

FUJITSU Software BS2000 C/C++-RS

\*6  
\*12

Version 3.2E  
June 2018

Release Notice

All rights reserved, including intellectual property rights.  
Technical data subject to modifications and delivery subject to availability. Any liability that the data and illustrations are complete, actual or correct is excluded. Designations may be trademarks and/or copyrights of the respective manufacturer, the use of which by third parties for their own purposes may infringe the rights of such owner.

© 2018 Fujitsu Technology Solutions GmbH

Fujitsu and the Fujitsu logo are trademarks or registered trademarks of Fujitsu Limited in Japan and other countries. BS2000 is a trademark of Fujitsu Technology Solutions GmbH in Germany and other countries.

<b>1</b>	<b>General</b>	<b>4</b>
1.1	Ordering	4
1.2	Delivery	5
1.3	Documentation	5
<b>2</b>	<b>Software extensions</b>	<b>7</b>
2.1	Corrections of known errors	7
2.2	Controllable treatment of double Entries in POSIX	7
2.3	Generation of X86 code	7
2.4	Initialization	7
2.5	Listing generation	7
2.6	Switch optimization	7
<b>3</b>	<b>Technical information</b>	<b>8</b>
3.1	Resource requirements	8
3.2	Software configuration	8
3.3	Product installation	8
3.3.1	Installation and deinstallation	8
3.3.1.1	Public installation (SOLIS) for BS2000	9
3.3.1.1.1	Performing the installation	9
3.3.1.1.2	Preloadable subsystems	10
3.3.1.1.3	Use	10
3.3.1.1.4	Deinstallation	10
3.3.1.2	Automatic public installation (SOLIS) for POSIX	10
3.3.1.2.1	Performing the installation	11
3.3.1.2.2	Preloadable subsystem	11
3.3.1.2.3	Use	11
3.3.1.2.4	Deinstallation	11
3.3.1.3	Manual public installation for POSIX with IMON	11
3.3.1.3.1	Prerequisites	11
3.3.1.3.2	Performing the installation	11
3.3.1.3.3	Errors during the installation	12
3.3.1.3.4	Installation directory	12
3.3.1.3.5	Preloadable subsystem	12
3.3.1.3.6	Use	13
3.3.1.3.7	Deinstallation	13
3.3.1.4	Manual public installation for POSIX without IMON	13
3.3.1.4.1	Prerequisites	13
3.3.1.4.2	Performing the installation	13
3.3.1.4.3	Errors during installation	14
3.3.1.4.4	Installation directory	14
3.3.1.4.5	Preloadable subsystem	14
3.3.1.4.6	Use	14
3.3.1.4.7	Deinstallation	14
3.3.1.5	Private installation for BS2000	15
3.3.1.5.1	Prerequisites	15
3.3.1.5.2	Performing the installation	15
3.3.1.5.3	Errors during the installation	15
3.3.1.5.4	Use	15
3.3.1.5.5	Deinstallation	16
3.3.1.6	Private installation for POSIX	16
3.3.1.6.1	Prerequisites	16
3.3.1.6.2	Performing the installation	17
3.3.1.6.3	Errors during the installation	17
3.3.1.6.4	Preloadable subsystem	17
3.3.1.6.5	Use	17
3.3.1.6.6	Deinstallation	17
3.3.1.7	Installation of the II-Update tool	17
3.3.1.7.1	Prerequisites	18
3.3.1.7.2	Performing the modification	18
3.3.1.7.3	Errors during modification	18
3.3.1.7.4	Use	18

3.3.2	Use of other products	19
3.4	Product use	19
3.4.1	C++ source and object compatibility	19
3.4.2	Native code	19
3.5	Obsolete functions and those to be discontinued	20
3.5.1	Obsolete functions	20
3.5.1.1	Generation of RISC(MIPS) Code	20
3.5.2	Functions to be discontinued	20
3.5.2.1	POSIX options	20
3.6	Incompatibilities	21
3.6.1	LL Libraries	21
3.6.2	File names in II files	21
3.6.3	Combination ANSI-C++ and ASCII- or IEEE-Option	21
3.6.4	Important notes on CRTE	21
3.7	Restrictions	21
3.7.1	Restrictions with native code generation	21
3.7.2	Maximum name length	21
3.7.3	LISTING option COMMENTS is not supported	22
3.7.4	AID errors with constructors of length 0	22
3.7.5	Peculiarities in Cfront C++ language mode	22
3.7.6	CFE1079 when using options IEEE or ASCII	22
3.8	Procedure in the event of errors	23
<b>4</b>	<b>Hardware support</b>	<b>24</b>

# 1 General

C/C++-RS V3.2 is the follow-up version of C/C++-RS V3.1.

- \*11 C/C++-RS is only available for internal users; the generally released product variant is C/C++ V3.2.

The name of the delivery group and a component part of the product file names is CPP-RS.

C/C++ is the strategic BS2000/OSD compiler for developing and porting applications from the open world (e.g. OO applications) to BS2000 Business Servers.

- \*7 This Release Notice is a summary of the major extensions, requirements and operating information with regard to C/C++-RS V3.2E.

- \*12 The release level is that of June 2018.

- \*12 Changes to release level June 2017 are marked with “\*12”.

- \*11 Changes to release level June 2016 are marked with “\*11”.

- \*10 Changes to release level November 2015 are marked with “\*10”.

- \*9 Changes to release level May 2015 are marked with “\*9”.

- \*8 Changes to release level December 2014 are marked with “\*8”.

- \*7 Changes to release level June 2014 are marked with “\*7”.

- \*6 Changes to release level June 2013 are marked with “\*6”.

