

FUJITSU Software BS2000 COBOL2000-BC und  
FUJITSU Software BS2000 COBOL2000

Version V1.6A  
Juni 2018

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.

© 2018 Fujitsu Technology Solutions GmbH

Die Marke Fujitsu und das Fujitsu Logo sind Marken oder registrierte Marken von Fujitsu Limited in Japan und in anderen Ländern. BS2000 ist eine Marke von Fujitsu Technology Solutions GmbH in Deutschland.

<b>1 Allgemeines</b>	<b>3</b>
1.1 Bestellung	4
1.2 Auslieferung	4
1.2.1 Lieferbestandteile der Basisvariante	4
1.2.2 Lieferbestandteile der Vollvariante	4
1.3 Dokumentation	5
<b>2 Software-Erweiterungen</b>	<b>6</b>
2.1 XML-Unterstützung	6
2.1.1 Anweisung XML GENERATE	6
2.1.2 SOURCE-PROPERTIES Option KEEP-XML-NAMES	6
<b>3 Technische Hinweise</b>	<b>7</b>
3.1 Ressourcenbedarf	7
3.2 SW-Konfiguration	7
3.3 Produkt-Installation	8
3.3.1 Installation zum Einsatz im POSIX	8
3.4 Produkt-Einsatz	9
3.4.1 Produkteinsatz mit dem Precompiler ESQL-COBOL	9
3.4.2 Produkteinsatz mit UTM	9
3.4.3 Produkteinsatz mit COLUMBUS85	10
3.4.4 Produkteinsatz mit AID	10
3.5 Entfallene (und gekündigte) Funktionen	11
3.6 Inkompatibilitäten	11
3.7 Einschränkungen	12
3.8 Verhalten im Fehlerfall	12
<b>4 Hardware-Anforderungen</b>	<b>13</b>
<b>5 Firmware-Stände</b>	<b>14</b>

# 1 Allgemeines

COBOL2000, der COBOL-Compiler im BS2000 für objektorientiertes Programmieren, ermöglicht Programmierung nach dem COBOL-Standard ANSI X3.23-1985 (= ISO/IEC 1989-1985 = EN 21989 = DIN 66028-1986).

COBOL2000 V1.6A unterstützt den Zusatz zur Norm ANSI X3.23a-1989 bzw. ISO/IEC 1989 Amendment1, Intrinsic Function Module ebenso wie 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.

Ferner enthält COBOL2000 die "Data Manipulation Language DML" zur Bedienung des Datenbanksystems UDS.

COBOL2000 V1.6A unterstützt neben dem COBOL85-Standard die wesentlichen objektorientierten Funktionalitäten und Basisfunktionen aus dem COBOL2002-Standard (= ISO/IEC 1989:2002(E) Programming Language COBOL).

Neben den Spracherweiterungen werden in COBOL2000 V1.6A zusätzlich POSIX- und XPG4-Programmierschnittstellen unterstützt.

Der Compiler COBOL2000 V1.6A wird ohne Laufzeitsystem ausgeliefert. Das gemeinsame Laufzeitsystem für COBOL85, COBOL2000, C und C++ im BS2000 ist das Common Runtime Environment CRTE.

COBOL2000 V1.6A wird in zwei Varianten angeboten:

- Basisvariante (BC - Basic Configuration)
- Vollvariante

In der Basisvariante COBOL2000-BC V1.6A werden folgende Funktionen nicht unterstützt:

- REPORT WRITER
- CODASYL-DML (native UDS-Unterstützung)
- Symbolisches Testen mit AID
- Ausgabe einer Liste aller Fehlermeldungen
- POSIX-Schnittstellen
- Starterphase

Diese Freigabemitteilung enthält in gedrängter Form die wesentlichen Erweiterungen, Abhängigkeiten und Betriebshinweise zu COBOL2000 V1.6A unter dem Betriebssystem BS2000.

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

Der Inhalt entspricht dem Freigabestand: Juni 2018.

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

Werden mit dem Einsatz der vorliegenden Produktversion eine oder mehrere Vorgängerversionen übersprungen, so sind auch die Hinweise aus den Freigabemitteilungen (bzw. README-Dateien) der Vorgängerversionen zu berücksichtigen.

## 1.1 Bestellung

COBOL2000 V1.6A kann über Ihre zuständige Vertriebsgesellschaft bezogen werden. Für COBOL2000 V1.6A gelten die allgemeinen Bedingungen zum Vertrag über die Nutzung und Betreuung von Software-Produkten.

