Fujitsu Software BS2000 C/C++
Version 4.0C
Januar 2026

Freigabemitteilung

1 Allgemeines

C/C++ V4.0 ist die Nachfolgeversion von C/C++ V3.2.

Der Name der Liefergruppe und damit Namensbestandteil der Produktdateien ist CPP.

C/C++ ist der strategische BS2000-Compiler zur Entwicklung und zur Portierung von Anwendungen aus der offenen Welt (z.B. OO-Anwendungen) auf BS2000 Business Server.

Diese Freigabemitteilung enthält in gedrängter Form die wesentlichen Erweiterungen, Abhängigkeiten und Betriebshinweise zu C/C++ V4.0.

*10 Der Inhalt entspricht dem Freigabestand: Januar 2026.

*1 *1 Änderungen gegenüber dem Freigabestand: Dezember 2019 sind mit „*1“ gekennzeichnet.

*2 *2 Änderungen gegenüber dem Freigabestand: November 2020 sind mit „*2“ gekennzeichnet.

*3 *3 Änderungen gegenüber dem Freigabestand: November 2021 sind mit „*3“ gekennzeichnet.

*4 *4 Änderungen gegenüber dem Freigabestand: Juni 2022 sind mit „*4“ gekennzeichnet.

*5 *5 Änderungen gegenüber dem Freigabestand: Juni 2023 sind mit „*5“ gekennzeichnet

*6 *6 Änderungen gegenüber dem Freigabestand: November 2023 sind mit „*6“ gekennzeichnet

*7 *7 Änderungen gegenüber dem Freigabestand: Juni 2024 sind mit „*7“ gekennzeichnet

*8 *8 Änderungen gegenüber dem Freigabestand: November 2024 sind mit „*8“ gekennzeichnet

*9 *9 Änderungen gegenüber dem Freigabestand: Juni 2025 sind mit „*9“ gekennzeichnet

*10 *10 Änderungen gegenüber dem Freigabestand: November 2025 sind mit „*10“ gekennzeichnet

*3 *3 Diese und andere aktuelle Freigabemitteilungen sind online verfügbar unter https://bs2manuals.ts.fujitsu.com/.

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

1.1 Bestellung

C/C++ V4.0 kann über Ihre zuständige Vertriebsgesellschaft bezogen werden.

1.2 Auslieferung

Die Lieferung der Dateien zu C/C++ V4.0 erfolgt mit dem Lieferverfahren SOLIS.

Folgende Lieferbestandteile werden auf jeder Hardware benötigt:

	SINLIB.CPP.040	Compiler-Bibliothek (POSIX)
	SINPRC.CPP.040	Bibliothek mit Privatinstallation-Prozeduren
*2	(Entfällt)	
	SYSLNK.CPP.040	Compiler-Bibliothek (BS2000)
	SYSMES.CPP.040	Meldungsdatei
	SYSSDF.CPP.040	SDF-Syntaxdatei
	SYSSDF.CPP.040.IU.USER	SDF-Benutzer-Syntaxdatei II-UPDATE
	SYSSDF.CPP.040.USER	SDF-Benutzer-Syntaxdatei Privatinstallation
	SYSSII.CPP.040	IMON-Installationsdatei
	SYSSPR.CPP.040.IU	SDF-Prozedur für START-II-UPDATE
	SYSSSC.CPP.040.POSIX	Subsystem-Deklaration (POSIX)
	SYSSSC.CPP.040	Subsystem-Deklaration (BS2000)

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

1.3 Dokumentation

Folgende Dokumentation ist für C/C++ V4.0 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):

C/C++ V4.0 Benutzerhandbuch (BHB)	C/C++ V4.0 User Guide

Compiler-Handbuch (POSIX-Syntax):

C/C++ V4.0 POSIX-Kommandos Benutzerhandbuch	C/C++ V4.0 POSIX Commands User Guide

C-Bibliotheksfunktionen:

C-Bibliotheksfunktionen Referenzhandbuch	C Library Functions Reference Manual

C-Bibliotheksfunktionen für POSIX:

C-Bibliotheksfunktionen für POSIX-Anwendungen Referenzhandbuch	C Library Functions for POSIX Applications Reference Manual

C++-Bibliotheksfunktionen für den V3-Modus (ANSI-C++):

-	Standard C++ Library V1.2 User's Guide and Reference

C++-Bibliotheksfunktionen für den V2-Modus (Cfront):

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

Tools.h++-Klassenbibliothek (für den V3-Modus, ANSI-C++):

-	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:

*8	ab	CRTE V21.0 Common Runtime Environment Benutzerhandbuch	CRTE V21.0 Common Runtime Environment User Guide

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

*3 Die Dokumentation ist im Internet unter https://bs2manuals.ts.fujitsu.com verfügbar. Dort finden Sie sowohl einzelne Handbücher als auch (unter dem Reiter „Softbooks“) das ISO-Image einer DVD mit dem Gesamtbestand.

