des Ablaufs		X

Zu Diagnosezwecken ist eine genaue Beschreibung der Fehlersituation und Angabe, ob und wie der Fehler reproduzierbar ist, nötig.

## **4 Hardware-Anforderungen**

- \*5 COBOL85 V2.3A ist auf den Modellen mit /390-, RISC- und X86-System-
- \*5 Architektur (keine 25 Bit-Anlagen) ablauffähig, die von den in Abschnitt 3.2 ge-
- \*5 nannten Betriebssystemversionen unterstützt werden.

## \*2 5 Anhang

### \*2 5.1 Konvertierung relativer Dateien mit FCBTYPE=PAM

\*2 Wie im Abschnitt 3.7 bereits erläutert, müssen relative Dateien mit  
 \*2 FCBTYPE=PAM, bevor sie auf NONKEY-Platten gebracht werden, durch ein  
 \*2 COBOL-Programm in relative Dateien mit FCBTYPE=ISAM umgesetzt wer-  
 \*2 den. Eine Anpassung des die Datei verarbeitenden COBOL-Programms ist in  
 \*2 der Regel nicht erforderlich.

\*2 Das folgende Programm demonstriert, wie eine solche Umsetzung realisiert wer-  
 \*2 den könnte.

```
*2
*2 IDENTIFICATION DIVISION.
*2 PROGRAM-ID. RELCONV.
*2
*2 ENVIRONMENT DIVISION.
*2 CONFIGURATION SECTION.
*2 INPUT-OUTPUT SECTION.
*2 FILE-CONTROL.
*2
*2 * Klauseln der existierenden Datei
*2 * ORGANIZATION RELATIVE
*2 * ACCESS SEQUENTIAL
*2 * RELATIVE KEY ...
*2 SELECT RELPAM
*2 ORGANIZATION IS RELATIVE
*2 ACCESS IS SEQUENTIAL
*2 RELATIVE KEY IS REL-KEY
*2 ASSIGN TO "INFILE".
*2
*2 * Klauseln der zu erzeugenden Datei
*2 * ORGANIZATION RELATIVE
*2 * ACCESS RANDOM
*2 * RELATIVE KEY ...
*2 SELECT RELISAM
*2 ORGANIZATION IS RELATIVE
*2 ACCESS IS RANDOM
*2 RELATIVE KEY IS REL-KEY
*2 ASSIGN TO "OUTFILE".
*2
*2 DATA DIVISION.
*2 FILE SECTION.
*2
*2 * Klauseln und Satzbeschreibung
*2 * der existierenden Datei
*2 FD RELPAM.
*2 01 REC-PAM.
*2 ...
*2
*2 * Klauseln und Satzbeschreibung
*2 * der zu erzeugenden Datei
*2 FD RELISAM.
*2 01 REC-ISAM.
*2 ...
```

```
*2          WORKING-STORAGE SECTION.
*2          01 REL-KEY          ...
*2          01 IO-STATUS        PIC X(3) VALUE SPACES.
*2          88 EOF              VALUE "EOF".
*2          88 NOT-EOF          VALUE SPACES.
*2
*2          PROCEDURE DIVISION.
*2              OPEN INPUT RELPAM, OUTPUT RELISAM
*2              READ RELPAM
*2              AT END SET EOF TO TRUE
*2              END-READ
*2              PERFORM UNTIL EOF
*2                  WRITE REC-ISAM FROM REC-PAM
*2          *  INVALID KEY sollte nicht passieren!
*2              INVALID KEY ...
*2              END-WRITE
*2              READ RELPAM
*2              AT END SET EOF TO TRUE
*2              END-READ
*2              END-PERFORM
*2              CLOSE RELPAM, RELISAM
*2              STOP RUN.
```