

C/C++ (BS2000/OSD) Version 3.2

Issue May 2007

Pages 2

The C/C++ compiler supports object-oriented programming (OOP) with C++ on BS2000/OSD business servers.

Product Characteristics

Depending on the selected language mode, the C/C++ V3.2 compiler in BS2000/OSD supports:

- C code conforming to the Kernighan & Ritchie C definition,
- C code conforming to the ANSI/ISO C Standard 9899:1990 with Amendment 1:1994(E),
- C++ code conforming to the Stroustrup C++ definition (C/C++ V3.0B or later),
- C++ code conforming to the ANSI/ISO C++ Draft.

C++ is a powerful programming language which supports the principles of object-oriented programming, such as encapsulation, inheritance and polymorphism. C++ is particularly suitable for use in the development of reusable software building blocks in the form of class libraries.

C/C++ permits selective use of the advantages of object-oriented programming. The C language set is also available.

The C/C++ V3.2 programming system is available in the following selectable units:

C/C++:

Full configuration for OSD /390:

- with AID support,
- with /390 code generator,
- without RISC code generator.

The language set of the C++ compiler supports the following elementary functions for object-oriented programming:

- Templates
- Exception handling
- Run-time type information
- New-style casts
- Abstract data types
- Hidden information
- Classes
- Overloading
- Multiple inheritance.

C/C++-RS (shipment via special release only):

Full configuration for business servers with and SPARC processor architecture:

- with AID support,
- with /390 code generator,
- with RISC code generators for business servers with RISC processor architecture (SX-series).

C/C++ V3.2 supports the POSIX functionality and the POSIX file system in BS2000/OSD.

CRTE is the common runtime environment for C/C++, COBOL85 and COBOL2000. CRTE is a software requirement

for use of the C/C++ compiler and for running C/C++ applications.

CRTE is not shipped in combination with the C/C++ compiler and must be ordered separately.

Functional Description

C++ supports object-oriented programming, which is based on the following principles:

Encapsulation:

Objects encapsulate states and functions. In C++, objects are described by means of class definitions. A class definition collectively defines data and the functions that operate on this data.

Software produced according to this principle is more robust, easier to maintain and easier to extend, since there are fewer dependencies between the modules and the details of the implementation are encapsulated in classes.

Inheritance:

Classes can inherit attributes from other classes. Inheritance permits better structuring of the software and helps reduce the amount of code, as common sections of code can be reused.

Polymorphism:

Objects of different types can share a common function interface, enabling a developer to use the various objects without needing to know their type. The use of polymorphism produces software that is more general-purpose, more flexible and more reusable.

C++ supports the creation of class libraries. Class libraries are reusable software building blocks.

C++ avoids runtime errors by strict type checking. This greatly improves the stability of the programs.

C/C++ supports the POSIX functionality in BS2000/OSD-BC V2.0 or higher.

The C/C++ compiler can be invoked under the POSIX shell in accordance with XPG4 specifications. Sources and includes can be read from the POSIX file system UFS. Generated objects and compiler listings can be stored in UFS.

Input/output operations on UFS files are possible via the C/C++ POSIX RTS. This is particularly useful for processing unstructured data streams, which are a common feature in UNIX-systems environments.

AID can be used for symbolic and non-symbolic debugging of C++ programs in BS2000/OSD.

This provides the same test environment for C/C++ that many developers are familiar with from ASSEMBH, COBOL85, COBOL2000, FORTRAN or PLI1.

Program Description

The C/C++ development system comprises the C/C++ compiler and the Common Runtime Environment (CRTE).

The C/C++ compiler supports the language set of the ANSI/ISO C++ Draft 1996 as defined in the "Working Paper for Draft Proposed International Standard for Information Systems Programming Language C++, Doc.No.: X3J16/96-0219R1 or ISO WG21/N1037 dated 2.12.1996".

This includes templates, exception handling, new-style casts, namespaces and run-time type information (RTTI).

The C language set as defined by Kernighan & Ritchie and C ANSI/ISO incl. Amendment 1 is also supported.

Code is generated directly as machine code for business servers with /390 instruction architecture.

The generated /390 format guarantees object compatibility for the execution of BS2000/OSD customer applications even in the event of future changes in architecture.

CRTE includes language-specific and language-neutral libraries, e.g. for program linking, mathematics, standardized event and error handling, as well as for storage and I/O management.

The header files for the C and C++ library functions are also included in CRTE. Some CRTE libraries are shareable and can be preloaded as a subsystem.

With CRTE the standard C++ library conforming to the ANSI C++ Draft and the Tools.h++ © Rogue Wave library are also shipped.

The standard C++ library includes a string class, container classes, iterators, generic algorithms, numeric classes and operations, as well as input/output classes.

The Tools.h++ © library contains general-purpose "Foundation Classes". These include string classes with pattern matching mechanisms, classes for handling date and time, virtual streams, container classes and internationalization classes.

The following functions can be used in BS2000/OSD versions in which a POSIX subsystem is available:

- Input/output of POSIX files during compilation
 - Use of POSIX library functions conforming to XPG4
 - Control of the C/C++ compiler via the POSIX shell
- CRTE is required as the runtime environment for the C/C++ compiler and programs generated with it. CRTE is the common runtime environment for C/C++, COBOL85 and COBOL2000 programs.

Modules of the C/C++ runtime systems and the ILCS are additionally available as s as SPARC code for the SX series servers.

Technical data	
TECHNICAL REQUIREMENTS HARDWARE	Business servers of the S series (/390 architecture) Business servers of the SX series (SPARC architecture)
TECHNICAL REQUIREMENTS SOFTWARE	BS2000/OSD-BC V6.0 or higher for S series servers OSD/XC V2.0 or higher for SX series servers CRTE V2.6 or higher EDT V16.6 or higher SDF V4.1 or higher BINDER V2.3 or higher BUILDER V1.0 or higher LLMAM V3.2 or higher PLAM ab V3.1 or higher Optional software: AID for symbolic debugging POSIX-BC for the C/C++ compiler under POSIX DAB to speed up load times
OPERATING MODE	Batch and interactive dialog
IMPLEMENTATION LANGUAGE	C++, C, SPL4 and Assembler
USER INTERFACE	Commands (English) Messages (English or German)
INSTALLATION	Refer to the relevant release notices.
DOCUMENTATION	Documentation in English and German: C/C++ Compiler User Guide POSIX Commands of the C/C++ Compiler User Guide CRTE User Guide AID Debugger for C/C++ User Guide BS2000/OSD C Libraries Reference Manual POSIX C Libraries Reference Manual BS2000/OSD C++ Libraries Reference Manual Documentation in English only: Standard C++ Library User Guide and Reference Tools.h++ © User Guide Tools.h++ © Class Reference Tools.h++ Copyright © Rogue Wave Software, Inc.
USER REQUIREMENTS	Knowledge of the C++ and C programming languages and BS2000/OSD
TRAINING	See course offer at: http://www.fujitsu-siemens.com/training
CONDITIONS	This software product is supplied to the customer against a single payment or instalments in accordance with our conditions for the use of software products.
WARRANTY	Class: A Delivery format: Machine language; source code for C++ libraries
ORDERING AND DELIVERY	This software product may be obtained from your local Fujitsu Siemens Computers regional office.