- \*5 Changes to release level June 2012 are marked with “\*5”.

- \*4 Changes to release level December 2011 are marked with “\*4”.

- \*3 Changes to release level December 2010 are marked with “\*3”.

- \*2 Changes to release level April 2009 are marked with “\*2”.

- \*2 Changes to release level November 2007 are marked with “\*1”.

- \*3 This and other Release Notice(s) are contained on the Softbooks DVD and are also available online at <http://manuals.ts.fujitsu.com/>.
- \*3

If one or more previous versions are skipped when this product version is used, the information from the Release Notices (and README files) of the previous versions must be noted.

## 1.1 Ordering

- \*11 C/C++-RS V3.2 is only available for internal users.

## 1.2 Delivery

The C/C++-RS V3.2 files are supplied via SOLIS.

The following files are delivered:

SINLIB.CPP-RS.032	Compiler library (POSIX)
SINPRC.CPP-RS.032	Library with private installation procedures
SYSGM.CPP-RS.032.D	Release Notice (German)
SYSGM.CPP-RS.032.E	Release Notice (English)
SYSLNK.CPP-RS.032	Compiler library (BS2000)
SYSTEMS.CPP-RS.032	Message file
SYSSDF.CPP-RS.032	SDF syntax file
SYSSDF.CPP-RS.032.IU.USER	SDF user syntax file for II-UPDATE
SYSSDF.CPP-RS.032.USER	SDF user syntax file for private installation
SYSSII.CPP-RS.032	IMON installation file
SYSSPR.CPP-RS.032.IU	SDF procedure for START-II-UPDATE
SYSSSC.CPP-RS.032.POSIX	Subsystem declaration (POSIX)
SYSSSC.CPP-RS.032	Subsystem declaration (BS2000)

The current file and volume characteristics are listed in the SOLIS2 delivery cover letter.

## 1.3 Documentation

\*4 The following descriptions are available for C/C++-RS V3.2:

German version	English version
----------------	-----------------

C language description:

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 manual (general part and SDF syntax):

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

Compiler manual (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 manual:

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

## C library functions:

- |    |   |   |
|----|---|---|
| *9 | C-Bibliotheksfunktionen<br>Referenzhandbuch | C Library Functions<br>Reference Manual |
| *9 | Stand November 2015                         | Edition November 2015                   |

## C library functions for POSIX:

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

## C++ library functions for ANSI-C++ mode:

- |   |   |
|---|---|
| - | Standard C++ Library V1.2<br>User's Guide and Reference |
|---|---|

## C++ library functions for Cfront mode:

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

## Tools.h++ class library (for ANSI-C++ mode):

- |   |                                   |
|---|-----------------------------------|
| - | Tools.h++ V7.0<br>User's Guide    |
| - | Tools.h++ V7.0<br>Class Reference |

The following additional documentation is also available that is not intended exclusively for C/C++ users:

## CRTE runtime system:

- |     |       |  |  |
|-----|-------|--|--|
| *11 | as of | CRTE V2.9A<br>Common Runtime Environment<br>Benutzerhandbuch | CRTE V2.9A<br>Common Runtime Environment<br>User Guide |
|-----|-------|--|--|

The manuals of the BS2000 basic configuration are additionally required for operating C/C++-RS.

- \*10 The BS2000 documentation is available in German and English on DVD under the  
\*7 title "BS2000/OSD Softbooks".

The documentation is also available on the internet at

- \*7 <http://manuals.ts.fujitsu.com>. Manuals which are displayed with an order number  
\*7 can also be ordered in printed form.

- \*6 For manuals there may be additional readme files. Information about changes and  
\*6 extensions not yet contained in the manuals can be found there. The README  
files are available on the SoftBooks-DVD or online at <http://manuals.ts.fujitsu.com/>.  
Readme files are named:

- |                           |           |
|---------------------------|-----------|
| SYSRME.CPP-RS.<version>.E | (English) |
| SYSRME.CPP-RS.<version>.D | (German)  |

## 2 Software extensions

The extensions and improvements over the previous version C/C++-RS V3.1 are described in the following section.

### 2.1 Corrections of known errors

C/C++-RS V3.2 contains a number of error corrections which should be used.

### 2.2 Controllable treatment of double Entries in POSIX

With C/C++-RS V3.2 it is possible to recognize during the production phase, if a program contains multiple entries and therefore it is not executable. The new Options '-z dup\_warning' and '-z dup\_error' control whether in such a case only a warning is issued or an error will be raised.

### 2.3 Generation of X86 code

\*10 With C/C++-RS as of V3.2B X86 code can be generated.

### 2.4 Initialization

\*2 In C/C++-RS as of V3.2C the initialization of arrays and structures with non-constant is supported.  
\*2

### 2.5 Listing generation

\*3 The list generation was reimplemented in C/C++-RS V3.2D. Many errors and imperfections have been corrected.  
\*3

\*3 Another improvement is that the list generation now works faster.

### 2.6 Switch optimization

\*6 C/C++-RS V3.2E contains a newly implemented switch optimization.

## 3 Technical information

### 3.1 Resource requirements

The following memory range is required in the system address space for running C/C++-RS:

at least 128 MB class 6 memory

This value represents the minimum requirement that may increase due to the amount of data involved and the application (e.g. when using templates in C++ sources).

- \*12 C/C++-RS V3.2E occupied after the load 2215 PAM pages in class 6 memory,
- \*10 with preloaded subsystem CPP-RS 950 PAM pages. When generating X86 code
- \*12 it requires an additional 1597 PAM pages for ASSTRAN.
  