2 Software-Erweiterungen

Im Folgenden werden die Erweiterungen bzw. Verbesserungen gegenüber der Vorgängerversion C/C++ V3.2 beschrieben.

2.1 Korrekturen bekannter Fehler

C/C++ V4.0 enthält eine Reihe von Fehlerkorrekturen, deren Einsatz wir nachdrücklich empfehlen.

2.2 Unterstützung des C11-Standards

Der Compiler akzeptiert Code entsprechend dem C11-Standard.
*1 Es gibt einige optionale Funktionen aus dem C11-Standard, die CPP 4.0 nicht un-
*1 terstützt.
Einschränkungen sind in den Manualen zu C/C++ V4.0 beschrieben.

2.3 Unterstützung des C++2017-Standards

Der Compiler akzeptiert Code entsprechend dem C++2017-Standard.
*1 Es gibt einige obligatorische Funktionen aus dem C++2017-Standard, die CPP 4.0
*1 nicht unterstützt.
Einschränkungen sind in den Manualen zu C/C++ V4.0 beschrieben.

*1 2.4 #pragma instantiate_inline

*1 Der Compiler unterstützt das `#pragma instantiate_inline`.

*2 2.5 Unterstützung des C++2020-Standards

*2 Der Compiler akzeptiert Code entsprechend dem C++2020-Standard.
*2 Es gibt einige obligatorische Funktionen aus dem C++2020-Standard, die CPP 4.0
*2 nicht unterstützt. Zudem ist die Bibliothek des C++2020-Standards nicht verfüg-
*2 bar, es kann jedoch die Bibliothek des C++2017-Standards stattdessen verwendet
*2 werden.
*2 Einschränkungen sind in den Manualen zu C/C++ V4.0 beschrieben.

*2 2.6 Formatüberprüfung für printf- und scanf-Funktionen

*2 Die Formatüberprüfung für die Funktionen fprintf(), printf(), sprintf(), snprintf(),
*2 fscanf(), scanf() und sscanf() sowie deren ASCII- und IEEE-Varianten wird einge-
*2 schaltet (siehe Benutzerhandbuch Pragmas `__printf_args` bzw. `__scanf_args`).

*2 Diese Prüfung verlangt CRTE V11.1A40, bzw. V21.0A10.

*3 2.7 C++ Prädefines

*3 Die im C++-Standard definierten Prädefines (beginnend mit „__cpp“) sind definiert.
*3 Diese zeigen die Anwesenheit von bestimmten C++-Features im Compiler an.
*3 Manche der betroffenen Features hängen vom Sprachmodus ab. Dementspre-
*3 chend ist der Wert des korrespondierenden Prädefines vom Sprachmodus abhän-
*3 gig.

*3 2.8 C++2020 „using enum“-Feature

*3 Es wird die Anwesiung „using enum E;“ unterstützt. Dadurch können Enumerato-
*3 ren einer „enum class E{…};“ besser angesprochen werden.

*4 2.9 Unterstützung von Bibliotheks-Versionen

*5 Es gibt die neue Anweisung //MODIFY-LIBRARY-VERSION zur Angabe der C++
*5 Bibliotheks-Versionen.
*4 Details zu dieser Anweisung sind in den Manualen zu C/C++ V4.0 beschrieben.

*7 2.10 Unterstützung von %m im printf-Format

*7 Die Angabe von `printf("%m")` entspricht der Angabe von
*7 `printf("%s", strerror(errno));`

*7 2.11 Bitfelder „long long“

*7 Der Compiler akzeptiert „long long“ Bitfelder ab dieser Version.

3 Technische Hinweise

3.1 Ressourcenbedarf

Beim Ablauf wird folgender Speicherbereich im System- und Benutzeradressraum benötigt:

mindestens 64 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).

