

FUJITSU Software BS2000 C/C++-RS

*6
*12

Version 3.2E
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 Europa und in anderen Ländern.

1 Allgemeines	4
1.1 Bestellung	5
1.2 Auslieferung	5
1.3 Dokumentation	5
2 Software-Erweiterungen	8
2.1 Korrekturen bekannter Fehler	8
2.2 Steuerbare Behandlung doppelter Entries im POSIX	8
2.3 Erzeugung von X86-Code	8
2.4 Initialisierung	8
2.5 Listengenerierung	8
2.6 Switch-Optimierung	8
3 Technische Hinweise	9
3.1 Ressourcenbedarf	9
3.2 SW-Konfiguration	9
3.3 Produkt-Installation	10
3.3.1 Installation und Deinstallation	10
3.3.1.1 Öffentliche Installation (SOLIS) für BS2000	11
3.3.1.1.1 Durchführung der Installation	11
3.3.1.1.2 Vorladbare Subsysteme	11
3.3.1.1.3 Benutzung	11
3.3.1.1.4 Deinstallation	11
3.3.1.2 Automatische öffentliche Installation (SOLIS) für POSIX	12
3.3.1.2.1 Durchführung der Installation	12
3.3.1.2.2 Vorladbares Subsystem	12
3.3.1.2.3 Benutzung	12
3.3.1.2.4 Deinstallation	12
3.3.1.3 Manuelle öffentliche Installation für POSIX mit IMON	12
3.3.1.3.1 Voraussetzungen	13
3.3.1.3.2 Durchführung der Installation	13
3.3.1.3.3 Fehler bei der Installation	13
3.3.1.3.4 Installationsverzeichnis	14
3.3.1.3.5 Vorladbares Subsystem	14
3.3.1.3.6 Benutzung	14
3.3.1.3.7 Deinstallation	14
3.3.1.4 Manuelle öffentliche Installation für POSIX ohne IMON	15
3.3.1.4.1 Voraussetzungen	15
3.3.1.4.2 Durchführung der Installation	15
3.3.1.4.3 Fehler bei der Installation	15
3.3.1.4.4 Installationsverzeichnis	16
3.3.1.4.5 Vorladbares Subsystem	16
3.3.1.4.6 Benutzung	16
3.3.1.4.7 Deinstallation	16
3.3.1.5 Private Installation für BS2000	16
3.3.1.5.1 Voraussetzungen	16
3.3.1.5.2 Durchführung der Installation	17
3.3.1.5.3 Fehler bei der Installation	17
3.3.1.5.4 Benutzung	17
3.3.1.5.5 Deinstallation	18
3.3.1.6 Private Installation für POSIX	18
3.3.1.6.1 Voraussetzungen	18
3.3.1.6.2 Durchführung der Installation	19
3.3.1.6.3 Fehler bei der Installation	19
3.3.1.6.4 Vorladbares Subsystem	19
3.3.1.6.5 Benutzung	19
3.3.1.6.6 Deinstallation	19
3.3.1.7 Installation II-Update-Tool	20
3.3.1.7.1 Voraussetzungen	20
3.3.1.7.2 Durchführung der Modifikation	20
3.3.1.7.3 Fehler bei der Modifikation	21
3.3.1.7.4 Benutzung	21

3.3.2	Benutzung anderer Produkte	21
3.4	Produkt-Einsatz	21
3.4.1	C++-Source- und Objekt-Kompatibilität	21
3.4.2	Native Code	21
3.5	Entfallene (und gekündigte) Funktionen	22
3.5.1	Entfallene Funktionen	22
3.5.1.1	RISC(MIPS)-Code-Erzeugung	22
3.5.2	Gekündigte Funktionen	23
3.5.2.1	POSIX-Optionen	23
3.6	Inkompatibilitäten	24
3.6.1	LL-Bibliotheken	24
3.6.2	Dateinamen im II-File	24
3.6.3	Kombination ANSI-C++ und ASCII- bzw. IEEE-Option	24
3.6.4	Wichtiger Hinweis zu CRTE	24
3.7	Einschränkungen	25
3.7.1	Einschränkungen bei der Native-Code Erzeugung	25
3.7.2	Maximale Namenslänge	25
3.7.3	Nicht unterstützte LISTING-Option COMMENTS	25
3.7.4	AID-Fehler bei leeren Konstruktoren	25
3.7.5	Besonderheiten im Cfront-C++-Sprachmodus	25
3.7.6	CFE1079 bei Verwendung der Optionen IEEE bzw. ASCII	26
3.8	Verhalten im Fehlerfall	26
4	Hardware-Unterstützung	27

1 Allgemeines

C/C++-RS V3.2 ist die Nachfolgeversion von C/C++-RS V3.1.

- *11 C/C++-RS ist nur für interne Nutzer verfügbar; die allgemein freigegebene Produktvariante ist C/C++ V3.2.
- Der Name der Liefergruppe und damit Namensbestandteil der Produktdateien ist CPP-RS.
- C/C++ ist der strategische BS2000/OSD-Compiler zur Entwicklung und zur Portierung von Anwendungen aus der offenen Welt (z.B. OO-Anwendungen) auf BS2000 Business Server.
- *6 Diese Freigabemitteilung enthält in gedrängter Form die wesentlichen Erweiterungen, Abhängigkeiten und Betriebshinweise zu C/C++-RS V3.2E unter dem Betriebssystem BS2000/OSD.
- *12 Der Inhalt entspricht dem Freigabestand: Juni 2018.
- *12 Änderungen gegenüber Freigabestand Juni 2017 sind mit „*12“ gekennzeichnet.
- *11 Änderungen gegenüber Freigabestand Juni 2016 sind mit „*11“ gekennzeichnet.
- *10 Änderungen gegenüber Freigabestand November 2015 sind mit „*10“ gekennzeichnet.
*10
- *9 Änderungen gegenüber Freigabestand Mai 2015 sind mit „*9“ gekennzeichnet.
- *8 Änderungen gegenüber Freigabestand Dezember 2014 sind mit „*8“ gekennzeichnet.
*8
- *7 Änderungen gegenüber Freigabestand Juni 2014 sind mit „*7“ gekennzeichnet.
- *6 Änderungen gegenüber Freigabestand Juni 2013 sind mit „*6“ gekennzeichnet.
- *5 Änderungen gegenüber Freigabestand Juni 2012 sind mit „*5“ gekennzeichnet.
- *4 Änderungen gegenüber Freigabestand Dezember 2011 sind mit „*4“ gekennzeichnet.
*4
- *3 Änderungen gegenüber Freigabestand Dezember 2010 sind mit „*3“ gekennzeichnet.
*3
- *2 Änderungen gegenüber Freigabestand April 2009 sind mit „*2“ gekennzeichnet.
*2
- *2 Änderungen gegenüber Freigabestand November 2007 sind mit „*1“ gekennzeichnet.
*2
- *3 Diese und andere aktuelle Freigabemitteilungen sind auf der SoftBooks-DVD enthalten und online verfügbar unter <http://manuals.ts.fujitsu.com/>.
*3
- 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

*11 C/C++-RS V3.2 steht nur internen Nutzern zur Verfügung.

1.2 Auslieferung

Die Lieferung der Dateien zu C/C++-RS V3.2 erfolgt mit dem Lieferverfahren SOLIS.