- \*6 Compared to the previous version, the compiler memory requirement is reduced.
  
- \*3 The memory required for preloading the CPP-RS subsystem is approximately
- \*12 5,4 MB.
- \*3 The memory required for preloading the CPP-RSP subsystem is approximately
- \*12 5,2 MB.

### 3.2 Software configuration

- \*12 BS2000/OSD-BC as of V9.0 (S series Business Servers), OSD/XC as of V9.0 (SQ
- \*12 series Business Servers) or OSD/XC as of V9.5 (SE Server) is required for C/C++
- \*12 V3.2E.

Note:

- \*11 OSD V6.0B and CRTE-BASYS 1.6F were still enough for version 3.2E40.

- \*6 C/C++-RS V3.2E requires the following correction levels of software products:

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

- \*3 and the products: BINDER, BUILDER, CRTE, PLAM und SDF used in releases
- \*3 matching to the OSD version.

- \*3 Additional software used in releases matching to the OSD version is required for
- \*3 using particular functions:

- \*3 - BLSSERV for dynamic binding/loading
- \*3 - DSSM for preloading the compiler
- \*3 - LMS for private C/C++-RS installations
- \*3 - POSIX-HEADER for using the compiler in POSIX
- \*3 - POSIX-BC for using the compiler in POSIX

- \*11 The compiler CPP-RS V3.2E contains ASSTRAN V5.2B07.

### 3.3 Product installation

#### 3.3.1 Installation and deinstallation

- \*3
- \*3
- \*6 C/C++-RS V3.2 consists of the components (compiler, listing generator, II-update
- \*3 tool) with an SDF interface that can be used in the BS2000, and the POSIX com-
- \*3 ponents (compiler, listing generator, tools) which can be optionally installed for use
- \*3 under POSIX.



\*6 C/C++-RS V3.2 is normally installed in the BS2000 using SOLIS or IMON. If re-  
\*6 quired, the POSIX parts of C/C++-RS must be installed manually, unless IMON as  
\*3 of version 2.8 is used. With IMON as of version 2.8 the POSIX parts of the product  
\*3 can already be automatically installed by SOLIS.

\*3 All in all support is provided for the following installation types, which are de-  
\*3 scribed in more detail below:

- \*3 - Public installation (SOLIS) for BS2000
- \*3 - Automatic public installation (SOLIS) for POSIX
- \*3 - Manual public installation for POSIX with and without IMON
- \*3 - Private installation (scripts from SINPRC) for BS2000
- \*3 - Private installation (scripts from SINPRC) for POSIX

\*3 A public installation is normally available for all users of a system and requires ap-  
\*3 propriate administrator privileges for the installation, whereas a private installation  
\*3 is mostly only intended for the user performing the installation and does not require  
\*3 administrator privileges.

\*3 No update installation is supported. The old version must be deinstalled before a  
\*3 new version or a correction version is installed. This is also valid in case errors oc-  
\*3 cur during the installation.

\*3 The deinstallation of an older correction version (not an older version) is also pos-  
\*3 sible using a newer SINLIB of the same release unit and must therefore no longer  
\*3 be performed before the SOLIS volume is installed.

\*3 If you want to install several versions and/or correction versions in parallel, you  
\*3 must ensure that the product files for each installation are available separately.  
\*3 This means that, e.g. for versions in which only the correction version differs, they  
\*3 must be located under different user ids or that the name of the release items  
\*3 must be different (e.g. prefix). The installation directory for the POSIX part of the  
\*3 installation must be selected differently from every other installation.

### \*3 3.3.1.1 Public installation (SOLIS) for BS2000

\*3 This is the standard installation variant for the BS2000 part of the product and  
\*3 should be the recommended installation variant for most customers. In this regard,  
\*3 SOLIS automatically takes over all the tasks, such as the placing of the product  
\*3 files (in accordance with the installer), activation of syntax and message files and  
\*3 registration of the preloadable subsystems. The possibilities offered by SOLIS for  
\*3 the installation on any user ids and/or with modified file names are supported, as  
\*3 are parallel installations under different product versions.

#### \*3 3.3.1.1.1 Performing the installation

\*3 See the SOLIS/IMON documentation for the description of the SOLIS installation.

\*3 3.3.1.1.2 Preloadable subsystems

\*3

\*3 The compiler for the BS2000 and also the compiler for POSIX are available as  
\*3 preloadable subsystems. The relevant subsystem declarations have already been  
\*3 entered in the system catalog for the SOLIS installation. The subsystem names  
\*6 are: CPP-RS and CPP-RSP.

\*3

\*3 The respective system administration must decide which one of the subsystems is  
\*3 preloaded and whether this already takes place during the system start. However,  
\*3 preloading considerably reduces the load times when the compiler is called.

\*3

\*3 All of the above-named subsystems can theoretically be preloaded in parallel. On  
\*3 the other hand, parallel preloading of subsystems of the same name but of a dif-  
\*3 ferent version is not permitted.

\*3

\*3 3.3.1.1.3 Use

\*3

\*3 After successful installation with SOLIS, the compiler and the listing generator can  
\*3 be called via their start commands without any further action. On the other hand,  
\*3 an additional manual installation step, which is described in section 3.3.1.7, is al-  
\*3 most always required to use the II-Update tool.

\*3

\*3 However, you should ensure that only the product version or variant that was last  
\*3 installed publicly in the BS2000 is and can be the owner of the start commands  
\*3 START-CPLUS-COMPILER and START-CPLUS-LISTING-GENERATOR.

