*4

Fujitsu Technology Solutions

COBOL85-BC, COBOL85 und COBOL85-R (BS2000/OSD)

Version 2.3A
*5 June 2012

Release Notice

Inhalt

1   General

*5   This Release Notice is a summary of the major extensions, dependencies
*5   and operating information with respect to COBOL85 V2.3A under the
*5   BS2000 operating system.
*5
*5   The release level is that of: June 2012.
*5
*5   Changes to release level October 2000 are marked with *4.
*5
*5   Changes to release level March 2009 are marked with *5.
*5
*5   This and other current Release Notices are shipped on the SoftBooks DVD
*5   and are available online at http://manuals.ts.fujitsu.com/.

COBOL85, the COBOL compiler in BS2000/OSD *), provides you with the means of
programming according to the COBOL ANSI X3.23-1985 standard (= ISO/IEC
1989-1985 = EN 21989 = DIN 66028-1986).

As of COBOL85 (BS2000) V2.1A, support is provided for the standard ANSI
X3.23a-1989 or ISO/IEC 1989 Amendment1, Intrinsic Function Module.

As with V2.0A, COBOL85 V2.3A also fully supports the COBOL X3.23-1985 High
Level standard.

The optional "Report write" and "Segmentation" modules are provided with full func-
tionality, apart from the mandatory modules required by the standard.

COBOL85 also contains the "Data Manipulation Language DML" for operat-
ing

the UDS database system.

With COBOL85 V2.3A, you can display new passwords of the future stan-
dard.

In addition to the language extensions, the POSIX and XPG4programming interfac-
es are also supported in COBOL85 V2.3A.

As of COBOL85 (BS2000) V2.1A, the COBOL85 compiler is supplied w i t h o u t
the runtime system. The common runtime system for COBOL85, C and C++ in
BS2000 is the Common Run-Time Environment CRTE.

COBOL85 is "ready for 2000" **) as of V2.2.

Three variants of COBOL85 V2.3A are available:

- BC - Basic Configuration
- Full configuration
- R - RISC code generation variant

_____
*4   *) BS2000/OSD (R) is a registered trademark of Fujitsu Technology Solutions
*4   **) A product is "Ready for 2000", if
*4   - no problems which hinder use are known, or if corrections are available in good
*4   time for them, and
*4   - a free of charge update or upgrade will be supplied to eliminate any problems in
*4   the product which may occur in connection with the year 2000

The last variant is only released on special request (special release).

- POSIX interfaces

The functions in the full configuration not supported in the basic configuration are listed below:

- REPORT WRITER
- CODASYL-DML (native UDS support)
- Symbolic debugging with AID
- COBOL85 structurizer
- Output of a list of all error messages
- Output of object lists

The RISC variant includes the following a d d i t i o n a l function over the full configuration:

- RISC code generation

This Release Notice describes the full configuration, if nothing to the contrary is noted.

If one or more previous versions are skipped when the product is installed, the installation information from the Release Notices of the previous versions must be noted.

1.1   Ordering

*4   You can order the software from your local region of Fujitsu Technology
*4   Solutions. This product is subject to the general terms and conditions of
*3   the software product use and service agreement.

1.2   Delivery

The COBOL85 V2.3A files are supplied via SOLIS. The current file and volume characteristics are listed in the SOLIS2 delivery cover letter.

Basic configuration product components:

SINPRC.COBOL85-BC.023.DO-INST	Call installation procedure
SINPRC.COBOL85-BC.023.INSTAL	Internal installation procedure
SYSFGM.COBOL85-FGM.023.D	Release Notice (German)
SYSFGM.COBOL85-FGM.023.E	Release Notice (English)
SYSMES.COBOL85-BC.023	Message file (MSGMAKER format)
SYSMSV.COBOL85-BC.023	Message primary file
SYSPRG.COBOL85-BC.023	Compiler
SYSSDF.COBOL85-BC.023	System syntax file for BS2000/OSD
SYSSDF.COBOL85-BC.023.INSTAL	User syntax file for Installation

	SYSSDF.COBOL85-BC.023.SYSTEM	System syntax file for BS2000 V10
	SYSSDF.COBOL85-BC.023.USER	User syntax file
	SYSSII.COBOL85-BC.023	IMON information file
	SYSSII.COBOL85-FGM.023	IMON information file for FGM
	SYSSPR.COBOL85-BC.023.COMPILE	SDF start procedure for compiler