Folgende Lieferbestandteile werden auf jeder Hardware benötigt:

SINLIB.CPP-RS.032	Compiler-Bibliothek (POSIX)
SINPRC.CPP-RS.032	Bibliothek mit Privatinstallation-Prozeduren
SYSGM.CPP-RS.032.D	Freigabemitteilung (deutsch)
SYSGM.CPP-RS.032.E	Freigabemitteilung (englisch)
SYSLNK.CPP-RS.032	Compiler-Bibliothek (BS2000)
SYSTEMS.CPP-RS.032	Meldungsdatei
SYSSDF.CPP-RS.032	SDF-Syntaxdatei
SYSSDF.CPP-RS.032.IU.USER	SDF-Benutzer-Syntaxdatei II-UPDATE
SYSSDF.CPP-RS.032.USER	SDF-Benutzer-Syntaxdatei Privatinstallation
SYSSII.CPP-RS.032	IMON-Installationsdatei
SYSSPR.CPP-RS.032.IU	SDF-Prozedur für START-II-UPDATE
SYSSSC.CPP-RS.032.POSIX	Subsystem-Deklaration (POSIX)
SYSSSC.CPP-RS.032	Subsystem-Deklaration (BS2000)

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

1.3 Dokumentation

*4 Folgende Dokumentation ist für C/C++-RS V3.2 verfügbar:

deutsche Ausgabe	englische Ausgabe
------------------	-------------------

C-Sprachbeschreibung:

Programmieren in C 2. Ausgabe ANSI-C Kernighan und Ritchie	The C Programming Language 2nd Edition - ANSI-C Kernighan and Ritchie
--	---

C++-Sprachbeschreibung:

Die C++-Programmiersprache 2. Ausgabe von Bjarne Stroustrup	The C++ Programming Language (2nd Edition) by Bjarne Stroustrup
Die C++-Programmiersprache 3. Ausgabe von Bjarne Stroustrup	The C++ Programming Language (3rd Edition) by Bjarne Stroustrup

Compiler-Handbuch (allgemeiner Teil und SDF-Syntax):

*3	C/C++ V3.2D C/C++-Compiler Benutzerhandbuch	C/C++ V3.2D C/C++ Compiler User Guide
----	---	---

Compiler-Handbuch (POSIX-Syntax):

*3	C/C++ V3.2D POSIX-Kommandos des C/C++-Compilers Benutzerhandbuch	C/C++ V3.2D POSIX Commands of the C/C++ Compiler User Guide
----	---	--

System-Compiler-Handbuch:

*7	C/C++-RS V3.2B C/C++ Entwicklungssystem für die Systementwicklung	
----	--	--

C-Bibliotheksfunktionen:

*9	C-Bibliotheksfunktionen Referenzhandbuch Stand November 2015	C Library Functions Reference Manual Edition November 2015
----	--	--

C-Bibliotheksfunktionen für POSIX:

*9	C-Bibliotheksfunktionen für POSIX-Anwendungen Referenzhandbuch Stand November 2015	C Library Functions for POSIX Applications Reference Manual Edition November 2015
----	---	--

C++-Bibliotheksfunktionen für den ANSI-C++-Modus:

-		Standard C++ Library V1.2 User's Guide and Reference U25582-J-Z145-1-76
---	--	---

C++-Bibliotheksfunktionen für den Cfront-Modus:

C++ V2.1 C++-Bibliotheksfunktionen.	C++ V2.1 C++ Library Functions
--	-----------------------------------

Tools.h++-Klassenbibliothek (für den ANSI-C++-Modus):

-	Tools.h++ V7.0 User's Guide
-	Tools.h++ V7.0 Class Reference

Zusätzlich liegen folgende Dokumentationen vor, die sich nicht ausschließlich an den C/C++-Nutzer wenden:

Laufzeitsystem CRTE:

*11	ab CRTE V2.9A Common Runtime Environment Benutzerhandbuch	CRTE V2.9A Common Runtime Environment User Guide
-----	---	--

Daneben ist die Dokumentation der BS2000-Standardkonfiguration für den Betrieb von C/C++-RS notwendig.

*8	Die Dokumentation zu BS2000 ist unter dem Titel „BS2000 SoftBooks“ in deutscher und englischer Sprache auch auf DVD erhältlich.
*10	

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

Zu diesen Handbüchern kann es zusätzlich README-Dateien geben. Sie enthal-
ten Änderungen und Erweiterungen zum Handbuch des jeweiligen Produktes. Die
*6 README-Dateien sind auf der SoftBooks-DVD enthalten bzw. online unter
*6 <http://manuals.ts.fujitsu.com> verfügbar. Die Dateien haben folgenden Namensauf-
bau:

SYSRME.CPP-RS.<version>.D (Datei mit deutschem Text)
SYSRME.CPP-RS.<version>.E (Datei mit englischem Text)

2 Software-Erweiterungen

Im Folgenden werden die Erweiterungen bzw. Verbesserungen gegenüber der Vorgängerversion C/C++-RS V3.1 beschrieben.

2.1 Korrekturen bekannter Fehler

C/C++-RS V3.2 enthält eine Reihe von Fehlerkorrekturen, deren Einsatz wir nachdrücklich empfehlen.

2.2 Steuerbare Behandlung doppelter Entries im POSIX

Mit C/C++-RS V3.2 ist es möglich, bereits während der Produktionsphase zu erkennen, wenn ein Programm Entries mehrfach enthält und deswegen nicht ablauf-fähig ist. Die neuen Optionen '-z dup_warning' bzw. '-z dup_error' steuern, ob in einem solchen Fall lediglich eine Warnung ausgegeben oder ein Fehler ausgelöst werden soll.

2.3 Erzeugung von X86-Code

*10 Mit C/C++-RS ab V3.2B kann X86-Code erzeugt werden.

2.4 Initialisierung

*2 In C/C++-RS ab V3.2C wird die Initialisierung von Arrays und Strukturen mit nicht-Konstanten unterstützt.
*2

2.5 Listengenerierung

*3 Die Listengenerierung wurde in C/C++-RS V3.2D neu implementiert.
*3 Dabei wurden zahlreiche Fehler und Unschönheiten bereinigt.

*3 Eine weitere Verbesserung ist, dass die Listengenerierung jetzt schneller funktio-
*3 niert.

2.6 Switch-Optimierung

*6 C/C++-RS V3.2E enthält eine neu implementierte switch-Optimierung.

3 Technische Hinweise

3.1 Ressourcenbedarf

Beim Ablauf von C/C++-RS wird folgender Speicherbereich im Systemadressraum benötigt:

mindestens 128 MB Klasse-6-Speicher

Der Wert stellt einen Minimalbedarf dar, der sich je nach Datenmenge und Anwendung vergrößern kann (z.B. durch Verwendung von Templates in C++-Quellen).

- *12 C/C++-RS V3.2E belegt nach dem Laden 2215 PAM-Seiten Klasse 6 Speicher.
- *6 bei vorgeladenem Subsystem CPP-RS 950 PAM-Seiten. Beim Erzeugen von
- *12 X86-Code kommen noch 1597 PAM-Seiten für den ASSTRAN hinzu.

- *6 Im Vergleich zur Vorgängerversion hat sich der Speicherbedarf des Compilers
- *6 verringert.

- *3 Der Speicherbedarf für das Vorladen der Subsysteme CPP-RS beträgt ca.
- *12 5,4 MB.
- *3 Der Speicherbedarf für das Vorladen der Subsysteme CPP-RSP beträgt ca.
- *12 5,2 MB.

3.2 SW-Konfiguration

- *12 C/C++ V3.2E ist mit BS2000/OSD-BC ab V9.0 (Business Server der S-Serie),
- *12 OSD/XC ab V9.0 (Business Server der SQ-Serie) und OSD/XC V9.5 (SE Server)
- *12 ablauffähig.

- Hinweis:
- *11 Für die Version 3.2E40 reichte noch OSD V6.0B und CRTE-BASYS V1.6F.

- *6 C/C++-RS V3.2E setzt folgende Korrekturstände der SW-Produkte voraus:

- *11 - CRTE-BASYS ab V1.9A (V10.0A, V11.0A)
- *6 - LLMAM ab V3.4A30

- *3 und die Produkte: BINDER, BUILDER, CRTE, PLAM und SDF in den zu der ein-
- *3 gesetzten OSD-Version passenden Versionen.