\*3

\*3 It is therefore not recommended for several product variants or versions to be in-  
\*3 stalled publically in parallel, but it is possible in principle. The other installations  
\*3 can then only be called using the START-PROGRAM or the START-  
\*3 EXECUTABLE-PROGRAM, e.g. with:

\*3

\*6 /START-EXE \*LIB(\$.SYSLNK.CPP-RS.032,SDFCC)

\*3

or

\*6 /START-EXE \*LIB(\$.SYSLNK.CPP-RS.032,SDFLISTGEN)

\*3

\*3 3.3.1.1.4 Deinstallation

\*3

\*3 Deinstallation of the product is also performed using SOLIS. However, it should be  
\*3 noted that a manually installed POSIX part of the product must also be deinstalled  
\*3 manually beforehand.

\*3

\*3 3.3.1.2 Automatic public installation (SOLIS) for POSIX

\*3

\*3 This installation variant is possible on systems with IMON as of V2.8 and should  
\*3 be the recommended installation variant there for most customers. The POSIX  
\*3 part of the installation takes place automatically during or after the normal  
\*3 SOLIS/IMON installation of the product and no further manual installation step is  
\*3 required.

\*3

\*3 This installation variant is simple and convenient, but has certain restrictions:

\*3

\*3 1. The installation path cannot be freely selected. The product is always installed  
\*3 in POSIX under the standard path /opt/C.

\*3

\*3 2. Before the installation all versions of the same release unit that have already  
\*3 been publically installed in POSIX are deinstalled, regardless of the version  
\*3 and the installation location. This installation variant does not permit any public  
\*3 parallel installations of different versions and/or correction versions.

\*3 3.3.1.2.1 Performing the installation

\*3

\*3 See the SOLIS/IMON documentation for the description of the SOLIS installation.  
 \*3 See the documentation for IMON V2.8 for how IMON is induced to perform the au-  
 \*3 tomatic POSIX installation.

\*3

\*3 3.3.1.2.2 Preloadable subsystem

\*3

\*3 A publically installed POSIX compiler can be preloaded as the subsystem by the  
 \*3 system administrator (see section 3.3.1.1.2), which reduces the load times when  
 \*3 starting the compiler. On the other hand, the compiler cannot be preloaded with  
 \*3 the POSIX loader posdbl.

\*3

\*3 3.3.1.2.3 Use

\*3

\*3 After the installation has been completed, the compiler and the listing generator  
 \*3 can be called using their POSIX commands (e.g. cc) without any further action.

\*3

\*3 3.3.1.2.4 Deinstallation

\*3

\*3 Deinstallation of the POSIX part also takes place automatically if another product  
 \*6 version of C/C++-RS is installed in this way or if the delivery unit is removed with  
 \*3 SOLIS/IMON from the system.

\*3

\*3 3.3.1.3 Manual public installation for POSIX with IMON

\*3

\*3 The installation is performed in accordance with the POSIX Basic Principles Man-  
 \*3 ual, in the section entitled Delivery and installation procedure for POSIX program  
 \*3 packages. This is the recommended installation variant for all systems with an  
 \*3 IMON version older than V2.8.

\*3

\*3 It is possible to choose the installation path in the POSIX file system, but it should  
 \*3 not be already occupied by other products or installations. As a result of selecting  
 \*3 an installation path the public parallel installation of different versions and/or cor-  
 \*6 rection versions of C/C++-RS is possible.

\*3

\*3 3.3.1.3.1 Prerequisites

\*3

\*3 The installation must be performed on a privileged basis under the system default  
 \*3 userid (mostly TSOS) and the release items must be installed with IMON, i.e. also  
 \*3 be registered. Ideally, this installation requires the prior public installation for  
 \*3 BS2000.

\*3

\*3 3.3.1.3.2 Performing the installation

\*3

\*3 The POSIX installation is started with

\*3

\*3 /START-POSIX-INSTALLATION

\*3 In the following mask it is necessary to select "Install packages on POSIX".

\*3

\*3 The following values can be entered in the next mask:

\*3

\*3 IMON: Y

\*3 Product: CPP-RS

\*3

\*3 Package: V03.2 (optional)

\*3

\*12 Correction version: E60 (or a different version, optional)

\*3 Specification of the version and/or correction version is only required if several  
 \*3 versions of the product are registered with IMON.  
 \*3  
 \*3 After the screen has been sent, the mask is shown again, and a field to enter the  
 \*3 required installation directory is now also shown. The field has the standard instal-  
 \*3 lation path default /opt/C.  
 \*3  
 \*3 The field can now be modified. This can make sense if you want to install several  
 \*3 versions and/or correction versions of a product in parallel. The installation scripts  
 \*3 check whether another product is already installed under this path, in which case  
 \*3 the installation is aborted.  
 \*3 The installation starts automatically as soon as the screen is sent.

\*3 3.3.1.3.3 Errors during the installation

\*3 As the output of installation scripts is not always displayed correctly by the POSIX  
 \*3 installation tool, a file is stored under /var/tmp/inst.<release\_unit> in the case of an  
 \*3 error, from which more exact information about the cause of the error can possibly  
 \*3 be provided. The POSIX installation tool then shows an error message, which refers  
 \*3 to a problem during the execution of the product-specific scripts, e.g.:  
 \*3 shell script ".../install\_pre" reports error 102.

\*3 However, the POSIX installation tool does not always abort the installation when  
 \*3 errors occur and the user must therefore by all means deinstall the incorrectly in-  
 \*3 stalled product before attempting another installation. In particular, a partially cre-  
 \*3 ated installation directory is not automatically deleted again if an error occurs.  
 \*3 The number specified in the error message provides a reference to the error that  
 \*3 has occurred, the actual error messages of the shell or other commands are found  
 \*3 in the error output file in /var/tmp.