Full configuration product components:

	SINLIB.COBOL85.023	Library for installation in POSIX
	SINPRC.COBOL85.023.DO-INST	Call installation procedure
	SINPRC.COBOL85.023.INSTAL	Internal installation procedure
*5	SYSFGM.COBOL85.023.D	Release Notice (German)
*5	SYSFGM.COBOL85.023.E	Release Notice (English)
	SYSMES.COBOL85.023	Message file (MSGMAKER format)
	SYSMSV.COBOL85.023	Message primary file
	SYSPRG.COBOL85.023	Compiler
	SYSPRG.COBOL85.023.IND	Structurizer Beautify
	SYSPRG.COBOL85.023.VERDI	Structurizer Pretty-Print
	SYSSDF.COBOL85.023	System syntax file for BS2000/OSD
	SYSSDF.COBOL85.023.INSTAL	User syntax file for installation
	SYSSDF.COBOL85.023.SYSTEM	System syntax file for BS2000 V10
	SYSSDF.COBOL85.023.USER	User syntax file
	SYSSII.COBOL85.023	IMON information file
	SYSSII.COBOL85-FGM.023	IMON information file for FGM
	SYSSPR.COBOL85.023.COMPILE	SDF start procedure for compiler
	SYSSPR.COBOL85.023.STRUCT	SDF start procedure for structurizer

RISC variant product components:

	SINLIB.COBOL85-R.023	Library for installation in POSIX
	SINPRC.COBOL85-R.023.DO-INST	Call installation procedure
	SINPRC.COBOL85-R.023.INSTAL	Internal installation procedure
	SYSFGM.COBOL85-FGM.023.D	Release Notice (German)
	SYSFGM.COBOL85-FGM.023.E	Release Notice (English)

SYSLNK.COBOL85-R.023.ASS	Dynamic load library with ASSTRAN
SYSMES.COBOL85-R.023	Message file (MSGMAKER format)
SYSMSV.COBOL85-R.023	Message primary file
SYSPRG.COBOL85-R.023	Compiler
SYSPRG.COBOL85-R.023.IND	Structurizer Beautify
SYSPRG.COBOL85-R.023.VERDI	Structurizer Pretty-Print
SYSSDF.COBOL85-R.023	System syntax file for BS2000/OSD
SYSSDF.COBOL85-R.023.INSTAL	User syntax file for installation
SYSSDF.COBOL85-R.023.USER	User syntax file
SYSSII.COBOL85-R.023	IMON information file
SYSSII.COBOL85-FGM.023	IMON information file for FGM
SYSSPR.COBOL85-R.023.COMPILE	SDF start procedure for compiler
SYSSPR.COBOL85-R.023.STRUCT	SDF start procedure for structurizer

1.3   Documentation

The following manuals apply for COBOL85 V2.3A (for the basic and full configurations and the RISC variant):

COBOL85 (BS2000/OSD) V2.3
Sprachbeschreibung

COBOL85 (BS2000/OSD) V2.3
Language Reference Manual

COBOL85 (BS2000/OSD) V2.3
Benutzerhandbuch

COBOL85 (BS2000/OSD) V2.3
User Guide

COBOL85 (BS2000/OSD) V2.3
Tabellenheft

COBOL85 (BS2000/OSD) V2.3
Ready Reference

In addition to the above, the manuals for CRTE (BS2000/OSD) V2.1 and for the BS2000 standard configuration are recommended for operating COBOL85.

The following manuals are additionally recommended for using the full configuration of the compiler:
- The relevant AID manuals for debugging with AID
- The relevant POSIX manuals when using POSIX

*5 The BS2000/OSD documentation is available in German and English on
*5 DVD with the title BS2000/OSD SoftBooks.
*5
*5 The documentation is also available in the form of online manuals at
*5 http://manuals.ts.fujitsu.com or can be ordered at extra cost at
*5 http://manualshop.ts.fujitsu.com. In the manual shop only manuals are available for
*5 products which are still orderable.
*5
*5 The manuals may be supplemented with README files. These contain changes
*5 and extensions to the manual of the product concerned.
*5
*5 The file names are made up as follows:
*5
*5 SYSRME.<product>.<version>.E (file with English text)
*5 SYSRME.<product>.<version>.D (file with German text)
*5
*5 When printing the files, you should specify the CONTR-CHAR=EBCDIC operand in
*5 the PRINT-FILE command.