- *3 Die Nutzung bestimmter Funktionalitäten setzt weitere Software
- *3 in den zu der eingesetzten OSD-Version passenden Versionen
- *3 voraus:

- *3 - BLSSERV für das dynamische Binden/Laden
- *3 - DSSM zum Vorladen des Compilers
- *3 - LMS für private Installationen von C/C++-RS
- *3 - POSIX-HEADER zum Einsatz des Compilers im POSIX
- *3 - POSIX-BC zum Einsatz des Compilers im POSIX

- *11 Der Compiler CPP-RS V3.2E enthält ASSTRAN V5.2B07.

3.3 Produkt-Installation

*3 3.3.1 Installation und Deinstallation

*3

*3

*6

*3

*3

*3

*3

*6

*6

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

*3

C/C++-RS V3.2 besteht aus den im BS2000 nutzbaren Komponenten (Compiler, Listing-Generator, II-Update-Tool) mit SDF-Oberfläche und den optional installierbaren POSIX-Komponenten (Compiler, Listing-Generator, Tools) für die Nutzung unter POSIX.

C/C++-RS V3.2 wird im BS2000 normalerweise mittels SOLIS bzw. IMON installiert. Die POSIX-Teile von C/C++-RS müssen bei Bedarf dann noch manuell installiert werden, wenn nicht IMON ab Version 2.8 eingesetzt wird. Mit IMON ab Version 2.8 können die POSIX-Teile des Produktes bereits automatisch durch SOLIS installiert werden.

Insgesamt werden folgende Installationsarten unterstützt, die im Folgenden noch näher beschrieben werden:

- Öffentliche Installation (SOLIS) für BS2000
- Automatische öffentliche Installation (SOLIS) für POSIX
- Manuelle öffentliche Installation für POSIX mit und ohne IMON
- Private Installation (Skripte aus SINPRC) für BS2000
- Private Installation (Skripte aus SINPRC) für POSIX

Eine öffentliche Installation ist normalerweise für alle Nutzer einer Anlage verfügbar und erfordert zur Installation entsprechende Administrator-Privilegien, wogegen eine private Installation meist nur für den installierenden Nutzer gedacht ist und auch keine Administrator-Privilegien erfordert.

Es wird keine Update-Installation unterstützt. Vor der Installation einer neuen Version oder eines Korrekturstandes sollte also unbedingt die alte Version deinstalliert werden. Das gleiche trifft zu, wenn während der Installation Fehler auftraten.

Die Deinstallation eines älteren Korrekturstandes (nicht einer älteren Version) ist auch mit einer neueren SINLIB der gleichen Release-Unit möglich und muss daher nicht mehr unbedingt vor dem Einspielen des SOLIS-Bandes erfolgen.

Will man mehrere Versionen und/oder Korrekturstände parallel installieren, so muss man darauf achten, dass die Produktdateien für jede Installation getrennt vorhanden sind, d.h. z.B. für Versionen, die sich nur im Korrekturstand unterscheiden, dass sie auf verschiedenen Kennungen untergebracht sein müssen oder sich durch die Namensgebung der Release-Items unterscheiden müssen (z.B. Präfix). Für den POSIX-Teil der Installation muss das Installationsverzeichnis verschieden von jeder anderen Installation gewählt werden.

*3 3.3.1.1 Öffentliche Installation (SOLIS) für BS2000

*3 Dies ist die Standard-Installationsvariante für den BS2000-Teil des Produktes und
 *3 sollte für die meisten Kunden die empfohlene Installationsvariante sein. Dabei
 *3 übernimmt SOLIS automatisch alle Aufgaben, wie die Platzierung der Produktda-
 *3 teien (nach Vorgabe des Installateurs), Aktivierung von Syntax- und Meldungsda-
 *3 teien sowie Registrierung der vorladbaren Subsysteme. Die von SOLIS angebote-
 *3 nen Möglichkeiten zur Installation auf beliebigen Kennungen und/oder mit modifi-
 *3 zierten Dateinamen werden unterstützt, ebenso wie Parallel-Installationen unter-
 *3 schiedlicher Produktversionen.

*3 3.3.1.1.1 Durchführung der Installation

*3 Die Beschreibung der SOLIS-Installation kann der Dokumentation zu SO-
 *3 LIS/IMON entnommen werden.

*3 3.3.1.1.2 Vorladbare Subsysteme

*3 Der Compiler für BS2000, ebenso wie der für POSIX, liegt als vorladbares Sub-
 *3 system vor. Bei der SOLIS-Installation sind die zugehörigen Subsystem-
 *3 Deklarationen bereits in den System-Katalog eingetragen worden. Die Subsystem-
 *6 Namen sind: CPP-RS und CPP-RSP.

*3 Welches der Subsysteme vorgeladen wird und ob dies bereits beim Systemstart
 *3 geschieht, muss die jeweilige Systemverwaltung entscheiden. Durch das Vorladen
 *3 reduzieren sich die Ladezeiten beim Aufruf des Compilers aber deutlich.

*3 Die oben genannten Subsysteme können theoretisch parallel vorgeladen werden.
 *3 Ein paralleles Vorladen von Subsystemen gleichen Namens aber unterschiedli-
 *3 cher Version ist dagegen nicht zugelassen.

*3 3.3.1.1.3 Benutzung

*3 Nach erfolgter SOLIS-Installation können der Compiler und der Listing-Generator
 *3 über ihre Start-Kommandos ohne weitere Vorkehrungen aufgerufen werden. Für
 *3 die Benutzung des II-Update-Tools dagegen ist fast immer ein zusätzlicher manu-
 *3 eller Installationsschritt erforderlich, der im Abschnitt 3.3.1.7 beschrieben ist.

*3 Allerdings muss man beachten, dass nur die jeweils zuletzt öffentlich im BS2000
 *3 installierte Produkt-Version bzw. -Variante Eigentümer der Start-Kommandos
 *3 START-CPLUS-COMPILER und START-CPLUS-LISTING-GENERATOR ist und
 *3 sein kann.

*3 Es wird daher nicht empfohlen, mehrere Produkt-Varianten oder
 *3 -Versionen öffentlich parallel zu installieren, es ist aber prinzipiell möglich. Die an-
 *3 deren Installationen können dann nur unter Benutzung von START-PROGRAM
 *3 bzw. START-EXECUTABLE-PROGRAM aufgerufen werden, z.B. mit:

```
*6 /START-EXE *LIB($.SYSLNK.CPP-RS.032,SDFCC)
*3 bzw.
*6 /START-EXE *LIB($.SYSLNK.CPP-RS.032,SDFLISTGEN)
```

*3 3.3.1.1.4 Deinstallation

*3 Die Deinstallation des Produktes erfolgt ebenso mit SOLIS. Es ist aber zu beach-
 *3 ten, dass ein manuell installierter POSIX-Teil des Produktes auch zuvor manuell
 *3 deinstalliert werden muss.

*3 3.3.1.2 Automatische öffentliche Installation (SOLIS) für POSIX

*3
 *3 Diese Installationsvariante ist auf Systemen mit IMON ab V2.8 möglich und sollte
 *3 dort für die meisten Kunden die empfohlene Installationsvariante sein. Dabei er-
 *3 folgt der POSIX-Teil der Installation automatisch während bzw. nach der normalen
 *3 SOLIS/IMON-Installation des Produktes und es ist kein weiterer manueller Instal-
 *3 lationsschritt erforderlich.

*3 Diese Installationsvariante ist einfach und bequem, hat aber gewisse Einschrän-
 *3 kungen:

- *3 1. Der Installationspfad ist nicht frei wählbar. Das Produkt wird im POSIX immer
 *3 unter dem Standard-Pfad /opt/C installiert.
- *3 2. Vor der Installation werden alle im POSIX bereits öffentlich installierten Versio-
 *3 nen der gleichen Release-Unit deinstalliert, unabhängig von der Version und
 *3 dem Installationsort. Mit dieser Installationsvariante sind also keine öffentlichen
 *3 Parallel-Installationen von unterschiedlichen Versionen und/oder Korrektur-
 *3 ständen möglich.