Error number	Description
100	The POSIX installation tool used does not have the version required for the installation of C/C++-RS V3.2. An update of POSIX to the most current version is necessary.
101	The installation directory was not created by the POSIX installation tool (follow-up error).
102	The specified installation path is already being used by another product installation.
103	A POSIX installation that uses the same SINLIB already exists.

\*3 3.3.1.3.4 Installation directory

\*3 The installation in the POSIX file system is into the selected directory. Symbolic  
 \*3 links are also created for the commands to /usr/bin if and only if the standard in-  
 \*3 stallation path /opt/C was used for this installation.

\*3 3.3.1.3.5 Preloadable subsystem

\*3 A publically installed POSIX compiler can be preloaded as the subsystem by the  
 \*3 system administrator (see section 3.3.1.1.2), which reduces the load times when  
 \*3 starting the compiler. On the other hand, the compiler cannot be preloaded with  
 \*3 the POSIX loader posdbl.

\*3 3.3.1.3.6 Use

\*3

\*6 The C/C++-RS commands are accessible via /usr/bin if the installation was into  
 \*3 the standard path. This directory is entered in the standard search path of every  
 \*3 POSIX user, and should therefore not require any further provisions for this type of  
 \*6 installation in order to use C/C++-RS. The commands of other C/C++-RS installa-  
 \*3 tions can be achieved by specifying the explicit command paths for the call or by  
 \*3 entering the respective path <posix\_install\_path>/bin in the command search path  
 \*3 of the caller.

\*3

\*3 3.3.1.3.7 Deinstallation

\*3

\*3 Deinstallation of the POSIX part of a public installation is performed using the  
 \*3 POSIX installation command as TSOS by calling:

\*3

\*3 /START-POSIX-INSTALLATION

\*3

\*3 In the following mask it is necessary to select "Delete packages from POSIX". In  
 \*3 the next mask the required installation must be selected and marked on the basis  
 \*3 of the version and/or the installation directory. Once sent (DUE), it is necessary to  
 \*3 confirm again with DUE.

\*3

\*3 The product can then (and not before) be removed from the system with  
 \*3 SOLIS/IMON.

\*3

\*3 3.3.1.4 Manual public installation for POSIX without IMON

\*3

\*3 The installation is performed in accordance with the POSIX Basic Principles Man-  
 \*3 ual, in the section entitled Delivery and installation procedure for POSIX program  
 \*3 packages.

\*3

\*3 It is possible to choose the installation path in the POSIX file system, but it should  
 \*3 not be already occupied by other products or installations. As a result of selecting  
 \*3 an installation path the public parallel installation of different versions and/or cor-  
 \*6 rection versions of C/C++-RS is possible.

\*3

\*3 When installing without IMON, it is possible to install the same correction version  
 \*6 of C/C++-RS in public repeatedly (e.g. when configuring data center usersids),  
 \*3 which is not possible with IMON.

\*3

\*3 3.3.1.4.1 Prerequisites

\*3

\*3 The installation must be performed on a privileged basis under the system default  
 \*3 userid (mostly TSOS) and the release items may be installed on any userid.

\*3

\*3 3.3.1.4.2 Performing the installation

\*3

\*3 The POSIX installation program is started with

\*3

\*3 /START-POSIX-INSTALLATION

\*3

\*3 In the following mask it is necessary to select "Install packages on POSIX".  
 \*3 The following values can be entered in the next mask:

\*3

*3	IMON:	N
*3	Product:	CPP-RS
*3	Package:	
*3	Version:	032
*3	Userid:	<i>Name of the identifier with the release items</i>

\*3 After the screen has been sent, the mask is shown again, and a field to enter the  
\*3 required installation directory is now also shown. The field has the standard instal-  
\*3 lation path default /opt/C.

\*3 The field can now be modified. This can make sense if you want to install several  
\*3 versions and/or correction versions of a product in parallel. The installation scripts  
\*3 check whether another product is already installed under this path, in which case  
\*3 the installation is aborted.

\*3 The installation starts automatically as soon as the screen is sent.

#### \*3 3.3.1.4.3 Errors during installation

\*3 See section 3.3.1.3.3.

#### \*3 3.3.1.4.4 Installation directory

\*3 The installation in the POSIX file system is into the selected directory. Symbolic  
\*3 links are also created for the commands to /usr/bin if and only if the standard in-  
\*3 stallation path /opt/C was used for this installation.

#### \*3 3.3.1.4.5 Preloadable subsystem

\*3 A publically installed POSIX compiler can be preloaded as the subsystem by the  
\*3 system administrator (see section 3.3.1.1.2), which reduces the load times when  
\*3 starting the compiler. On the other hand, the compiler cannot be preloaded with  
\*3 the POSIX loader posdbl.

\*3 The subsystem declarations are not automatically entered in the system catalog  
\*3 for an installation that has not been performed with SOLIS/IMON. This would have  
\*3 to be done manually, but is not recommended for this installation type.

#### \*3 3.3.1.4.6 Use

\*3 The C/C++-RS commands are accessible via /usr/bin if the installation was into  
\*3 the standard path. This directory is entered in the standard search path of every  
\*3 POSIX user, and should therefore not require any further provisions for this type of  
\*3 installation in order to use C/C++-RS. The commands of other C/C++-RS installa-  
\*3 tions can be achieved by specifying the explicit command paths for the call or by  
\*3 entering the respective path <posix\_install\_path>/bin in the command search path  
\*3 of the caller.