## 1.2 Auslieferung

Die Lieferung der Dateien zu COBOL2000 V1.6A erfolgt mit dem Lieferverfahren SOLIS.

### 1.2.1 Lieferbestandteile der Basisvariante

Folgende Lieferbestandteile werden unabhängig vom Hardware-Typ (HSI) benötigt:

SYSFGM.COBOL2000-GEM.016.D	Freigabemitteilung (deutsch)
SYSFGM.COBOL2000-GEM.016.E	Freigabemitteilung (englisch)
SYSTEMS.COBOL2000-BC.016	Meldungsdatei (MSGMAKER-Format)
SYSPRG.COBOL2000-BC.016	Compiler
SYSSDF.COBOL2000-BC.016	System-Syntaxdatei
SYSSDF.COBOL2000-BC.016.USER	User-Syntaxdatei
SYSSII.COBOL2000-BC.016	IMON Informationsdatei
SYSSII.COBOL2000-GEM.016	IMON Informationsdatei für gemeinsame Lieferbestandteile
SYSSPR.COBOL2000-BC.016.COMPIL	SDF-Startprozedur für Compiler

### 1.2.2 Lieferbestandteile der Vollvariante

Folgende Lieferbestandteile werden unabhängig vom Hardware-Typ (HSI) benötigt:

SINLIB.COBOL2000.016	Bibliothek zur Installation im POSIX
SYSFGM.COBOL2000-GEM.016.D	Freigabemitteilung (deutsch)
SYSFGM.COBOL2000-GEM.016.E	Freigabemitteilung (englisch)
SYSTEMS.COBOL2000.016	Meldungsdatei (MSGMAKER-Format)
SYSPRG.COBOL2000.016	Compiler
SYSPRG.COBOL2000.016.START	Starterphase, wird unter dem Dateinamen COBOL2000 installiert
SYSSDF.COBOL2000.016	System-Syntaxdatei
SYSSDF.COBOL2000.016.USER	User-Syntaxdatei
SYSSII.COBOL2000.016	IMON Informationsdatei

SYSSII.COBOL2000-GEM.016	IMON Informationsdatei für gemeinsame Lieferbestandteile
SYSSPR.COBOL2000.016.COMPIL	SDF-Startprozedur für Compiler

Im SOLIS2-Lieferanschreiben sind die einzelnen Dateien mit den jeweils gültigen Datei- und Datenträger-Merkmalen aufgeführt.

### 1.3 Dokumentation

Folgende Dokumentation ist für COBOL2000 V1.6 verfügbar:

Deutsch:	Englisch:
COBOL2000 V1.6 COBOL Compiler Sprachbeschreibung	COBOL2000 V1.6 COBOL Compiler Reference Manual
COBOL2000 V1.6 COBOL-Compiler Benutzerhandbuch	COBOL2000 V1.6 COBOL Compiler User Guide

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

Die Dokumentation ist auch im Internet unter <http://manuals.ts.fujitsu.com> verfügbar. Handbücher, die mit einer Bestellnummer angezeigt werden, können auch in gedruckter Form bestellt werden.

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

SYSRME.<produkt>.<version>.D (Datei mit deutschem Text)  
SYSRME.<produkt>.<version>.E (Datei mit englischem Text)

Die README-Dateien sind auf der SoftBooks-DVD enthalten bzw. online unter <http://manuals.ts.fujitsu.com> verfügbar.

## 2 Software-Erweiterungen

Im Folgenden werden nur die Erweiterungen bzw. Verbesserungen gegenüber der Vorgängerversion COBOL2000 V1.5 beschrieben.

### 2.1 XML-Unterstützung

Ab COBOL2000 V1.5A werden das Lesen und die Analyse von XML-Dokumenten unterstützt.

Ab COBOL2000 V1.6A können XML-Daten generiert werden.

#### 2.1.1 Anweisung XML GENERATE

Ab COBOL2000 V1.6A können Daten in XML-Format erzeugt werden. Dies geschieht mit der Anweisung XML GENERATE.

Eine Beschreibung der Anweisung finden Sie in den COBOL2000-Handbüchern.

#### 2.1.2 SOURCE-PROPERTIES Option KEEP-XML-NAMES

Mit dieser Option wird festgelegt, ob beim Erzeugen von XML-Daten die XML-Elementnamen in Großbuchstaben umgewandelt werden oder die Originalnamen beibehalten werden.