2 Software extensions

Only the main extensions and enhancements over the previous version COBOL85 V2.2C are described in the following sections.

2.1 Support   for converting to the next COBOL standard

Only the main extensions and enhancements over the previous version COBOL85 V2.2C are described in the following sections.

If necessary, the compiler informs the user of the new keywords of the next CO-BOL standard. To use this functionality, the COMOPT MARK-NEW-KEYWORDS or SDF option of the same name is to be used.

The new compiler option simplifies conversion to the next COBOL standard. If the option is set, keywords from the future standard are identified with severity I error messages. This avoids unnecessary conversion in new programs and also allows the conversion to be started now.

The syntax and semantics of the compiler option are documented in the User Guide.

2.2 Segmentation handling

The language features for segmentation will become obsolete with the COBOL85 standard and will not be included in the next standard.

For this reason, a compiler option has been created for controlling segmentation handling. The name of the COMOPT and its values are ELABORATE-SEGMENTATION=YES/NO. The corresponding SDF option is SEGMENTA-TION=ELABORATE/IGNORE.

If ELABORATE-SEGMENTATION has the value NO or SEGMENTATION =IGNORE was specified with the relevant SDF option, segmentation-oriented lan-guage features are ignored and acknowledged with a warning. This behavior cor-responds to the DEFAULT setting of the currently supplied compiler.

If ELABORATE-SEGMENTATION has the value YES or SEGMENTATION =ELABORATE was specified with the relevant SDF option, segmentation is sup-ported as previously as long as the program does not contain any nested routines.

The syntax and semantics of the compiler option are documented in the User Guide.

2.3 Support for change of century

The DATE-TO-YYYYMMDD, DAY-TO-YYYYDDD and YEAR-TO-YYYY functions are for converting two-digit years into four-digit years. They are an extension to the COBOL85 standard and will be incorporated into the next COBOL standard.

To supplement this, a language extension has been implemented into the SORT/MERGE statement to enable two-digit years to be used as SORT keys, de-pendent on a defined century window. This SORT interface extension simplifies the changeover to the year 2000. This language extension is not part of a COBOL standard.

The syntax and semantics of the functions and statements are documented in the language description.

2.4 Additional language extensions

The DELIMITED BY SIZE parameter of the STRING statement may be omitted immediately before the INTO parameter. In this case, DELIMITED BY SIZE is assumed. This language extension will be incorporated into the next COBOL standard.

The syntax and semantics of the STRING statement are documentedin the language description.

2.5 RISC code generation support

Note: this point only applies for the RISC variant.

The RISC variant of the COBOL85 compiler can generate both /390 and RISC code. CROSS compiling is supported.

*2 With RISC code generation, object code specific to the RISC processor is
*2 generated. This will only run on SR2000/MIPS. The generated object code is more
*2 efficient since it runs native on the RISC processor and not via the firmware. The
*2 increase in efficiency is only visible in realtime and not in "standardized CPU time",
*2 which is set by default in most cases. The reason for this is that standardized CPU
*2 time weights CPU utilization for /390 and RISC code differently in order to stan-
*2 dardize accounting of elementary operations which are always related to /390 op-
*2 erations.

To use this functionality, the COMOPT GENERATE-RISC-CODE or SDF DESTI-NATION-CODE=RISC-4000 option must be specified.

The syntax and semantics are not described in the User Guide.
This compiler variant can only be obtained as a special release. The syntax of the corresponding COMOPT is as follows:

        COMOPT GENERATE-RISC-CODE=YES/<u>NO</u>
or
        COMOPT GEN-RISC=YES/<u>NO</u>

The target code generated is as follows:
- If NO is specified, /390 code is generated.
- If YES is specified, RISC code is generated.

The syntax of the SDF option is shown below:

        COMPILER-ACTION=.../<u>MODULE-GENERATION</u>(...)

          MODULE-GENERATION(...)
               | ...
               | ,DESTINATION-CODE=<u>STD</u>/RISC-4000

DESTINATION-CODE=STD/RISC-4000
The target code generated is as follows:
- If STD is specified, /390 code is generated.
- If RISC-4000 is specified, RISC code is generated.

Segmentation-oriented language features are ignored when generating RISC code (see also section 2.2).

RISC code generation is not supported in POSIX. It is also not possible to store RISC objects in POSIX.

A file must be opened, accessed and closed in the same processor mode (/390 mode or RISC mode). This is always ensured if no EXTERNAL-clause in the file description is used.