#### \*3 3.3.1.4.7 Deinstallation

\*3 The deinstallation of the POSIX part of a public installation takes place with the  
\*3 POSIX installation command as TSOS by calling:

\*3 /START-POSIX-INSTALLATION

\*3 In the following mask it is necessary to select "Delete packages from POSIX". In  
\*3 the next mask the required installation must be selected and marked on the basis  
\*3 of the version and/or the installation directory. Once sent (DUE), it is necessary to  
\*3 confirm again with DUE.

\*3 The product can then (but not before) be removed from the system with  
\*3 SOLIS/IMON.

\*3 3.3.1.5 Private installation for BS2000

\*3

\*3 The installation is performed using procedures that are supplied with the  
\*6 C/C++-RS distribution.

\*3

\*3 3.3.1.5.1 Prerequisites

\*3

\*3 The installation can take place under any (non-privileged) userid.

\*3

\*3 The release items of C/C++-RS must be accessible on this or another userid for  
\*3 the caller, the caller must above all also have write authorization for the file SYS-  
\*3 LNK. The content of this file must also be in its original status and may not already  
\*3 have been modified by a prior private installation.

\*3

\*3 The release items that are absolutely necessary for a private installation are as fol-  
\*3 lows (specified here with their original names):

\*3

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

\*3

\*3 3.3.1.5.2 Performing the installation

\*3

\*3 If the product files exist under their original names on the caller's userid, the instal-  
\*3 lation can then be performed with the command

\*3

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

\*3

\*3 In principle, the release items can be randomly renamed if this is taken into con-  
\*3 sideration during subsequent use. If the product files exist under a different userid  
\*3 and/or under a different name, the installation must take place in the following  
\*3 general form:

\*3

\*3 /CALL-PROCEDURE (<sinprc\_name>,INSTALL.SDF), -  
\*3 / (SYSLNK=<syslnk\_name>,SYSMES=<sysmes\_name>, -  
\*3 / SINPRC=<sinprc\_name>)

\*3

\*3 The names of the release items can be normal file names with/without an userid  
\*3 and with/without a pubset specification.

\*3

\*3 3.3.1.5.3 Errors during the installation

\*3

\*3 If the installation is ended with an error, it is possible for the file SYSLNK to al-  
\*3 ready have been modified. It must be put back into the original status before an-  
\*3 other attempt at installation is made.

\*3

\*3 3.3.1.5.4 Use

\*3

\*3 After the installation has been completed, the compiler and listing generator can  
\*3 be used. It should be noted that the message catalog and the syntax file of the pri-  
\*3 vate installation must be activated at least once per session. Then the compiler or  
\*3 the listing generator can be called with START-PROGRAM or START-  
\*3 EXECUTABLE-PROGRAM.

\*6 For a private installation on the caller ID with the original names a compiler  
 \*6 invocation could be done as follows:

```
*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, -
*6 / R-M=A(SHARE-SCOPE=*NONE),P-M=A)
```