Die Option wurde realisiert als COMOPT-Option:

```
{COMOPT | COBRUN} KEEP-XML-NAMES={YES | NO}
```

und als SDF-Option:

```
SOURCE-PROPERTIES = *PARAMETERS(...)  
XML-NAMES = {* KEEP | *UPPER}
```

Eine Beschreibung der Option finden Sie in den COBOL2000-Handbüchern.

## 3 Technische Hinweise

### 3.1 Ressourcenbedarf

Die mit dem Produkt ausgelieferten Dateien belegen

- ca. 2000 PAM-Seiten für die Basisvariante
- ca. 5600 PAM-Seiten für die Vollvariante

Wird bei der Vollvariante das POSIX-Dateisystem nicht genutzt, kann die Bibliothek SINLIB.COBOL2000.016 gelöscht werden. Dabei wird das Software Configuration Inventory (SCI) nicht angepasst.

Bei der Übersetzung sollten dem Compiler ca. 50 MB virtueller Adressraum zur Verfügung stehen. Dieser Wert stellt einen Minimalbedarf dar, der sich je nach Datenmenge und Anwendung vergrößern kann.

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

Für COBOL2000 V1.6A wird BS2000 ab Version 10.0 benötigt, d.h. OSD/BC ab V10.0, bzw. OSD/XC ab V10.0.

Für S-Server benötigen Sie die folgenden Lieferbestandteile:

- BS2GA.LLMAM ab V10.0
- BS2GA.SDF ab V10.0
- BS2GA.CRTE-BAS ab V10.0

Wenn kein expliziter SOLIS-Änderungsstand spezifiziert wurde, werden die Korrekturstände des Korrekturpaketes 1/2018 vorausgesetzt.

COBOL2000 V1.6A setzt die Installation von "Common Runtime Environment" CRTE ab V10.0B bzw. ab V11.0B voraus. CRTE ist eine eigene Liefereinheit und muss für S-Server getrennt bestellt werden.

Für SQ-Server und x86-basierten Server Units der BS2000 SE Serie benötigen Sie die folgenden Lieferbestandteile:

- BS2GA.BS2OSD ab V10.0
- BS2GA.CRTE-BAS ab V10.0

Wenn kein expliziter SOLIS-Änderungsstand spezifiziert wurde, werden die Korrekturstände des Korrekturpaketes 1/2018 vorausgesetzt.

Für die Nutzung des POSIX-Dateisystems durch den Compiler muss das POSIX-Subsystem geladen sein.

Die Unterstützung von Unicode (NATIONAL) setzt openNet Server ab V3.6 voraus.

Für die Übersetzung von COBOL-Programmen werden folgende Produkte vorausgesetzt:

- CRTE ab V10.0B bzw. ab V11.0B
- JV (1) ab V15.1
- UDS/SQL (1) ab V2.7
- XHCS-SYS (2) ab V2.2

- (1) nur, wenn entsprechende COBOL-Sprachmittel/Systemdienste verwendet werden
- (2) nur, wenn der Datentyp NATIONAL bzw. XML-Sprachmittel verwendet werden

Zum Ablauf von COBOL2000-Anwendungen werden folgende Produkte vorausgesetzt:

- AID (1) ab V3.4
- CRTE ab V10.0B bzw. V11.0B
- LMS (2) ab V3.4
- SORT (1) ab V7.9C
- UDS/SQL (1) ab V2.7
- UTM (3) ab V6.3
- XHCS-SYS (4) ab V2.2

- (1) nur, wenn entsprechende COBOL-Sprachmittel/Systemdienste verwendet werden
- (2) nur wenn LINE-SEQUENTIAL I/O verwendet wird
- (3) nur wenn COBOL-Module als UTM-Teilprogramme ablaufen sollen
- (4) nur, wenn der Datentyp NATIONAL bzw. XML-Sprachmittel verwendet werden

Für den gleichzeitigen, parallelen Betrieb mehrerer Versionen des COBOL2000-Compilers wird COBOL2000 ab V1.0B vorausgesetzt.

### 3.3 Produkt-Installation

Die Installation des Produktes COBOL2000 muss mit dem Installationsmonitor IMON durchgeführt werden. Für die Installation des Produktes müssen neben den Hinweisen in der vorliegenden Freigabemitteilung auch die Informationen zur Installation im Lieferanschreiben berücksichtigt werden.