*3 3.3.1.2.1 Durchführung der Installation

*3 Die Beschreibung der SOLIS-Installation kann der Dokumentation zu SO-
 *3 LIS/IMON entnommen werden. Wie IMON veranlasst wird, die automatische PO-
 *3 SIX-Installation durchzuführen, ist der Dokumentation zu IMON V2.8 zu entneh-
 *3 men.

*3 3.3.1.2.2 Vorladbares Subsystem

*3 Ein öffentlich installierter POSIX-Compiler kann als Subsystem vom Systemver-
 *3 walter vorgeladen werden (siehe Abschnitt 3.3.1.1.2), das reduziert die Ladezeiten
 *3 beim Start des Compilers. Dagegen kann der Compiler nicht mit dem POSIX-
 *3 Lader posdbl vorgeladen werden.

*3 3.3.1.2.3 Benutzung

*3 Nach erfolgter Installation können der Compiler und der Listing-Generator über ih-
 *3 re POSIX-Kommandos (z.B. cc) ohne weitere Vorkehrungen aufgerufen werden.

*3 3.3.1.2.4 Deinstallation

*3 Die Deinstallation des POSIX-Teiles erfolgt ebenso automatisch, wenn eine ande-
 *6 re Produktversion von C/C++-RS auf diese Weise installiert wird oder wenn die
 *3 Liefereinheit mit SOLIS/IMON vom System entfernt wird.

*3 3.3.1.3 Manuelle öffentliche Installation für POSIX mit IMON

*3 Die Installation erfolgt entsprechend dem POSIX Grundlagen Manual, Kapitel Lie-
 *3 fer- und Installationsverfahren für POSIX-Programmpakete. Für alle Systeme mit
 *3 älterem IMON als V2.8 ist das die empfohlene Installationsvariante.

*3 Es ist die Wahl des Installationspfades im POSIX-Dateisystem möglich, aber die-
 *3 ser sollte nicht bereits durch andere Produkte oder Installationen belegt sein.
 *3 Durch die Wahl eines Installationspfades ist die öffentliche Parallel-Installation
 *6 verschiedener Versionen und/oder Korrekturständen von C/C++-RS möglich.

*3 3.3.1.3.1 Voraussetzungen

*3 Die Installation muss privilegiert unter der System-Default-Kennung (meist TSOS)
 *3 erfolgen und die Release-Items müssen mit IMON installiert, also auch registriert
 *3 sein. Idealerweise setzt diese Installation auf einer vorangegangenen öffentlichen
 *3 Installation für BS2000 auf.

*3 3.3.1.3.2 Durchführung der Installation

*3 Das POSIX-Installationsprogramm wird durch

*3 /START-POSIX-INSTALLATION

*3 gestartet. In der folgenden Maske muss "Install packages on POSIX" ausgewählt
 *3 werden.

*3 Die nächste Maske kann mit folgenden Werten belegt werden:

*3	IMON:	Y
*6	Produkt:	CPP-RS
*3	Package:	
*3	Version:	V03.2 (optional)
*12	Korrekturstand:	E60 (oder ein anderer Stand, optional)

*3 Die Angabe von Version und/oder Korrekturstand ist nur erforderlich, wenn mehre-
 *3 re Versionen des Produktes bei IMON registriert sind.

*3 Nach Abschicken des Schirms wird die Maske nochmals angezeigt, wobei jetzt
 *3 zusätzlich ein Feld zum Eintragen des gewünschten Installationsverzeichnis
 *3 angezeigt wird. Das Feld ist vorbelegt mit dem Standard-Installationspfad /opt/C.

*3 Das Feld kann nun modifiziert werden. Das kann sinnvoll sein, wenn man mehrere
 *3 Versionen und/oder Korrekturstände eines Produktes parallel installieren will. Die
 *3 Installationsskripte prüfen, ob unter diesem Pfad bereits ein anderes Produkt in-
 *3 stalliert ist und brechen die Installation in diesem Fall ab.
 *3 Die Installation startet nach Abschicken des Schirmes automatisch.

*3 3.3.1.3.3 Fehler bei der Installation

*3 Da die Ausgaben von Installationsskripten durch das POSIX-Installations-Tool
 *3 nicht immer korrekt angezeigt werden, wird im Fehlerfall unter
 *3 /var/tmp/inst.<release_unit> eine Datei hinterlassen, der möglicherweise genauere
 *3 Angaben zur Fehlerursache entnommen werden können. Vom POSIX-
 *3 Installations-Tool wird dann eine Fehlermeldung angezeigt, die auf ein Problem
 *3 während der Ausführung der produktspezifischen Skripte hinweist, z.B.:
 *3 shell script ".../install_pre" reports error 102.

*3 Das POSIX-Installations-Tool bricht aber die Installation bei Fehlern nicht immer
 *3 ab, der Anwender muss daher unbedingt das fehlerhaft installierte Produkt selbst
 *3 wieder deinstallieren, bevor er einen neuen Installationsversuch unternimmt. Ins-
 *3 besondere wird ein partiell angelegtes Installationsverzeichnis im Fehlerfall nicht
 *3 automatisch wieder gelöscht.

*3 Die in der Fehlermeldung angegebene Nummer gibt einen Hinweis auf den aufge-
 *3 tretenen Fehler, die eigentlichen Fehlermeldungen der Shell oder anderer Kom-
 *3 mandos findet man dann in der Fehler-Ausgabedatei in /var/tmp.

Fehler- Nummer	Beschreibung
100	Das verwendete POSIX-Installations-Tool hat nicht die für die Installation von C/C++-RS V3.2 erforderliche Version. Ein Update von POSIX auf die aktuellste Version ist erforderlich.
101	Das Installationsverzeichnis wurde vom POSIX-Installations-Tool nicht angelegt (Folgefehler).
102	Der angegebene Installationspfad wird bereits von einer anderen Produkt-Installation benutzt.
103	Es existiert bereits eine POSIX-Installation, die die gleiche SINLIB benutzt.

*3 3.3.1.3.4 Installationsverzeichnis

*3 Die Installation im POSIX-Dateisystem erfolgt in das ausgewählte Verzeichnis.
 *3 Außerdem werden symbolische Links für die Kommandos nach /usr/bin erzeugt,
 *3 wenn und nur wenn der Standard-Installationspfad /opt/C für diese Installation be-
 *3 nutzt wurde.

*3 3.3.1.3.5 Vorladbares Subsystem

*3 Ein öffentlich installierter POSIX-Compiler kann als Subsystem vom Systemver-
 *3 walter vorgeladen werden (siehe Abschnitt 3.3.1.1.2), das reduziert die Ladezeiten
 *3 beim Start des Compilers. Dagegen kann der Compiler nicht mit dem POSIX-
 *3 Lader posdbl vorgeladen werden.

*3 3.3.1.3.6 Benutzung

*3 Die C/C++-RS-Kommandos sind über /usr/bin erreichbar, wenn die Installation in
 *3 den Standard-Pfad erfolgte. Dieses Verzeichnis ist im Standard-Suchpfad jedes
 *3 POSIX-Nutzers eingetragen. Es sollte also für eine derartige Installation keine wei-
 *3 teren Vorkehrungen benötigen, um C/C++-RS zu benutzen. Die Kommandos an-
 *3 derer C/C++-RS-Installationen erreicht man durch Angabe der expliziten Kom-
 *3 mando-Pfade beim Aufruf oder indem der jeweilige Pfad <posix_install_path>/bin
 *3 in den Kommando-Suchpfad des Aufrufers eingetragen wird.

*3 3.3.1.3.7 Deinstallation

*3 Die Deinstallation des POSIX-Teils einer öffentlichen Installation erfolgt mit dem
 *3 POSIX-Installationskommando als TSOS durch Aufruf von:

*3 /START-POSIX-INSTALLATION

*3 In der folgenden Maske muss "Delete packages from POSIX" ausgewählt werden.
 *3 In der nächsten Maske die gewünschte Installation anhand der Version und/oder
 *3 des Installationsverzeichnisses ausgewählt und markiert werden. Nach Absenden
 *3 (DUE) muss nochmals mit DUE bestätigt werden.
 *3 Das Produkt kann danach (keinesfalls vorher) mit SOLIS/IMON vom System ent-
 *3 fernt werden.

