FUJITSU Software BS2000 COBOL2000-BC and FUJITSU Software BS2000 COBOL2000

Version V1.6A
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.

1 General

COBOL2000, the COBOL compiler for object-oriented programming in BS2000, enables programming to the COBOL Standard ANSI X3.23-1985 (= ISO/IEC 1989-1985 = EN 21989 = DIN 66028-1986).

COBOL2000 V1.6A supports the supplement to the ANSI Standard X3.23a-1989 and ISO/IEC 1989 Amendment 1, Intrinsic Function Module as well as the COBOL Standard X3.23-1985 High Level.

The optional modules "Report Writer" and "Segmentation" are provided in their full functionality in addition to the mandatory modules required by the standard.

COBOL2000 also contains the "Data Manipulation Language DML" for operating the UDS database system.

COBOL2000 V1.6A supports the COBOL85 Standard as well as the main object-oriented functions and basic functionality from the COBOL2002 Standard (= ISO/IEC 1989:2002(E) Programming Language COBOL).

In addition to the language extensions, COBOL2000 V1.6A also supports the POSIX and XPG4 programming interfaces.

The COBOL2000 V1.6A compiler is supplied without the runtime system. The common runtime system for COBOL85, COBOL2000, C and C++ in BS2000 is the Common Runtime Environment CRTE.

COBOL2000 V1.6A is available in two configurations:

- Basic configuration (BC)
- Full configuration

The following functions are not supported in the basic configuration COBOL2000-BC V1.6A:

- REPORT WRITER
- CODASYL-DML (native UDS support)
- Symbolic debugging with AID
- Output of a list of all error messages
- POSIX interfaces
- Starter phase

This Release Notice is a summary of the major extensions, requirements and operating information with respect to COBOL2000 V1.6A under the BS2000 operating system.

This Release Notice describes the full configuration unless otherwise stated.

The release level is that of June 2018.

This and other current Release Notices are shipped on the SoftBooks DVD and are available online at http://manuals.ts.fujitsu.com/.

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 also be noted.

1.1 Ordering

COBOL2000 V1.6A can be ordered from your local distributors and is subject to the general terms and conditions of the software product use and service agreement.

1.2 Delivery

The COBOL2000 V1.6A files are supplied via SOLIS.

1.2.1 Delivery components of the basic configuration

The following delivery components are required regardless of the hardware type (HSI):

SYSFGM.COBOL2000-GEM.016.D	Release Notice (German)
SYSFGM.COBOL2000-GEM.016.E	Release Notice (English)
SYSMES.COBOL2000-BC.016	Message file (MSGMAKER format)
SYSPRG.COBOL2000-BC.016	Compiler
SYSSDF.COBOL2000-BC.016	System syntax file
SYSSDF.COBOL2000-BC.016.USER	User syntax file
SYSSII.COBOL2000-BC.016	IMON information file
SYSSII.COBOL2000-GEM.016	IMON information file for common delivery components
SYSSPR.COBOL2000-BC.016.COMPIL	SDF start procedure for compiler

1.2.2 Delivery components of the full configuration

The following delivery components are required regardless of the hardware type (HSI):

SINLIB.COBOL2000.016	Library for installation in POSIX
SYSFGM.COBOL2000-GEM.016.D	Release Notice (German)
SYSFGM.COBOL2000-GEM.016.E	Release Notice (English)
SYSMES.COBOL2000.016	Message file (MSGMAKER format)
SYSPRG.COBOL2000.016	Compiler
SYSPRG.COBOL2000.016.START	Starter phase, installed under the file name COBOL2000
SYSSDF.COBOL2000.016	System syntax file
SYSSDF.COBOL2000.016.USER	User syntax file
SYSSII.COBOL2000.016	IMON information file
SYSSII.COBOL2000-GEM.016	IMON information file for common delivery components
SYSSPR.COBOL2000.016.COMPIL	SDF start procedure for compiler

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

1.3 Documentation

The following documentation is available for COBOL2000 V1.6:

German:	English:
COBOL2000 V1.6 COBOL Compiler Sprachbeschreibung	COBOL2000 V1.6 COBOL Compiler Reference Manual
COBOL2000 V1.6 COBOL-Compiler Benutzerhandbuch	COBOL2000 V1.6 COBOL Compiler User Guide

The BS2000 documentation is available in German and English on DVD with the title BS2000 SoftBooks.