Die erforderlichen Eingaben sowie der Ablauf der Installation sind im Handbuch zu IMON beschrieben.

#### 3.3.1 Installation zum Einsatz im POSIX

Hinweis: Dieser Punkt gilt nicht für COBOL2000-BC.

Für die Installation von COBOL2000 in POSIX ist wie folgt zu verfahren:

- Falls in POSIX eine Version des Produktes COBOL85 installiert ist, muss diese deinstalliert werden. Hierzu rufen Sie unter TSOS das POSIX-Installationsprogramm (Beschreibung siehe Handbuch „POSIX Grundlagen“) auf und wählen den Punkt „Delete packages from POSIX“. Hier wählen Sie das Produkt COBOL85 aus.
- Auch eine bereits im POSIX installierte Vorgängerversion des Produktes COBOL2000 muss zuvor deinstalliert werden. Hierzu ist analog vorzugehen.

- Die Installation von COBOL2000 im POSIX-Dateisystem erfolgt nun wahlweise
  - direkt mit IMON (POSIX-Installation muss angewählt werden) oder alternativ
  - nach erfolgreicher IMON-Installation (ohne POSIX-Installation) mit dem POSIX-Installationstool. Hierzu wählen Sie nach Start des POSIX-Installationstools „Install packages on POSIX“. Im Weiteren geben Sie als Produkt COBOL2000 und als Produktversion 016 an.

### 3.4 Produkt-Einsatz

#### 3.4.1 Produkteinsatz mit dem Precompiler ESQL-COBOL

Neben den Hinweisen in der Freigabemitteilung zum Produkt ESQL-COBOL sind bei der Nutzung des Precompilers ESQL-COBOL für SESAM nachfolgende Punkte zu beachten:

- ESQL-COBOL ab V3.0B unterstützt ebenfalls Quellprogramme im COBOL Free Reference Format.
- Da der ESQL-Precompiler vor dem COBOL-Compiler läuft, wirken sich die Sprachmittel zur bedingten Kompilierung ( z.B. ">> EVALUATE ..." oder ">> IF ..." ) nicht auf die von diesen Anweisungen eingeklammerte SQL-Anweisungen aus, da der Precompiler diese Anweisungen unabhängig vom lokalen Status der bedingten Kompilierung immer verarbeitet. Diese unterschiedliche Interpretation des Programmtextes durch Precompiler und Compiler führt nicht unbedingt zu Fehlermeldungen. Deshalb sollten SQL-Anweisungen nicht mit bedingter Kompilierung eingeklammert werden.
- Syntaktische Einschränkungen bezüglich der Mischung von ESQL und neuen Programmstrukturen zur Objekt Orientierung mit rekursiven Programmen und zugehörigen lokalen Datenbereichen sind nicht bekannt. Um Probleme durch unterschiedliche Sichtbarkeit von Daten und begrenzte Lebensdauer der Objekte sicher zu vermeiden, wird empfohlen SQL-Declare Sections nicht in der Local-Storage Section zu definieren und auch SQL-Anweisungen nur in klassischen COBOL-Programmen, nicht aber in Klassen und Methoden anzuwenden.

Der ESQL-Precompiler betrachtet eine Übersetzungsgruppe als ein Quellprogramm. Die Nutzung von SQL-Sprachmitteln in mehreren Übersetzungseinheiten führt zu Problemen mit der Sichtbarkeit der referenzierten Daten.

#### 3.4.2 Produkteinsatz mit UTM

Neben den Hinweisen in der Freigabemitteilung zum Produkt UTM sind folgende Punkte zu beachten:

- Die mit dem COBOL2000-Compiler erzeugten Teilprogramme können, ebenso wie die mit dem COBOL85-Compiler erzeugten Teilprogramme, bei UTM mit COMP=ILCS oder COMP=COB1 generiert werden. COMP=COB1 muss verwendet werden, wenn von dem COBOL2000-Modul Unterprogramme aufgerufen werden, die nicht ILCS-fähig sind. Ist dies nicht der Fall, sollte COMP=ILCS verwendet werden.
- Die mit dem COBOL2000-Compiler übersetzten Programme können keine TCB-Entries verwenden, unabhängig davon, ob sie bei UTM mit COMP=COB1 oder COMP=ILCS generiert werden.