*3 3.3.1.4 Manuelle öffentliche Installation für POSIX ohne IMON

*3 Die Installation erfolgt entsprechend dem POSIX Grundlagen Manual, Kapitel Liefer- und Installationsverfahren für POSIX-Programmpakete.

*3 Es ist die Wahl des Installationspfades im POSIX-Dateisystem möglich, aber dieser sollte nicht bereits durch andere Produkte oder Installationen belegt sein. Durch die Wahl eines Installationspfades ist die öffentliche Parallel-Installation verschiedener Versionen und/oder Korrekturständen von C/C++-RS möglich.

*3 Bei der Installation ohne IMON kann man auch den gleichen Korrekturstand von C/C++-RS mehrfach öffentlich installieren (z.B. Bestückung von RZ-Kennungen), was mit IMON nicht möglich ist.

*3 3.3.1.4.1 Voraussetzungen

*3 Die Installation muss privilegiert unter der System-Default-Kennung (meist TSOS) erfolgen. Die Release-Items dürfen dabei auf einer beliebigen Kennung installiert sein.

*3 3.3.1.4.2 Durchführung der Installation

*3 Das POSIX-Installationsprogramm wird durch

*3 `/START-POSIX-INSTALLATION`

*3 gestartet. In der folgenden Maske muss "Install packages on POSIX" ausgewählt werden.

*3 Die nächste Maske kann mit folgenden Werten belegt werden:

*3	IMON:	N
*6	Produkt:	CPP-RS
*3	Package:	
*3	Version:	032
*3	Userid:	<i>Name der Kennung mit den Release-Items</i>

*3 Nach Abschicken des Schirms wird die Maske nochmals angezeigt, wobei jetzt zusätzlich ein Feld zum Eintragen des gewünschten Installationsverzeichnis angezeigt wird. Das Feld ist vorbelegt mit dem Standard-Installationspfad `/opt/C`.

*3 Das Feld kann nun modifiziert werden. Das kann sinnvoll sein, wenn man mehrere Versionen und/oder Korrekturstände eines Produktes parallel installieren will. Die Installationsskripte prüfen, ob unter diesem Pfad bereits ein anderes Produkt installiert ist und brechen die Installation in diesem Fall ab.

*3 Die Installation startet nach Abschicken des Schirmes automatisch.

*3 3.3.1.4.3 Fehler bei der Installation

*3 Siehe Abschnitt 3.3.1.3.3.

*3 3.3.1.4.4 Installationsverzeichnis

*3

*3 Die Installation im POSIX-Dateisystem erfolgt in das ausgewählte Verzeichnis.
*3 Außerdem werden symbolische Links für die Kommandos nach /usr/bin erzeugt,
*3 wenn und nur wenn der Standard-Installationspfad /opt/C für diese Installation be-
*3 nutzt wurde.

*3

*3 3.3.1.4.5 Vorladbares Subsystem

*3

*3 Ein öffentlich installierter POSIX-Compiler kann als Subsystem vom Systemver-
*3 walter vorgeladen werden (siehe Abschnitt 3.3.1.1.2), das reduziert die Ladezeiten
*3 beim Start des Compilers. Dagegen kann der Compiler nicht mit dem POSIX-
*3 Lader posdbl vorgeladen werden.

*3

*3 Für eine nicht mit SOLIS/IMON vorgenommene Installation werden die Subsys-
*3 tem-Deklarationen nicht automatisch in den System-Katalog eingetragen. Dies
*3 müsste von Hand vorgenommen werden, wird für diese Installationsart aber nicht
*3 empfohlen.

*3

*3 3.3.1.4.6 Benutzung

*3

*6 Die C/C++-RS-Kommandos sind über /usr/bin erreichbar, wenn die Installation in
*3 den Standard-Pfad erfolgte. Dieses Verzeichnis ist im Standard-Suchpfad jedes
*3 POSIX-Nutzers eingetragen. Es sollte also für eine derartige Installation keine wei-
*6 teren Vorkehrungen benötigen, um C/C++-RS zu benutzen. Die Kommandos an-
*6 derer C/C++-RS-Installationen erreicht man durch Angabe der expliziten Kom-
*3 mando-Pfade beim Aufruf oder indem der jeweilige Pfad
*3 <posix_install_path>/bin in den Kommando-Suchpfad des Aufrufers eingetragen
*3 wird.

*3

*3 3.3.1.4.7 Deinstallation

*3

*3 Die Deinstallation des POSIX-Teils einer öffentlichen Installation erfolgt mit dem
*3 POSIX-Installationskommando als TSOS durch Aufruf von:

*3

*/START-POSIX-INSTALLATION

*3

*3 In der folgenden Maske muss "Delete packages from POSIX" ausgewählt werden.
*3 In der nächsten Maske die gewünschte Installation anhand der Version und/oder
*3 des Installationsverzeichnisses ausgewählt und markiert werden. Nach Absenden
*3 (DUE) muss nochmals mit DUE bestätigt werden.

*3

*3 Das Produkt kann danach (keinesfalls vorher) mit SOLIS/IMON vom System ent-
*3 fernt werden.

*3

*3 3.3.1.5 Private Installation für BS2000

*3

*6 Die Installation erfolgt durch Prozeduren, die mit der C/C++-RS-Distribution gelie-
*3 fert werden.

*3

*3 3.3.1.5.1 Voraussetzungen

*3

*3 Die Installation kann unter einer beliebigen (nicht privilegierten) Kennung erfolgen.

*3

*6 Die Release-Items von C/C++-RS müssen auf dieser oder einer anderen Kennung
*3 für den Aufrufer zugreifbar sein, insbesondere muss der Aufrufer für die Datei
*3 SYSLNK auch Schreibberechtigung haben.

*3 Diese Datei muss auch inhaltlich im Originalzustand sein und darf nicht bereits
 *3 durch eine vorhergehende Privat-Installation modifiziert worden sein.

*3 Die für eine Privat-Installation unbedingt erforderlichen Release-Items sind (hier
 *3 mit ihren Originalnamen benannt):

```
*6 SINPRC.CPP-RS.032
*6 SYSLNK.CPP-RS.032
*6 SYSMES.CPP-RS.032
*6 SYSSDF.CPP-RS.032.USER
```

*3 3.3.1.5.2 Durchführung der Installation

*3 Stehen die Produktdateien auf der Kennung des Aufrufers unter ihren Originalna-
 *3 men bereit, dann kann die Installation mit dem Kommando

```
*6 /CALL-PROCEDURE (SINPRC.CPP-RS.032,INSTALL.SDF)
```

*3 durchgeführt werden.

*3 Die Release-Items können prinzipiell beliebig umbenannt werden, wenn das bei
 *3 der anschließenden Verwendung berücksichtigt wird. Stehen die Produktdateien
 *3 auf einer anderen Kennung und/oder unter anderem Namen bereit, muss die In-
 *3 stallation in folgender allgemeinerer Form erfolgen:

```
*3 /CALL-PROCEDURE (<sinprc_name>,INSTALL.SDF), -
*3 / (SYSLNK=<syslnk_name>,SYSMES=<sysmes_name>, -
*3 / SINPRC=<sinprc_name>)
```

*3 Die Namen der Release-Items können normale Dateinamen mit/ohne Kennung
 *3 und mit/ohne Pubset-Angabe sein.

*3 3.3.1.5.3 Fehler bei der Installation

*3 Wird die Installation mit Fehler beendet, kann es sein, dass die Datei SYSLNK be-
 *3 reits modifiziert wurde. Sie muss dann vor einem erneuten Installationsversuch
 *3 wieder in den Originalzustand versetzt werden.