\*3 If the product files are available under a different userid and/or under a different  
 \*3 name, the call of the listing generator can e.g. take place in the following general  
 \*3 form:

```
*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 After the start, the compiler reports with the prompt for SDF statements, analog to  
 \*3 the start commands.

\*3 The option SHARE-SCOPE=\*NONE is necessary to ensure that a link is not made  
 \*3 to a preloaded subsystem of a different public installation with a suitable version,  
 \*3 but possibly with a different correction version.

\*3

### \*3 3.3.1.5.5 Deinstallation

\*3

\*3 Deinstallation of a private installation is done by simply deleting the release items.  
 \*3 If the same release items also form the basis of a POSIX installation, the product  
 \*3 must first be deinstalled in POSIX before the release items may be deleted.

\*3

### \*3 3.3.1.6 Private installation for POSIX

\*3

\*3 The installation is performed using procedures that are supplied with the  
 \*6 C/C++-RS distribution.

\*3

#### \*3 3.3.1.6.1 Prerequisites

\*3

\*3 The installation can take place under any (non-privileged) userid. This userid must  
 \*3 be entered as a POSIX user and the POSIX file system selected for the installa-  
 \*3 tion must have adequate free space.

\*3

\*6 The release items of C/C++-RS must be accessible on this or another userid for  
 \*3 the caller, the caller must above all also have write authorization for the file SIN-  
 \*3 LIB. The content of this file must also be in its original status and may not already  
 \*3 have been modified by a prior private installation.

\*3

\*3 The selected installation directory in the POSIX file system should not yet exist,  
 \*3 but the user performing the installation must have the authorization to create it.

\*3

\*3 The release items that are absolutely necessary for a private installation in POSIX  
 \*3 are as follows (specified here with their original names):

\*3

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



### \*3 3.3.1.6.2 Performing the installation

\*3  
\*3 If the product files exist under their original names on the caller's userid, the installation can then be performed with the command:

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

\*3 In principle, the release items can be randomly renamed if this is taken into consideration during subsequent use. If the product files exist under a different userid and/or under a different name, the installation must take place for the product variant CPP-RS in the following general form:

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

\*3 The names of the release items can be normal file names with/without an userid and with/without a pubset specification.

\*3 The specified installation path may not exist, but the caller must be authorized to create it. If the path name is a relative path name, it is created in the HOME directory of the user.

### \*3 3.3.1.6.3 Errors during the installation

\*3 If the installation is ended with an error, it is possible for the file SINLIB to already have been modified. It must be put back into the original status before another attempt at installation is made.

### \*3 3.3.1.6.4 Preloadable subsystem

\*3 The ability to preload is not planned for privately installed POSIX compilers. What is worse is that there is potential for conflict here. It is currently not possible in POSIX (analog to SHARE-SCOPE=\*NONE in SDF) to prevent the compiler from linking during the start with the preloaded subsystem of a different public installation with a suitable version, but possibly with a different correction version.

### \*3 3.3.1.6.5 Use

\*3 The C/C++-RS commands are accessible via <posix\_install\_path>/bin. The call can be achieved by specifying the explicit command paths or by entering the respective path <posix\_install\_path>/bin in the command search path of the caller.

### \*3 3.3.1.6.6 Deinstallation

\*3 Deinstallation of the POSIX part of a private installation is done by simply deleting the POSIX installation directory and the relevant release items in the BS2000, unless the latter are still being used for a private BS2000 installation.

### \*3 3.3.1.7 Installation of the II-Update tool

\*3 The II-Update tool that is supplied with the SDF compiler is not IMON-compliant and must therefore almost always be adapted to the actual compiler installation using a special procedure. This applies for all installation types, and the only exception is a compiler installation under the system default userid (mostly TSOS) while retaining the original name of the release items.

\*3 Special provisions are also required for the call.

\*3 3.3.1.7.1 Prerequisites

\*3

\*3 The modification should take place under the same userid, under which the  
\*6 C/C++-RS installation is to take place that is being modified.

\*3

\*6 The release items of C/C++-RS must be accessible on this or another userid for  
\*3 the caller, the caller must above all also have write authorization for the files SYS-  
\*3 SPR and SYSSDF.IU.USER. The contents of these files must also be in their orig-  
\*3 inal status and may not already have been changed by a prior modification.

\*3

\*6 The release items that are absolutely necessary for modification of the C/C++-RS  
\*3 installation are as follows (specified here with their original names):

\*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 Performing the modification

\*3

\*3 If the product files exist under their original names on the caller's userid, the modi-  
\*3 fication can then be performed with the command

\*3

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

\*3

\*3 In principle, the release items can be randomly renamed if this is taken into con-  
\*3 sideration during subsequent use. If the product files exist under a different userid  
\*3 and/or under a different name, the modification must take place in the following  
\*3 general form:

\*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 The names of the release items can be normal file names with/without a userid  
\*3 and with/without a subset specification.

\*3

\*3 3.3.1.7.3 Errors during modification

\*3

\*3 If the modification is ended with an error, it is possible for the file SYSSPR and/or  
\*3 SYSSDF.IU.USER to already have been modified. It must be put back into the  
\*3 original status before another attempt is made.

\*3

\*3 3.3.1.7.4 Use

\*3

\*3 After the modification has been completed, the II-Update tool can be used. It  
\*3 should be noted that the syntax file of the tool must be activated at least once per  
\*3 session. Then the start command can be called.

\*3

\*3 A tool call for an installation using the caller userid with the original names could  
\*3 be as follows:

\*3

\*6 /MOD-SDF-OPT \*ADD(SYSSDF.CPP-RS.032.IU.USER)  
\*6 /START-II-UPDATE <command\_parameter>

\*3

\*3 If the product files are available under a different userid and/or under a different  
\*3 name, the call of the tool can e.g. take place in the following form:

\*3

\*3 /MOD-SDF-OPT \*ADD(<sdfuser\_iu\_name>)  
\*3 /START-II-UPDATE <command\_parameter>

\*3

\*3 **3.3.2 Use of other products**

\*3  
 \*3 Other products are required on the appropriate system for the normal procedure of  
 \*3 the compiler. Regardless of the compiler installation type it determines the location  
 \*3 of the product files of the required products per IMON and if unsuccessful, it then  
 \*3 assumes a standard installation of the respective product under TSOS. If the  
 \*3 compiler cannot find the required external product files in this way or the required  
 \*3 minimum version is not available, the compiler cannot perform specific sub-  
 \*3 functions.

**3.4 Product use**

**3.4.1 C++ source and object compatibility**

The C++ language variant Cfront V3.1.3, which is the only one possible up to V2.2, is set as of V3.0 with language mode CPP (MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUSPLUS(MODE=CPP)) or in POSIX with -X d. Objects compiled in this way can be mixed in an application with C++ objects that were generated with C/C++ < V3.0.