- Die Lebensdauer der Instanzobjekte ist auf einen Teilprogrammlauf begrenzt. Beim PEND-Aufruf werden alle Objekte, die innerhalb eines Teilprogramm-laufs neu angelegt wurden, vom Laufzeitsystem freigegeben. Das gilt auch bei PEND-Varianten ohne Prozesswechsel. Deshalb dürfen Objekt-Referenzen nicht über einen Teilprogrammlauf hinaus aufbewahrt werden (in UTM-Speicherbereichen, im Shared Memory oder in einer Datenbank), um sie an nachfolgende Teilprogramme zu übergeben. Objekt-Referenzen sind nur in dem Teilprogrammlauf, der das Objekt angelegt hat, gültig. Es ist nicht möglich, Objekt-Referenzen über Taskgrenzen hinweg in einem anderen Programm zu verwenden.
- Der Programmaustausch von LLM's, die COBOL2000-Module mit Klassende-finitionen enthalten, setzt voraus, dass diese LLM's gemeinsam mit allen LLM's ersetzt werden, die die entsprechenden Klassendefinitionen erben o-der nutzen.

### 3.4.3 Produkteinsatz mit COLUMBUS85

Neben den Hinweisen in der Freigabemitteilung zum Produkt COLUMBUS85 sind folgende Punkte zu beachten:

- COLUMBUS85 besitzt keine direkte Schnittstelle zum COBOL-Compiler. Die mit COLUMBUS85-Werkzeugen bearbeiteten COBOL-Programme können grundsätzlich weiterhin mit dem COBOL2000-Compiler übersetzt werden. COLUMBUS85 unterstützt, wie der COBOL85-Compiler, lediglich den COBOL85-Sprachstandard.
- Es ist daher nicht möglich, COBOL-Programme unter Ausnutzung der neuen Sprachmittel des COBOL2000-Compilers weiter zu entwickeln und dabei für diese Programme die COLUMBUS85-Werkzeuge einzusetzen. Die COLUMBUS85-Werkzeuge reagieren mit Fehlermeldungen oder Fehlinterpretationen, sofern in den COBOL-Programmen Sprachmittel verwendet werden, die nicht mit dem COBOL85-Compiler übersetzt werden können. Dies sind unter anderem:
  - die neuen COBOL2000-Übersetzungseinheiten (z.B. Klassendefinitionen, Programm-Prototypen)
  - Verstöße gegen die Regeln bezüglich des Source-Formates (Aufteilung in A-Bereich und B-Bereich oder Free Format)
  - alle Direktiven
  - die neuen COBOL2000-Schlüsselwörter

### 3.4.4 Produkteinsatz mit AID

Beim Testen mit AID sollten zu Beginn einer Testsitzung die folgenden Punkte beachtet werden:

Unmittelbar nach dem Laden steht ein geladenes COBOL-Programm vor der ersten Anweisung in der PROCEDURE DIVISION und es wurden noch keine Initialisierungen durchgeführt. Deshalb können einzelne Kommandos zu einer Fehlermeldung führen oder es kann die Angabe von zusätzlichen Qualifikationen erforderlich sein.

Insbesondere bei objektorientierter Programmierung ist die Initialisierung der Klassen und Objekte noch nicht durchgeführt. AID kann daher Daten und Testpunkte in Objekten und Methoden sowie Objektreferenzen in den aufrufenden Programmen nicht in allen Fällen ordnungsgemäß behandeln. Dieses Problem lässt sich vermeiden, wenn vor anderen AID-Kommandos mit

```
/%TRACE 1 %STMT
```

die AID-Ablaufverfolgung eingeschaltet und das Programm gestartet wird. Hierbei wird die Initialisierung durchlaufen und das Programm fährt mit der ersten Anweisung fort.

Bei der Anzeige von BASED-Daten in rekursiven Programmen mit:

`/%SD`

wird stets der Inhalt der Daten aus der innersten, und damit der aktuellen Rekursion angezeigt. Der Inhalt äußerer Rekursionen muss anderweitig ermittelt werden, z.B. durch Anhalten des Programms an geeigneter Stelle und Ausgabe der gewünschten Daten.

### 3.5 Entfallene (und gekündigte) Funktionen

Zwischen COBOL2000 V1.5 und COBOL2000 V1.6A wurden keine Funktionen gestrichen.