The documentation is also available on the internet at http://manuals.ts.fujitsu.com. Manuals which are displayed with an order number can also be ordered in printed form.

The manuals may be supplemented with README files. These contain changes and extensions to the manual of the product concerned. The file names are made up as follows:

SYSRME.<product>.<version>.E	(file with English text)
SYSRME.<product>.<version>.D	(file with German text)

The README files are available on the SoftBooks-DVD or online under http://manuals.ts.fujitsu.com.

2 Software extensions

Only the extensions and improvements over the previous version COBOL2000 V1.5 are described in the following section.

2.1 Processing of XML documents

COBOL2000 as of version 1.5A supports the reading and analysis of XML documents.

As of COBOL2000 V1.6A, XML data can be generated.

2.1.1 Statement XML GENERATE

As of COBOL2000 V1.6A, data can be generated in XML format. This is done with the XML GENERATE statement.

For a description of the instruction, see the COBOL2000 manuals.

2.1.2 SOURCE-PROPERTIES Option KEEP-XML-NAMES

This option determines whether the XML element names are converted to uppercase or the original names are retained when generating XML data.

The option was realized as a COMOPT option:

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

and as a SDF option:

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

For a description of the option, see the COBOL2000 manuals.

3 Technical information

3.1 Resource requirements

The files supplied with the product occupy

- approximately 2000 PAM pages for the basic configuration
- approximately 5600 PAM pages for the full configuration

If the POSIX file system is not used with the full configuration, the library SINLIB.COBOL2000.016 can be deleted. The Software Configuration Inventory (SCI) is thereby not adapted.

50 MB virtual address space should be made available to the compiler for compilation. This value represents a minimum requirement that may increase depending on the amount of data and the application used.

If the POSIX functions are used (ENABLE-UFS-ACCESS=YES), this increases the space required by the linked programs. The space requirement can be reduced again by linking with the CRTE library SYSLNK.CRTE.PARTIAL-BIND and preloading the CRTESIS subsystem, a component of CRTE.

3.2 Software configuration

BS2000 as of version 10.0 is required for COBOL2000 V1.6A, i.e. OSD/BC as of V10.0 or OSD/XC as of V10.0.

For S servers you require the following delivery components:

-	BS2GA.LLMAM	as of V10.0
-	BS2GA.SDF	as of V10.0
-	BS2GA.CRTE-BAS	as of V10.0

If no explicit SOLIS change release was specified, the correction levels of correction package 1/2018 are required.

The product "Common Runtime Environment" CRTE as of V10.0B or as of V11.0B must be installed. CRTE is an independent delivery unit and must be ordered separately for S servers.

For SQ servers and x86-based Server Units of BS2000 SE series you require the following delivery components:

-	BS2GA.BS2OSD	as of V10.0
-	BS2GA.CRTE-BAS	as of V10.0

If no explicit SOLIS change release was specified, the correction levels of correction package 1/2018 are required.

The POSIX subsystem must be loaded to enable the compiler to use the POSIX file system.

openNet Server as of V3.6 is required for support of Unicode (NATIONAL).

The following products are required for compilation:

•	CRTE	as V10.0B or as of V11.0B
•	JV (1)	as of V15.1
•	UDS/SQL (1)	as of V2.7
•	XHCS-SYS (2)	as of V2.2

(1) only if the relevant COBOL language resources/system services are used
(2) only if the data type NATIONAL or XML language are used

For run of COBOL2000 applications, the following products are required:

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

(1) only if the relevant COBOL language resources/system services are used
(2) only if LINE-SEQUENTIAL I/O is used
(3) only if COBOL modules are to run as UTM subroutines
(4) only if the data type NATIONAL or XML language are used

To run multiple versions of the COBOL2000 compiler concurrently, COBOL2000 as of V1.0B is required.