*2
*2
*2
*2
*2

2.6 Support of the EURO character

As of V02.3A10, the EURO character will be supported in the CURRENCY SIGN clause and permitted as a currency symbol in the PICTURE clause. The fields converted in this way can be used for further processing or output.

*5
*5
*5
*5
*5
*5

2.7 Product-internal SW adaptation

To improve the maintainability of COBOL85 a product-internal SW adaptation has been implemented in Version 2.3A23.

3   Technical information

3.1   Resource requirements

The files supplied with the product occupy the number of PAM pages shown in the table below.

Variant:	Resources required:
Basic configuration	approx. 2500 PAM pages
Full configuration	approx. 9900 PAM pages
RISC variant	approx. 13600 PAM pages

*5

The SINLIB.COBOL85.023 library can be deleted if the POSIX file system is not used with the full configuration.

The compiler requires at least 11 MB of virtual address space during compilation.

Appreciably more virtual address space is required when generating RISC code with the RISC variant. The space required can be calculated approximately with:32 MB + DATA + NLOC * k KB where NLOC is the number of source lines without comments and DATA is the size of the user-specific data in the module. The constant k is between 0.5 and 1. Generating RISC code requires appreciably more time.

Using the POSIX functionality (ENABLE-UFS-ACCESS=YES) increases the space requirement of the linked program. This space requirement can be reduced by linking with the CRTE SYSLNK.CRTE.PARTIAL-BIND library and preloading the CRTESIS subsystem, which is a CRTE component.

3.2   Software configuration

COBOL85 V2.3A is released for BS2000/OSD as of V1.0A or OSD-SVP as of V2.0A. You need correction package

> BS2000/OSD-BC V1.0A (as of SOLIS change level 30), or
> BS2GA.BS2OSD V2.0A (as of SOLIS change level 10), or
> BS2GA.BS2OSD V3.0A (as of SOLIS change level 03).

CRTE-MSG V1.1 is included in these correction packages.

The "Common Run-Time Environment" CRTE as of V2.1A must be installed for running COBOL85 V2.3A objects. CRTE is a separate delivery unit and must be ordered separately for models with /390 system architecture. CRTE V2.1A is released for OSD as of V1.0 and is included in OSD-SVP as of V2.0A for the SR2000 models.

The POSIX file system can only be used by the compiler or the created objects (ENABLE-UFS-ACCESS=YES) as of OSD V2.0. The POSIX subsystem must be loaded for this. POSIX applications produced with CRTE V2.1A require POSIX-BC V1.1 under OSD V2.0, POSIX-BC V3.0 under OSD V3.0 and POSIX-BC V3.1 under OSD V3.1.

The following products are required for compilation:

Product	As of version
CRTE	V2.1A
SDF	V3.0
SDF-P-BASYS	V1.0B
SDF-P (1)	V1.0
ILAM/PLAM (2)	V3.0A
LLMAM (5	V2.4
SDF-A (3)	V3.0
UDS/SQL (4)	V1.2A
BINDER	V1.2

(1) Only for compilation in BS2000 with access to COPY elements in the POSIX file system or for writing listings into the POSIX file system

(2) For using the POSIX functionality as of V3.0B

(3) Only for using the installation procedure. An appropriate,free of charge SDF-U version can be used as an alternative (see section 3.3.3)

(4) Only if corresponding COBOL language features or system services are used

(5) For RISC code generation with the RISC variant as of V3.1A10

The following products are required for running COBOL85 objects:

Product	As of version
CRTE	V2.1A
BINDER (1)(2)	V1.0
SDF-P-BASYS (1)	V1.0B
AID (1)(3)	V2.1A
LMS (1)	V3.0
JV (1)	(6)
SORT (1)(7)	V7.4
UDS/SQL (1)(4)	V1.2A
UTM (5)	V3.3

(1) Only if corresponding COBOL language features or system services are used

(2) Only if LLMs were created or are to be processed otherwise, TSOSLNK or DBL suffices;
at least BINDER V1.2 is required for the cobol command in the POSIX shell

(3) If AID-FE is used, V2.1B is required

(4) FIND-7 features in particular may require a corresponding later UDS/SQL version

(5) Only if COBOL objects are to run as UTM subroutines

(6) Executable as of BS2000 version 10

(7) Language extension of the SORT/MERGE statement (see section 2.3) requires V7.7

3.3   Product installation

3.3.1   General

The standard installation procedure is SOLIS, and as of BS2000/OSD V3, IMON. If the product IMON (Installation MONitor) has been started on your system, it can be used as the standard installation procedure.