Die folgenden Flagging-Optionen werden in dieser Version letztmalig unterstützt:

- FLAG-ABOVE-MINIMUM
- FLAG-ABOVE-INTERMEDIATE
- FLAG-ALL-SEGMENTATION
- FLAG-INTRINSIC-FUNCTIONS
- FLAG-REPORT-WRITER
- FLAG-SEGMENTATION-ABOVE1

Dies gilt gleichermaßen für die dazugehörigen ACTIVATE-FLAGGING-Optionen.

### 3.6 Inkompatibilitäten

COBOL2000 V1.6A ist voll kompatibel zur der Vorversion COBOL2000 V1.5.

COBOL2000 V1.5 ist bis auf die folgenden Einschränkungen voll kompatibel zur Vorversion COBOL2000 V1.4.

Inkompatibilitäten des Sprachumfangs:

- Folgende neue Schlüsselwörter sind reserviert, falls XML verwendet wird (COBRUN ENABLE-XML-PROCESSING=YES):  
DOCUMENT, END-OPEN, END-XML, IDENTIFIED, VERSION-XML, XML
- Folgende neue Sonderregister sind reserviert, falls XML verwendet wird:  
XML-EVENT, XML-CODE, XML-TEXT, XML-NTEXT, XML-NAMESPACE,  
XML-NNAMESPACE, XML-NAMESPACE-PREFIX, XML-NNAMESPACE-PREFIX

Hat der Anwender solche Datennamen definiert, dann gehen seine Definitionen kommentarlos vor.

- ASSIGN TO datenname  
Diese Klausel ist in Zukunft standardkonform als  
"ASSIGN USING datenname" zu schreiben.

Inkompatibilitäten des Laufzeitsystems:

- wenn USAGE NATIONAL Daten verwendet werden, muss der XHCS-Adaptermodul GNLADPT eingebunden bzw. resolved werden. Ausnahme: Verwendung von partial-bind (hier ist der Adapter schon eingebunden).

### 3.7 Einschränkungen

Bei Einsatz von COBOL2000 V1.6A sind folgende Einschränkungen zu beachten:

- Alias Catalog System (ACS):  
Standard-Dateinamen (xxxLST/FIL.COBOL.<prog-id>) dürfen nicht als Alias-Dateinamen eingetragen werden, da ACS Aliasnamen mit und ohne vorangestellter Benutzerkennung nicht als gleich erkennt.
- 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.

Hinweis: Anwendungen dürfen stets nur mit einem einzigen CRTE gebunden werden.

- Bei der Garbage Collection für OO-Programme kommt es in seltenen Fällen dazu, dass auch die Routinen der Garbage Collection keinen Speicher mehr vom System erhalten. Das Programm wird dann mit einer entsprechenden Meldung abgebrochen.

### 3.8 Verhalten im Fehlerfall

Im Fehlerfall werden zu Diagnosezwecken folgende Fehlerunterlagen benötigt:

- genaue Beschreibung der Fehlersituation
- die Angabe, ob und wie der Fehler reproduzierbar ist
- Options-, Source-, Fehlerliste einschließlich Expansion der COPY-Elemente (LISTING-Option)
- Ablaufprotokoll (MSG=FH)
- Source einschließlich der COPY-Elemente und gegebenenfalls COSSD (COBOL-Subschema-Directory)
- Binderliste
- Ein-/Ausgabedateien
- erwartetes Ergebnis
- Kurzbeschreibung des Ablaufs
- Versionsnummer des Produkts
- verwendete Rep-Dateien
- DUMP, falls vorhanden
- Angabe der verwendeten Subsysteme

Die Störung ist an den zuständigen Service Provider zu melden. Beim Second Level Support wird eine Störmeldung eröffnet.

Für Diagnosezwecke ist der Anschluss von Teleservice mit Rückrufmöglichkeit unbedingt erforderlich! Falls kein Teleservice betrieben wird, kann der Service Provider zusätzliche Aufwände gesondert verrechnen.

## **4 Hardware-Anforderungen**

COBOL2000 V1.6A ist auf allen

- Business Servern der S-Linie (/390-Architektur)
- Business Servern der SQ-Linie (x86-Architektur)
- Business Servern der SE-Linie (/390- und x86-Architektur)

ablauffähig, die von den in Abschnitt 3.2 genannten Betriebssystemversionen unterstützt werden.

## **5 Firmware-Stände**

nicht relevant