*3 3.3.1.5.4 Benutzung

*3 Nach erfolgter Installation können der Compiler und der Listing-Generator benutzt
 *3 werden. Dabei ist zu beachten, dass zumindest einmal pro Session der Mel-
 *3 dungskatalog und die Syntaxdatei der Privat-Installation aktiviert werden müssen.
 *3 Danach können der Compiler bzw. der Listing-Generator mit START-PROGRAM
 *3 bzw. START-EXECUTABLE-PROGRAM aufgerufen werden.

*3 Für eine Privat-Installation unter der Aufrufer-Kennung mit den Originalnamen
 *3 könnte ein Compiler-Aufruf folgendermaßen erfolgen:

```
*6 /MOD-SDF-OPT *ADD(SYSSDF.CPP-RS.032.USER)
*6 /MOD-MSG-FILE ADD=SYSMES.CPP-RS.032
*6 /START-PROG *M(SYSLNK.CPP-RS.032,SDFCC, -
*3 / R-M=A(SHARE-SCOPE=*NONE),P-M=A)
```

*3 Stehen die Produktdateien auf einer anderen Kennung und/oder unter anderem
 *3 Namen bereit, kann z.B. der Aufruf des Listing-Generators in folgender allgemei-
 *3 nerer Form erfolgen:

```
*3 /MOD-SDF-OPT *ADD(<sdfuser_name>
*3 /MOD-MSG-FILE ADD=<sysmes_name>
*3 /START-PROG *M(<syslnk_name>,SDFLISTGEN, -
*3 / R-M=A(SHARE-SCOPE=*NONE),P-M=A)
```

*3 Nach dem Start meldet sich der Compiler mit der Eingabeaufforderung für SDF-
 *3 Anweisungen, analog zu den Start-Kommandos.

*3 Die Option SHARE-SCOPE=*NONE ist erforderlich um sicherzustellen, dass nicht
 *3 eine Verknüpfung zu einem vorgeladenen Subsystem einer anderen öffentlichen
 *3 Installation mit passender Version, aber evtl. unterschiedlichem Korrekturstand
 *3 erfolgt.

*3 3.3.1.5.5 Deinstallation

*3 Die Deinstallation einer Privat-Installation erfolgt einfach durch Löschen der Re-
 *3 lease-Items. Sind die gleichen Release-Items auch Basis einer POSIX-Installation,
 *3 so muss erst das Produkt im
 *3 POSIX deinstalliert werden, bevor die Release-Items gelöscht werden dürfen.

*3 3.3.1.6 Private Installation für POSIX

*3 Die Installation erfolgt durch Prozeduren, die mit der C/C++-RS -Distribution gelie-
 *3 fert werden.

*3 3.3.1.6.1 Voraussetzungen

*3 Die Installation kann unter einer beliebigen (nicht privilegierten) Kennung erfolgen.
 *3 Diese Kennung muss als POSIX-User eingetragen sein und das für die Installation
 *3 gewählte POSIX-Dateisystem muss genügend freien Platz aufweisen.

*6 Die Release-Items von C/C++-RS müssen auf dieser oder einer anderen Kennung
 *3 für den Aufrufer zugreifbar sein, insbesondere muss der Aufrufer für die Datei
 *3 SINLIB auch Schreibberechtigung haben. Diese Datei muss auch inhaltlich im
 *3 Originalzustand sein und darf nicht bereits durch eine vorhergehende Privat-
 *3 Installation modifiziert worden sein.

*3 Das gewählte Installationsverzeichnis im POSIX-Dateisystem darf noch nicht exis-
 *3 tieren, der installierende Nutzer muss aber die Berechtigung haben, es anzulegen.

*3 Die für eine Privat-Installation im POSIX unbedingt erforderlichen Release-Items
 *3 sind (hier mit ihren Originalnamen benannt):

```
*6 SINPRC.CPP-RS.032
*6 SINLIB.CPP-RS.032
*6 SYSMES.CPP-RS.032
```

*3 3.3.1.6.2 Durchführung der Installation

*3
 *3 Stehen die Produktdateien auf der Kennung des Aufrufers unter ihren Originalna-
 *3 men bereit, dann kann die Installation mit dem Kommando

```
*3 /CALL-PROCEDURE -
*6 / (SINPRC.CPP-RS.032,INSTALL.PSX), -
*3 / (IPATH='<posix_install_path>')
```

*3 durchgeführt werden.

*3 Die Release-Items können prinzipiell beliebig umbenannt werden, wenn das bei
 *3 der anschließenden Verwendung berücksichtigt wird. Stehen die Produktdateien
 *3 auf einer anderen Kennung und/oder unter anderem Namen bereit, muss die In-
 *6 stallation für die Produkt-Variante CPP-RS im POSIX in folgender allgemeinerer
 *3 Form erfolgen:

```
*3 /CALL-PROCEDURE (<sinprc_name>,INSTALL.PSX), -
*3 / (IPATH='<posix_install_path>',SINLIB=<sinlib_name>, -
*3 / SYSMES=<sysmes_name>,SINPRC=<sinprc_name>)
```

*3 Die Namen der Release-Items können normale Dateinamen mit/ohne Kennung
 *3 und mit/ohne Pubset-Angabe sein.

*3 Der angegebene Installationspfad darf nicht existieren, der Aufrufer muss aber die
 *3 Berechtigung haben, ihn anzulegen. Handelt es sich um einen relativen Pfadna-
 *3 men, so wird dieser im HOME-Verzeichnis des Benutzers angelegt.

*3 3.3.1.6.3 Fehler bei der Installation

*3 Wird die Installation mit Fehler beendet, kann es sein, dass die Datei SINLIB be-
 *3 reits modifiziert wurde. Sie muss dann vor einem erneuten Installationsversuch
 *3 wieder in den Originalzustand versetzt werden.

*3 3.3.1.6.4 Vorladbares Subsystem

*3 Für privat installierte POSIX-Compiler ist keine Vorladbarkeit vorgesehen. Es gibt
 *3 hier sogar ein Konflikt-Potential. Derzeit gibt es im POSIX keine Möglichkeit (ana-
 *3 log SHARE-SCOPE=*NONE im SDF) zu verhindern, dass sich der Compiler beim
 *3 Start mit dem vorgeladenen Subsystem einer anderen öffentlichen Installation mit
 *3 passender Version, aber evtl. unterschiedlichem Korrekturstand verknüpft.

*3 3.3.1.6.5 Benutzung

*3 Die C/C++-RS-Kommandos sind über <posix_install_path>/bin erreichbar. Der
 *6 Aufruf kann durch Angabe der expliziten Kommando-Pfade erfolgen oder indem
 *3 der jeweilige Pfad <posix_install_path>/bin in den Kommando-Suchpfad des Auf-
 *3 rufers eingetragen wird.

*3 3.3.1.6.6 Deinstallation

*3 Die Deinstallation des POSIX-Teils einer Privat-Installation erfolgt einfach durch
 *3 Löschen des POSIX-Installationsverzeichnisses und der zugehörigen Release-
 *3 Items im BS2000, es sei denn, diese werden noch für eine private BS2000-
 *3 Installation genutzt.

*3 3.3.1.7 Installation II-Update-Tool

*3

*3 Das II-Update-Tool, das mit dem SDF-Compiler ausgeliefert wird, ist nicht IMON-
 *3 fähig und muss daher fast immer mit einer speziellen Prozedur an die aktuelle
 *3 Compiler-Installation angepasst werden. Dies gilt für alle Installationsarten, einzige
 *3 Ausnahme ist eine Compiler-Installation unter der System-Default-Kennung (meist
 *3 TSOS) unter Beibehaltung der Original-Namen der Release-Items.

*3

*3 Auch zum Aufruf sind spezielle Vorkehrungen erforderlich.

*3

*3 3.3.1.7.1 Voraussetzungen

*3

*3 Die Modifikation sollte unter derselben Kennung erfolgen, unter der die
 *6 C/C++-RS-Installation erfolgt ist, die hiermit modifiziert wird.