The (more modern) ANSI mode is used as the default setting in V3.0 and later versions. Cfront mode cannot be mixed with ANSI modes: Cfront C++ Objects cannot be linked together with ANSI-C++ objects!

**3.4.2 Native code**

\*10 With code generation DESTINATION-CODE=\*X86 (or -K x86), native object code  
 \*10 is generated. This will only run on SQ servers and x86 Server Unit of SE Servers.  
 \*10 The generated object code is more efficient since it does not have to be emulated.

\*10 This increase in efficiency is only evident in real time and not in the 'standardized  
 CPU time' that is set as default on most SQ servers. The reason for this is that  
 standardized CPU time weights CPU usage for /390 and native code differently to  
 enable standardized accounting for required elementary operations, that are al-  
 ways based on /390 operations.

\*1 Note that native code is considerably more extensive than /390 code.  
 \*10 X86 native code is approximately 3 times as large as /390 code.  
 \*10 The C runtime system may have to be added to this, if it is hard-linked  
 into the native code.

\*1 X86 coded C runtime systems to be loaded dynamically  
 \*1 (SKULNK.CRTE.PARTIAL-BIND and SKULNK.CRTE.COMPL) are too large  
 to be loaded into the address space below 16 MB.

\*10 The translation times for X86 code is longer than for /390.

\*10 Please note that compared to /390 code generation the user address space has  
 to be increased by at least factor 3 for X86 code generation.

## 3.5 Obsolete functions and those to be discontinued

### 3.5.1 Obsolete functions

#### 3.5.1.1 Generation of RISC(MIPS) Code

As announced before V3.1 was the last version to support the generation of RISC (MIPS) code. The options

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

and

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

or

-K risc\_4000

therefore will not be offered any more in version 3.2.

### 3.5.2 Functions to be discontinued

#### 3.5.2.1 POSIX options

The following POSIX options will no longer be guaranteed in later versions:

Up to V2.2	as of V3.0 replaced by
-F R<=...>	-F loopunroll<,...>
-Ol<...>	-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 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 Incompatibilities

### 3.6.1 LL Libraries

The Long-Long Libraries (Suffix .LL) are no more needed for the binding. They are no longer delivered with C/C++-RS and the references have to be removed from the binding procedures.

### 3.6.2 File names in II files

For ANSI C++ sources containing templates the compiler generates II files storing (among other information) file names (e.g. the source library). When renaming such files or moving them to other pubsets or user-ids the ii file has to be updated. This can be done with the tool II-UPDATE.

As of version 3.1B the file names in the II file need not to be exactly the same as specified in the COMPILE statement (they may have been completed by adding missing Cat-Ids and User-Ids). This has to be taken into account when using II-UPDATE. If required the names in an II file can be displayed by use of the new SHOW function of II-UPDATE (for details see the user guide manual).

### 3.6.3 Combination ANSI-C++ and ASCII- or IEEE-Option

C++ Library Functions do not support the ASCII or the IEEE format. Since V3.2 such a combination is already detected at translation time and is rejected as an error with the new messages CFE2075 (ASCII) and CFE2077 (IEEE). As there are also cases where the combination is harmless, it is allowed to classify down the error messages as warnings (with Option MODIFY-DIAGNOSTIC-PROPERTIES or -R ...).

### 3.6.4 Important notes on CRTE

When producing ANSI-C++ applications incompatibilities can occur if the application uses standard C++ libraries (SYSLNK.CRTE.STDCPP) from different CRTE versions. The so-called associative containers (set, multiset, map and multimap classes) of CRTE as of V2.2A15 are incompatible to those of older CRTE versions. (Older CRTE versions (< V2.2A15) were delivered only until 1999).

If necessary, the application may have to be regenerated (compiled and linked) with the new CRTE version. (Recompilation is required to ensure that the headers and objects are from CRTE as of V2.2A15 and therefore compatible to each other.)

If attention is not paid to this incompatibility, results that cannot be generally forecast will occur: the incorrectly generated program may, for example, crash during startup or go into an endless loop.

## 3.7 Restrictions

### 3.7.1 Restrictions with native code generation

\*10 The following specific restrictions exist when generating native code (-K x86 or  
\*10 MODIFY-MODULE-PROPERTIES DESTINATION-CODE=\*X86)

### 3.7.2 Maximum name length

Entry names longer than 4096 characters can lead to errors (e.g. compilation abort with dump). It is not possible to name a length limit for user names since entry names are not directly assigned by the user, but are instead generated from the name selected by the user (and thereby increase in length).

The generated names can be displayed in the project list (MODIFY-LIST-PROPERTIES PROJECT-INFORMATION=\*YES or -N project) if required (the original names may thereby be shortened).

The following points are independent of the generated code:

### 3.7.3 LISTING option COMMENTS is not supported

The listing option MODIFY-LISTING-PROPERTIES PREPROCESSING-RESULT = \*YES(COMMENTS=\*NO) is not supported at the moment, i.e. comments will always be included in the preprocessor listing.

### 3.7.4 AID errors with constructors of length 0

Empty constructors in C++ sources may lead to AID errors in case of debugging:  
AID0252 AID error in module 56 : RTC 0E (CMD: TRACE)

### 3.7.5 Peculiarities in Cfront C++ language mode

For each function in C++, an external name is generated that also contains the parameter types in encrypted form. Unfortunately, errors occurred in V2.2 that are now also incorporated in the present version for compatibility reasons. They only occur in Cfront C++ mode (MODIFY-SOURCE-PROPERTIES LANGUAGE=CPLUS-PLUS(MODE=CPP) or -Xd).

The following three function pairs each receive the same external name and therefore lead to duplicates during binding or (if native code is generated) may lead to errors when the finished program is executed:

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

C/C++ V2.2 did not know signed and both functions are therefore mapped to the same name. This only affects 'signed char'.

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

The same names since the array dimensions are not considered.

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

if, for example, c was declared as 'typedef char c;': a const qualifier on a typedef remains invisible in the external name and both functions therefore receive the same name.

### 3.7.6 CFE1079 when using options IEEE or ASCII

Using the compile options FP-ARITHMETICS=\*IEEE (-K ieee\_floats) or LITERAL-ENCODING=\*ASCII/\*ASCII-FULL (-K literal\_encoding\_ascii / -K literal\_encoding\_ascii\_full) may lead to an error 'CFE1079 ERROR ...: expected a type specifier' if the requirements described in the compiler user guide (C library functions must not be declared explicitly in the source but only by including the corresponding CRTE header) have not been met.

### 3.8 Procedure in the event of errors

Following error documentation is required for diagnostic purposes in the event of errors:

- brief description of error situation
- description how and if the error is reproducible
- options-, source-, error list with expanded user includes (LISTING-Option)
- runtime log (MSG=FH)
- preprocessor output/source
- object list
- binder list
- input/output files
- expected results
- dump file (if a dump occurred)
- brief description of run

## 4 Hardware support

- \*10 C/C++-RS V3.2E may be used on all business servers fulfilling the software requirements.