3.3 Product installation

Installation of the product COBOL2000 with the installation monitor IMON is mandatory. The information concerning installation in the delivery cover letter and in the product documentation must be followed as well as the information given below.

The necessary inputs and the sequence of the installation are described in the IMON documentation.

3.3.1 Installation for use in POSIX

Note: this point does not apply for COBOL2000-BC.

Proceed as follows if you also want to install COBOL2000 in the POSIX file system:

- If a version of the product COBOL85 is already installed in POSIX, this must first be deinstalled in all cases. To do this, call the POSIX installation program under TSOS (refer to the "POSIX Basics" manual for details). Select 'Delete packages from POSIX', enter COBOL85 as the product.

- A previous version of the COBOL2000 product already installed in POSIX must also be uninstalled beforehand. Proceed analogously for this.

- You can then install COBOL2000 in the POSIX file system either
  - directly with IMON (select POSIX installation) or alternatively
  - after successful IMON installation (without POSIX installation), with the POSIX installation tool. To do this, after starting the POSIX installation tool, select "Install packages on POSIX" and then enter COBOL2000 as the product and 016 as the product version.

3.4 Product use

3.4.1 Product use with the ESQL-COBOL precompiler

In addition to the information in the Release Notice for the product ESQL-COBOL, the following points must be observed when using the precompiler ESQL-COBOL for SESAM:

- ESQL-COBOL as of V3.0B also supports programs in COBOL Free Reference Format.

- Because the ESQL precompiler precedes the COBOL-Compiler, the conditional compilation as expressed in the COBOL language (such as ">> EVALUATE ..." or ">> IF ...") does not have any effect on the precompiler. The precompiler processes the ESQL language elements always, independent of the flow of the COBOL conditional compilation. This different interpretation of the same program text by the precompiler and the COBOL compiler does not necessarily result in error messages. Therefore, SQL language elements as well as declare sections should not be contained in conditional compiled regions.

- Syntactic restrictions with respect to mixing ESQL and new program structures for object orientation with recursive programs and associated local data areas are not known. In order to rule out problems with different visibilities of data and limited object lifetimes, we recommend that no SQL declare sections be defined in the local storage section and that SQL statements are only used in classic COBOL programs, but not in classes and methods.

The ESQL precompiler considers a compilation group to be a source program. Using SQL language resources in several compilation units leads to problems in the visibility of the referenced data.

3.4.2 Product use with UTM

The following points must be observed, in addition to the information given in the Release Notice for the UTM product:

- The subprograms produced using the COBOL2000 compiler can be generated with COMP=ILCS or COMP=COB1 for UTM, in the same way as the subprograms generated using the COBOL85 compiler. COMP=COB1 must be used when subprograms which are not ILCS compliant are called from the COBOL2000 module. If this is not the case, COMP=ILCS should be used.

- The programs compiled using the COBOL2000 compiler cannot use TCB entries, regardless of whether they are generated with COMP=COB1 or COMP=ILCS for UTM.

- The lifetime of the instance objects is limited to one subprogram run. When PEND is called, all new objects created within a subprogram run are released by the runtime system. This also applies to PEND variants without process change. This is why object references must not be retained beyond a subprogram run (in UTM memory areas, in shared memory or in a database), in order to pass them to subsequent subprograms. Object references are only valid in the subprogram run in which the object was created. Object references cannot be used beyond task boundaries in another program.

- Program replacement of LLMs which contain COBOL2000 modules with class definitions, require replacement of these LLMs together with all the LLMs that inherit or use the relevant class definitions

3.4.3 Product use with COLUMBUS85

The following points must be observed, in addition to the information given in the Release Notice for the COLUMBUS85 product:

- COLUMBUS85 has no direct interface to the COBOL compiler. The COBOL programs processed with COLUMBUS85 tools can still generally be compiled with the COBOL2000 compiler. COLUMBUS85 only supports the COBOL85 language standard, as does the COBOL85 compiler.