*8 C/C++ V4.0 belegt nach dem Laden 4735 Seiten im Klasse 6 Speicher, bei vorge-
*8 ladenem Subsystem CPP 2202 Seiten.

Der Speicherbedarf für das Vorladen des Subsystems CPP beträgt ca. 5,4 MB.
Der Speicherbedarf für das Vorladen des Subsystems CPPP beträgt ca. 5,4 MB.

3.2 SW-Konfiguration

*8 C/C++ V4.0 ist mit BS2000 OS DX V1.0 ablauffähig.

C/C++ V4.0 setzt folgende Korrekturstände der SW-Produkte voraus:

*8
- CRTE-BASYS V21.0A
- LLMAM ab V3.4A30

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

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

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

3.3 Produkt-Installation

3.3.1 Installation und Deinstallation

C/C++ V4.0 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++ V4.0 wird im BS2000 normalerweise mittels SOLIS bzw. IMON installiert. Die POSIX-Teile von C/C++ 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.

3.3.1.1 Öffentliche Installation (SOLIS) für BS2000

Dies ist die Standard-Installationsvariante für den BS2000-Teil des Produktes und sollte für die meisten Kunden die empfohlene Installationsvariante sein. Dabei übernimmt SOLIS automatisch alle Aufgaben, wie die Platzierung der Produktdateien (nach Vorgabe des Installateurs), Aktivierung von Syntax- und Meldungsdateien sowie Registrierung der vorladbaren Subsysteme. Die von SOLIS angebotenen Möglichkeiten zur Installation auf beliebigen Kennungen und/oder mit modifizierten Dateinamen werden unterstützt, ebenso wie Parallel-Installationen unterschiedlicher Produktversionen.

3.3.1.1.1 Durchführung der Installation

Die Beschreibung der SOLIS-Installation kann der Dokumentation zu SOLIS/ IMON entnommen werden.

3.3.1.1.2 Vorladbare Subsysteme

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

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

Die oben genannten Subsysteme können theoretisch parallel vorgeladen werden. Ein paralleles Vorladen von Subsystemen gleichen Namens aber unterschiedlicher Version ist dagegen nicht zugelassen.

3.3.1.1.3 Benutzung

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

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

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

```
/START-EXE *LIB($.SYSLNK.CPP.040,SDFCC)
```

bzw.

```
/START-EXE *LIB($.SYSLNK.CPP.040,SDFLISTGEN)
```

3.3.1.1.4 Deinstallation

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

3.3.1.2 Automatische öffentliche Installation (SOLIS) für POSIX

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

Diese Installationsvariante ist einfach und bequem, hat aber gewisse Einschränkungen:

1. Der Installationspfad ist nicht frei wählbar. Das Produkt wird im POSIX immer unter dem Standard-Pfad /opt/C installiert.

2. Vor der Installation werden alle im POSIX bereits öffentlich installierten Versionen der gleichen Release-Unit deinstalliert, unabhängig von der Version und dem Installationsort. Mit dieser Installationsvariante sind also keine öffentlichen Parallel-Installationen von unterschiedlichen Versionen und/oder Korrekturständen möglich.

3.3.1.2.1 Durchführung der Installation

Die Beschreibung der SOLIS-Installation kann der Dokumentation zu SOLIS/IMON entnommen werden. Wie IMON veranlasst wird, die automatische POSIX-Installation durchzuführen, ist der Dokumentation zu IMON V2.8 zu entnehmen.

3.3.1.2.2 Vorladbares Subsystem

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

3.3.1.2.3 Benutzung

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

3.3.1.2.4 Deinstallation

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

3.3.1.3 Manuelle öffentliche Installation für POSIX mit IMON

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

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++ möglich.

3.3.1.3.1 Voraussetzungen

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

3.3.1.3.2 Durchführung der Installation

Das POSIX-Installationsprogramm wird durch

```
/START-POSIX-INSTALLATION
```

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

Die nächste Maske kann mit folgenden Werten belegt werden:

IMON:	Y
Produkt:	CPP
Package:	
Version:	V04.0 (optional)
Korrekturstand:	A00 (oder ein anderer Stand, optional)

Die Angabe von Version und/oder Korrekturstand ist nur erforderlich, wenn mehrere Versionen des Produktes bei IMON registriert sind.

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

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.
Die Installation startet nach Abschicken des Schirmes automatisch.