*3

*6 Die Release-Items von C/C++-RS müssen auf dieser oder einer anderen Kennung
 *3 für den Aufrufer zugreifbar sein, insbesondere muss der Aufrufer für die Dateien
 *3 SYSSPR und SYSSDF.IU.USER auch Schreibberechtigung haben. Diese Dateien
 *3 müssen auch inhaltlich im Originalzustand sein und dürfen nicht bereits durch eine
 *3 vorhergehende Modifikation geändert worden sein.

*3

*3 Die für diese Modifikation unbedingt erforderlichen Release-Items der
 *6 C/C++-RS-Installation sind (hier mit ihren Originalnamen benannt):

*3

*6 SINPRC.CPP-RS.032
 *6 SYSLNK.CPP-RS.032
 *6 SYSSPR.CPP-RS.032.IU
 *6 SYSSDF.CPP-RS.032.IU.USER

*3

*3 3.3.1.7.2 Durchführung der Modifikation

*3

*3 Stehen die Produktdateien auf der Kennung des Aufrufers unter ihren Originalna-
 *3 men bereit, dann kann die Modifikation mit dem Kommando

*3

*6 /CALL-PROCEDURE (SINPRC.CPP-RS.032,INSTALL.IU)

*3

*3 durchgeführt werden.

*3

*3 Die Release-Items können prinzipiell beliebig umbenannt werden, wenn das bei
 *3 der anschließenden Verwendung berücksichtigt wird. Stehen die Produktdateien
 *3 auf einer anderen Kennung und/oder unter anderem Namen bereit, muss die Mo-
 *3 difikation in folgender allgemeineren Form erfolgen:

*3

*3 /CALL-PROCEDURE (<sinprc_name>,INSTALL.IU), -
 *3 / (SYSLNK=<syslnk_name>,SYSSDF=<sdfuser_iu_name>, -
 *3 / SYSSPR=<sysSpr_name>,SINPRC=<sinprc_name>)

*3

*3 Die Namen der Release-Items können normale Dateinamen mit/ohne Kennung
 *3 und mit/ohne Pubset-Angabe sein.

*3 3.3.1.7.3 Fehler bei der Modifikation

*3
 *3 Wird die Modifikation mit Fehler beendet, kann es sein, dass die Datei SYSSPR
 *3 und/oder SYSSDF.IU.USER bereits modifiziert wurde. Sie müssen dann vor ei-
 *3 nem erneuten Versuch wieder in den Originalzustand versetzt werden.

*3 3.3.1.7.4 Benutzung

*3
 *3 Nach erfolgter Modifikation kann das II-Update-Tool benutzt werden. Dabei ist zu
 *3 beachten, dass zumindest einmal pro Session die Syntaxdatei des Tools aktiviert
 *3 werden muss. Danach kann das Start-Kommando aufgerufen werden.

*3 Für eine Installation unter der Aufrufer-Kennung mit den Originalnamen könnte ein
 *3 Tool-Aufruf folgendermaßen erfolgen:

```
*3      /MOD-SDF-OPT *ADD(SYSSDF.CPP-RS.032.IU.USER)
*3      /START-II-UPDATE <command_parameter>
```

*3 Stehen die Produktdateien auf einer anderen Kennung und/oder unter anderem
 *3 Namen bereit, kann z.B. der Aufruf des Tools in folgender Form erfolgen:

```
*3      /MOD-SDF-OPT *ADD(<sdfuser_iu_name>)
*3      /START-II-UPDATE <command_parameter>
```

*3 **3.3.2 Benutzung anderer Produkte**

*3
 *3 Für den normalen Ablauf des Compilers werden auf dem entsprechenden System
 *3 weitere Produkte benötigt. Unabhängig von der Installationsart des Compilers er-
 *3 mittelt er den Standort der Produktdateien der benötigten Produkte per IMON und
 *3 misslingt das, geht er von einer Standard-Installation des jeweiligen Produktes un-
 *3 ter TSOS aus. Kann der Compiler die benötigten fremden Produktdateien so nicht
 *3 finden oder haben sie nicht die benötigte Mindest-Version, kann der Compiler be-
 *3 stimmte Teilfunktionen nicht erbringen.

3.4 Produkt-Einsatz

3.4.1 C++-Source- und Objekt-Kompatibilität

Die bis V2.2 einzig mögliche C++-Sprachvariante (Cfront V3.1.3) wird seit der V3.0 mit dem Sprach-Modus CPP (MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUSPLUS(MODE=CPP) bzw. (im POSIX) mit -X d) eingestellt. Derart übersetzte Objekte können mit C++-Objekten, die mit C/C++ vor V3 erzeugt wurden, in einer Anwendung gemischt werden.

Als Defaulteinstellung wird ab der V3.0 der (modernere) ANSI-Modus verwendet. Cfront-Modus und ANSI-Modi dürfen nicht gemischt werden: Cfront-C++-Objekte können nicht mit ANSI-C++-Objekten zusammengebunden werden!

3.4.2 Native Code

*10 Bei Code-Generierung mit DESTINATION-CODE=*X86 (bzw. -K x86) wird
 *10 native Objektcode generiert. Dieser ist nur auf SQ-Servern und der x86-Server
 *10 Unit der SE-Server ablauffähig. Der erzeugte Objektcode ist wesentlich effizienter,
 *10 da er nicht emuliert werden muss.

- *10 Die Effizienzsteigerung ist nur aus der Realzeit, nicht aus der auf SQ-Servern in den meisten Fällen standardmäßig eingestellten 'normierten CPU-Zeit' ersichtlich. Die normierte CPU-Zeit gewichtet nämlich den CPU-Verbrauch von /390-Code und native Code unterschiedlich, um eine normierte Abrechnung der benötigten Elementaroperationen, die sich immer auf /390 Operationen beziehen, zu ermöglichen.
- *1 Beachten Sie, dass native-Code erheblich umfangreicher ist als /390-Code.
- *10 X86-Native-Code ist ca. dreimal so groß, wie /390-Code.
Hinzu kommt ggf. das C-Laufzeitsystem, falls es fest dazu gebunden wird.
- *1 Das dynamisch nachladbare C-Laufzeitsystem in X86-Codierung
- *1 (SKULNK.CRTE.PARTIAL-BIND bzw. SKULNK.CRTE.COMPL) ist zu groß, um noch in den unteren Adressraum geladen werden zu können.
- *10 Die Übersetzungszeiten für X86-Code sind länger als für /390-Code.
- *10 Beachten Sie auch, dass der Benutzeradressraum gegenüber der /390-Code-Generierung für X86 ca. dreimal so groß sein muss.

3.5 Entfallene (und gekündigte) Funktionen

3.5.1 Entfallene Funktionen

Die folgenden Funktionen der Version 3.1 sind in V3.2 entfallen:

3.5.1.1 RISC(MIPS)-Code-Erzeugung

Wie angekündigt wurde die Erzeugung von RISC(MIPS)-Code in der Version 3.1 letztmalig unterstützt. Die Optionen

MODIFY-MODULE-PROPERTIES DESTINATION-CODE = *RISC-4000

und

MODIFY-BIND-PROPERTIES RUNTIME-ENVIRONMENT = *RISC-4000

bzw.

-K risc_4000

werden dem entsprechend in der V3.2 nicht mehr angeboten.