- It is therefore not possible to continue developing COBOL programs using the new language resources of the COBOL2000 compiler while still using the COLUMBUS85 tools for these programs. The COLUMBUS85 tools react with error messages or faulty interpretations if language resources are used in the COBOL programs that cannot be compiled with the COBOL85 compiler. Among other things, these include:

- the new COBOL2000 compilation units (e.g. class definitions, program prototypes)
- violations against the source format rules (division into A area and B area or free format)
- all directives
- the new COBOL2000 keywords

3.4.4 Product use with AID

The following points should be observed at the start of a debug session when debugging with AID:

Immediately after loading a COBOL program, no initialization has taken place. Single commands can therefore cause error messages or additional qualifications may have to be entered.

Particularly with object-oriented programming, the classes and objects have not been initialized. AID can therefore not handle data and test points in objects and methods or object references correctly in all cases in the calling programs. This problem can be avoided by executing

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

before other AID commands to activate the AID run trace and start the program. The program is thereby initialized and proceeds to the first statement.

When displaying BASED data in recursive programs with:

```
/%SD
```

the contents of the data from the innermost and most current recursion is always shown. The contents of the outer recursions must be determined in another manner, e.g. by stopping the program at a suitable point and outputting the required data.

3.5 Obsolete functions (and those to be discontinued)

No functions have been deleted between COBOL2000 V1.5 and COBOL2000 V1.6A.

The following flagging options are supported for the last time in this version:

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

This also applies for the respective ACTIVATE-FLAGGING options. If these are used, an appropriate informative message is output.

3.6 Incompatibilities

COBOL2000 V1.6A is fully compatible with the previous version COBOL2000 V1.5.

COBOL2000 V1.5 is fully compatible with the previous version COBOL2000 V1.4, apart from the following restrictions.

Incompatibilities in the language scope:

- If XML is used, (COBRUN ENABLE-XML-PROCESSING=YES) the following new keywords are reserved:
  DOCUMENT, END-OPEN, END-XML, IDENTIFIED, VERSION-XML, XML

- If XML is used, the following new special registers are reserved:
  XML-EVENT, XML-CODE, XML-TEXT, XML-NTEXT, XML-NAMESPACE, XML-NNAMESPACE, XML-NAMESPACE-PREFIX, XML-NNAMESPACE-PREFIX.

  If the user has defined such data names, these definitions will take precedence without comment.

- ASSIGN TO data name
  This clause is in the future to write as "ASSIGN USING data name".

Incompatibilities at runtime

- If USAGE NATIONAL Data are used, the XHCS-adapter module GNLADPT must be bind or resolved, except use of partial bind where the adapter is already included.

3.7 Restrictions

The following restrictions that applied for COBOL2000 V1.6A are to note:

- Alias Catalog System (ACS)
  Standard file names (xxxLST/FIL.COBOL.<prog-id>) must not be entered as alias file names because ACS alias names with and without prefixed user ID are not taken to be identical.

- Prelinked large modules or link-load modules that contain older COBOL85 runtime modules are not supported by the current runtime system. Relevant large modules must be re-linked. However, this does not constitute a restriction in upwards compatibility to old COBOL85 object modules.

  Note: applications must always be linked with only a single CRTE.

- In rare cases, with garbage collection for OO programs, the garbage collection routines themselves may not get any more memory from the system. In this case, the program is aborted with a corresponding error message.

3.8 Procedure in the event of errors

If an error occurs, the following error documentation will be required for diagnostic purposes:

- a detailed description of the error condition
- indication as to whether and how the error can be reproduced.
- options, source and error listing including expansion of the COPY elements (LISTING option)
- execution log (MSG=FH)
- source including the COPY elements and COSSD if required (COBOL subschema directory)
- linker listing
- input/output files
- expected result
- brief description of execution
- product version number
- Rep files used
- DUMP, if available
- subsystems used

The error must be reported to the appropriate service provider.

*1 A PERLE error report will be opened for Second Level Support.

A Teleservice connection with callback option is essential for diagnostics. If this is not available, the service provider is entitled to invoice additional services rendered.

4 Hardware requirements

COBOL2000 V1.6A is executable on all

- S series Business Servers (/390 architecture)
- SQ series Business Servers (x86 architecture)
- SE series Business Servers (/390 and x86 architecture)

that are supported by the operating system versions listed in section 3.2.

5 Firmware levels

not applicable