3.3.1.3.3 Fehler bei der Installation

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

shell script ".../.install_pre" reports error 102.

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

Die in der Fehlermeldung angegebene Nummer gibt einen Hinweis auf den aufgetretenen Fehler, die eigentlichen Fehlermeldungen der Shell oder anderer Kommandos 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++ V4.0 erforderliche Version. Ein Update von POSIX auf die aktuellste Version ist erforderlich.
101	Das Installations-Verzeichnis 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.

3.3.1.3.4 Installationsverzeichnis

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

3.3.1.3.5 Vorladbares Subsystem

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

3.3.1.3.6 Benutzung

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

3.3.1.3.7 Deinstallation

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

/START-POSIX-INSTALLATION

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

3.3.1.4 Manuelle öffentliche Installation für POSIX ohne IMON

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

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++ möglich.

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

3.3.1.4.1 Voraussetzungen

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

3.3.1.4.2 Durchführung der Installation

Das POSIX-Installationsprogramm wird durch

/START-POSIX-INSTALLATION

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

Die nächste Maske kann mit folgenden Werten belegt werden:

IMON:	N
Produkt:	CPP
Package:	
Version:	040
Userid:	*Name der Kennung mit den Release-Items*

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

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.

Die Installation startet nach Abschicken des Schirmes automatisch.

3.3.1.4.3 Fehler bei der Installation

Siehe Abschnitt 3.3.1.3.3.

3.3.1.4.4 Installationsverzeichnis

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

3.3.1.4.5 Vorladbares Subsystem

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

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

3.3.1.4.6 Benutzung

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

3.3.1.4.7 Deinstallation

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

```
/START-POSIX-INSTALLATION
```

In der folgenden Maske muss "Delete packages from POSIX" ausgewählt werden. In der nächsten Maske die gewünschten Installation anhand der Version und/oder des Installations-Verzeichnisses ausgewählt und markiert werden. Nach Absenden (DUE) muss nochmals mit DUE bestätigt werden.

Das Produkt kann danach (keinesfalls vorher) mit SOLIS/IMON vom System entfernt werden.

3.3.1.5 Private Installation für BS2000

Die Installation erfolgt durch Prozeduren, die mit der C/C++-Distribution geliefert werden.

3.3.1.5.1 Voraussetzungen

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

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

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

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

```
SINPRC.CPP.040
SYSLNK.CPP.040
SYSMES.CPP.040
SYSSDF.CPP.040.USER
```

3.3.1.5.2 Durchführung der Installation

Stehen die Produktdateien auf der Kennung des Aufrufers unter ihren Originalnamen bereit, dann kann die Installation mit dem Kommando

```
/CALL-PROCEDURE (SINPRC.CPP.040,INSTALL.SDF)
```

durchgeführt werden.

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

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

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

3.3.1.5.3 Fehler bei der Installation

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

3.3.1.5.4 Benutzung

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

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

```
/MOD-SDF-OPT *ADD(SYSSDF.CPP.040.USER)
/MOD-MSG-FILE ADD=SYSMES.CPP.040
/START-PROG *M(SYSLNK.CPP.040,SDFCC, -
/    R-M=A(SHARE-SCOPE=*NONE),P-M=A)
```

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

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

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

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

3.3.1.5.5 Deinstallation

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

3.3.1.6 Private Installation für POSIX

Die Installation erfolgt durch Prozeduren, die mit der C/C++-Distribution geliefert werden.

3.3.1.6.1 Voraussetzungen

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

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

Das gewählte Installations-Verzeichnis im POSIX-Dateisystem darf noch nicht existieren, der installierende Nutzer muss aber die Berechtigung haben, es anzulegen.

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

```
SINPRC.CPP.040
SINLIB.CPP.040
SYSMES.CPP.040
```

3.3.1.6.2 Durchführung der Installation

Stehen die Produktdateien auf der Kennung des Aufrufers unter ihren Originalnamen bereit, dann kann die Installation mit dem Kommando

```
/CALL-PROCEDURE -
/   (SINPRC.CPP.040,INSTALL.PSX), -
/   (IPATH='<posix_install_path>')
```

durchgeführt werden.

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

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

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

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

3.3.1.6.3 Fehler bei der Installation

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

3.3.1.6.4 Vorladbares Subsystem

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

3.3.1.6.5 Benutzung

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

3.3.1.6.6 Deinstallation

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

3.3.1.7 Installation II-Update-Tool

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

Auch zum Aufruf sind spezielle Vorkehrungen erforderlich.