3.5.2 Gekündigte Funktionen

3.5.2.1 POSIX-Optionen

Folgende POSIX-Optionen werden in Folgeversionen nicht mehr garantiert:

bis V2.2	ab V3.0 ersetzt durch ..
-F R<=...>	-F loopunroll<,...>
-OI<...>	-F i<...>
-R Tc	-K statistics
-R Ti	-K no_integer_overflow
-R Tp	-K no_prompting
-R Ts=...	-K stacksize=...
-W 0	-R min_weight,errors
-W 1	-R min_weight,warnings
-W 2	-R min_weight,notes
-X I	-N ls oder -N source_error
-X la	-N la oder -N object
-X li=0	-K include_all
-X li=1	-K include_name
-X li=2	-K include_user
-X lign	-K pragmas_ignored
-X lpp=...	-N output,,,...
-X lm	-N data_allocation_map
-X lp	-N project
-X lr	-N output,,for_rotation_print
-X ls	-N ls oder -N source_error
-X lx	-N lx oder -N cross_reference
-X C	-K subcall_lab
-X EC	-K calendar_etpnd
-X EJ	-K julian_etpnd
-X IFN	-N cif,project
-X IFX	-N cif,cross_reference
-X II	-K ilcs_opt
-X IO	-K ilcs_out
-X L	-K enum_long
-X LLMK	-K llm_keep
-X LLML	-K llm_case_lower
-X M	-K external_multiple
-X OD=...	-N output,...
-X RC	-K roconst
-X RS	-K rostr
-X S	-K share
-X U	-K external_unique
-X W	-K workspace_stack

3.6 Inkompatibilitäten

3.6.1 LL-Bibliotheken

Die Long-Long-Bibliotheken (Suffix .LL) werden für das Binden nicht mehr gebraucht. Sie werden nicht mehr mit C/C++-RS ausgeliefert und müssen ggf. aus den Bindeprozeduren entfernt werden.

3.6.2 Dateinamen im II-File

Für ANSI-C++-Programme mit Templates erzeugt der Compiler II-Files, in denen u.a. Dateinamen (z.B. der Source-Bibliothek) hinterlegt sind. Bei Umbenennungen (bzw. Verlagerungen auf andere Kennungen oder Pubsets) müssen diese Änderungen im II-File nachgezogen werden. Dafür gibt es das Tool II-UPDATE.

Bereits seit der V3.1B stehen die Dateinamen im II-File nicht mehr genauso wie sie im COMPILE-Statement angegeben wurden, sondern werden eventuell um Cat-Ids und User-Ids ergänzt.

Dies ist bei der Verwendung von II-UPDATE zu berücksichtigen.

Gegebenenfalls können die Namen mit der neuen SHOW-Funktion des Tools angezeigt werden (s.a. Beschreibung im Benutzerhandbuch).

3.6.3 Kombination ANSI-C++ und ASCII- bzw. IEEE-Option

C++-Bibliotheksfunktionen unterstützen weder das ASCII- noch das IEEE-Format. Seit der V3.2 wird eine solche Kombination bereits zum Übersetzungszeitpunkt erkannt und mit den neuen Meldungen CFE2075 (ASCII) bzw. CFE2077 (IEEE) als Fehler abgewiesen. Da es auch Anwendungsfälle gibt, in denen die Kombination unschädlich ist, ist es erlaubt, die Fehlermeldungen ggf. zu Warnungen herunter zu stufen (mit Optionen MODIFY-DIAGNOSTIC-PROPERTIES bzw. -R ..).

3.6.4 Wichtiger Hinweis zu CRTE

Bei der Produktion von ANSI-C++-Anwendungen kann es zu Inkompatibilitäten kommen, wenn die Anwendung C++-Standard-Libraries (SYSLNK.CRTE.STDCPP) aus verschiedenen CRTE-Versionen benutzt. Inkompatibel sind die sog. assoziativen Container, das sind die Klassen set, multiset, map und multimap, von CRTE ab V2.2A15 gegenüber jenen älterer CRTE-Versionen. (Ältere CRTE-Versionen (< V2.2A15) wurden nur bis 1999 ausgeliefert).

Ggf. ist die Anwendung mit der neuen CRTE-Version neu zu produzieren (übersetzen und binden). (Die Neu-Übersetzung ist notwendig, damit Header und Objekte aus CRTE ab V2.2A15 stammen und somit zusammen passen.)

Wird die Inkompatibilität nicht beachtet, so kommt es zu nicht allgemeingültig vorhersagbaren Ergebnissen: Das nicht korrekt produzierte Programm kann beim Ablauf z.B. abstürzen oder in eine Endlosschleife geraten.

3.7 Einschränkungen

3.7.1 Einschränkungen bei der Native-Code Erzeugung

- *10 Bei der Native-Code-Erzeugung (-K x86 bzw.
 *10 MODIFY-MODULE-PROPERTIES DESTINATION-CODE=*X86)
 gibt es die folgenden spezifischen Einschränkungen:

3.7.2 Maximale Namenslänge

Entry-Namen, die länger als 4096 Zeichen sind, können zu Fehlern führen (z.B. Abbruch der Übersetzung mit Dump). Da Entry-Namen nicht direkt vom Anwender vergeben, sondern aus den vom Anwender gewählten Namen generiert werden (und dabei länger werden), kann kein Grenzwert für Anwendernamen genannt werden.

Die generierten Namen lassen sich ggf. in der Projekt-Liste (MODIFY-LIST-PROPERTIES PROJECT-INFORMATION=*YES bzw. -N project) anzeigen (die Originalnamen werden dabei ggf. gekürzt).

Die nachfolgenden Punkte sind unabhängig vom erzeugten Code:

3.7.3 Nicht unterstützte LISTING-Option COMMENTS

Die Listing-Option MODIFY-LISTING-PROPERTIES PREPROCESSING-RESULT = *YES(COMMENTS=*NO) wird dzt. nicht unterstützt, d.h. Kommentare sind immer im Präprozessor-Listing enthalten.

3.7.4 AID-Fehler bei leeren Konstruktoren

Enthalten C++-Programm leere Konstruktoren, so kann es u.U. beim Debuggen zu AID-Fehlern kommen:

```
AID0252 AID error in module 56 : RTC 0E (CMD: TRACE)
```

3.7.5 Besonderheiten im Cfront-C++-Sprachmodus

Für jede Funktion in C++ wird ein externer Name generiert, der in verschlüsselter Form auch die Typen der Parameter enthält. Leider hat es dabei in der Version 2.2 Fehler gegeben, die nun aus Kompatibilitätsgründen auch in der vorliegenden Version enthalten sind. Sie treten ausschließlich im Cfront-C++-Modus auf (MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUS-PLUS(MODE=CPP) bzw. -Xd).

Die folgenden drei Funktionspaare erhalten jeweils denselben externen Namen und führen daher beim Binden zu Duplikaten bzw. (wenn native Code erzeugt wird) u.U. zu Fehlern bei der Ausführung des fertigen Programms:

```
f(char)
f(signed char)
```

C/C++-RS V2.2 kannte kein signed und somit werden beide Funktionen auf denselben Namen abgebildet. Das betrifft nur 'signed char'.

```
f(char (*x)[15])
f(char (*x)[18])
```

Gleiche Namen, da die Grenzen der Arrays nicht berücksichtigt werden.

```
f(const c *)
f(c *)
```

wenn c z.B. als 'typedef char c;' deklariert wurde: Ein const-Qualifikator auf einen Typedef bleibt im externen Namen unsichtbar, beide Funktionen erhalten daher denselben Namen.

3.7.6 CFE1079 bei Verwendung der Optionen IEEE bzw. ASCII

Die Verwendung der Übersetzungsoptionen FP-ARITHMETICS=*IEEE (-K ieee_floats) bzw. LITERAL-ENCODING=*ASCII oder *ASCII-FULL (-K literal_encoding_ascii oder -K literal_encoding_ascii_full) kann zum Fehler 'CFE1079 ERROR ...: Typangabe erwartet' führen, wenn die im Compiler-Benutzerhandbuch beschriebenen Voraussetzungen (Deklaration der C-Bibliotheksfunktionen nicht explizit in der Quelle, sondern durch Inkludieren der entsprechenden CRTE-Header) nicht beachtet wurden.

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 Benutzer-Includes (LISTING-Option)
- Ablaufprotokoll (MSG=FH)
- Präprozessor-Output/Source
- Objektliste
- Binderliste
- Ein-/Ausgabedateien
- Erwartetes Ergebnis
- DUMP, falls vorhanden
- Kurzbeschreibung des Ablaufs

4 Hardware-Unterstützung

- *10 C/C++-RS V3.2E ist auf allen Business Servern einsetzbar, auf denen die vorausgesetzte Software freigegeben ist.