3.3.1.7.1 Voraussetzungen

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

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

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

```
SINPRC.CPP.040
SYSLNK.CPP.040
SYSSPR.CPP.040.IU
SYSSDF.CPP.040.IU.USER
```

3.3.1.7.2 Durchführung der Modifikation

Stehen die Produktdateien auf der Kennung des Aufrufers unter ihren Originalnamen bereit, dann kann die Modifikation mit dem Kommando

```
/CALL-PROCEDURE (SINPRC.CPP.040,INSTALL.IU)
```

durchgeführt werden.

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

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

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

3.3.1.7.3 Fehler bei der Modifikation

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

3.3.1.7.4 Benutzung

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

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

```
/MOD-SDF-OPT *ADD(SYSSDF.CPP.040.IU.USER)
/START-II-UPDATE <command_parameter>
```

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

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

3.3.2 Benutzung anderer Produkte

Für den normalen Ablauf des Compilers werden auf dem entsprechenden System weitere Produkte benötigt. Unabhängig von der Installationsart des Compilers ermittelt er den Standort der Produktdateien der benötigten Produkte per IMON und misslingt das, geht er von einer Standard-Installation des jeweiligen Produktes unter TSOS aus. Kann der Compiler die benötigten fremden Produktdateien so nicht finden oder haben sie nicht die benötigte Mindest-Version, kann der Compiler bestimmte 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 V4.0 mit dem Sprach-Modus *V2.
(MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUSPLUS(MODE=*V2) 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.

Die in V3.x zusätzliche C++-Sprachvariante (ANSI oder STRICT ANSI) wird seit V4.0 mit dem Sprach-Modus *V3.
(MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUSPLUS(MODE=*V3) oder (MODE=STRICT-ANSI) bzw. (im POSIX) mit -X w oder e) eingestellt. Derart übersetzte Objekte können mit C++-Objekten, die mit C/C++ vor V4 erzeugt wurden, in einer Anwendung gemischt werden.

*2 Als Default-Einstellung wird ab sofort der C++2020-Modus verwendet.
*V2 (Cfront), *V3 (ANSI) und C++2017-Modi dürfen nicht gemischt werden:
*V2-C++-, *V3-C++- und C++2017-Objekte können nicht zusammengebunden werden!

*2 Der C++2017-Modus und C++2020-Modus sind zueinander binärkompatibel.

3.5 Entfallene (und gekündigte) Funktionen

Die folgenden Funktionen der Version 3.2 sind in V4.0 entfallen oder gekündigt:

3.5.1 Entfallene Funktionen

Keine.

3.5.2 Gekündigte Funktionen

3.5.2.1 POSIX-Optionen

Folgende POSIX-Optionen werden in Folgeversionen nicht mehr garantiert:

bisher	ab V4.0 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 l	-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 llpp=...	-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 Initialisierung von Arrays und Strukturen

Seit C/C++ V4.0 ist die Initialisierung von Arrays und Strukturen mit nicht-konstanten Werte nur im C2011-Modus möglich.

*4 **3.6.2 Abkürzung //MO-LI ist jetzt mehrdeutig**

*4 Die Abkürzung //MO-LI bzw. //MODIFY-LI ist nicht mehr eindeutig. Neben
*4 //MODIFY-LISTING-PROPERTIES gibt es jetzt auch //MODIFY-LIBRARY-
*4 VERSION.

3.7 Einschränkungen

3.7.1 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)

3.7.2 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 *V2 (Cfront) C++-Modus auf (//MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUS-PLUS(MODE=*V2) bzw. -X d).
Die folgenden drei Funktionspaare erhalten jeweils denselben externen Namen und führen daher beim Binden zu Duplikaten:

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

C/C++ 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.

3.7.3 CFE1079 bei Verwendung der Optionen IEEE bzw. ASCII

Die Verwendung der Übersetzungsoptionen FP-ARITMETICS = *IEEE (-K ieee_floats) bzw. LITERAL-ENCODING = *ASCII oder *ASCII-FULL (-K literal_encoding_ascii oder -K literal_encoding_as-cii_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

C/C++ V4.0 ist auf allen Business Servern einsetzbar, auf denen die vorausgesetzte Software freigegeben ist.