This reads all product files into the BS2000 system.

If you also wish to install COBOL85 or COBOL85-R in POSIX, after successful SOLIS/IMON installation you call the POSIX installation program (refer to the "POSIX Basics" manual for a description) under TSOS with:
       /CALL-PROCEDURE(SINPRC.POSIX-BC.<version>,POSINST)
Select 'Install packages on POSIX' and then input COBOL85 or COBOL85-R as the product and 023 as the product version.

3.3.2   Installation for use with COMOPT control

The COBOL85 compiler can be installed under any ID.

3.3.3   Installation for use with SDF control

The following delivery components are installed under the default ID by default:

       - SYSPRG.COBOL85.023   (Compiler)
       - SYSSDF.COBOL85.023.USER   (User syntax file)
       - SYSSDF.COBOL85.023   (System syntax file)
       - SYSSPR.COBOL85.023.COMPILE (Compiler procedure)

The COBOL85 syntax file SYSSDF.COBOL85.023 is merged into the system syntax file and the message files are installed automatically with the standard SOLIS/IMON installation.

Using the SINPRC.COBOL85.023.DO-INST procedure

The name of the compiler is no longer COBOL85 as in earlier versions, but rather SYSPRG.COBOL85.023. This allows several COBOL85 compilers to be in-stalled under one ID. If different file names or a different installation ID is desired, the following values can be adjusted to individual requirements with the SINPRC.COBOL85.023.DO-INST installation procedure:

       - Name of the compiler
       - Name of the syntax file
       - User ID for compiler and syntax file

All required changes are to be made with a single call to the installation procedure. The file SYSSDF.COBOL85.023.SYSTEM is required for the proper process of the installation procedure.
If required the file can be deleted after the installation.

The installation procedure can also be called by an ID other than that used for reading the product files into the BS2000 system. If the product files were read in under the ID $<cobuid>, the procedure is called as follows:

>   /CALL-PROCEDURE $<cobuid>.SINPRC.COBOL85.023.DO-INST,-
>   /   PROC-PAR=(CURRENT-FILE-USERID=<cobuid>)

Caution: this procedure works by default with the software product SDF-A.

Under the condition described below, the free of charge SDF-U can be used instead of SDF-A in the installation procedure.

SDF-U V4.0A is required with Rep correction from the following correction package:

>   BS2GA.SDF   V11.0B (SOLIS change level 04) or
>   BS2GA.SDF   V2.0A (SOLIS change level 03)

If the appropriate SDF-U version is available, the installationprocedure is called as follows:

>   /CALL-PROCEDURE SINPRC.COBOL85.023.DO-INST,-
>   /   PROC-PAR=(SDFA='START-SDF-U')

After the installation procedure has terminated, the /START-COBOL85-COMPILER command can be activated by assigning the COBOL85 user syntax file (temporary) or by merging in the COBOL85 system syntax file (permanent).

Comment: for COBOL85-BC: COBOL85-BC must be used in all file names instead of COBOL85. For COBOL85-R: COBOL85-R must be used instead of COBOL85.

3.3.4   Installation for use in POSIX

Note: this point does not apply for COBOL85-BC.

Please use the POSIX installation tool to install the compiler in POSIX. Please also note that this tool expects the SINLIB file under the default user ID. If you have not installed the compiler under the default user ID, to enable use in POSIX you have to copy the SINLIB file to there (see the manual "POSIX Basics" for a description).

3.3.5   Installation for using the structurizer functions

Note: this point does apply for COBOL85-BC.

The following delivery components must be present under the same ID as the components named in section 3.3.3:

>   - SYSPRG.COBOL85.023.IND   (Structurizer Beautify)
>   - SYSPRG.COBOL85.023.VERDI   (Structurizer Pretty-Print)
>   - SYSSPR.COBOL85.023.STRUCT (Structurizer procedure)

The /START-COBOL85-STRUCTURIZER command is activated together with the /START-COBOL85-COMPILER command by assigning the COBOL85 user syntax file or by merging in the COBOL85 system syntax file.

3.4 Product use

The COBOL85 V2.3A compiler is executable as of BS2000/OSD V1.0.

The compiler POSIX functionality can be used as of BS2000/OSD V2.0.

3.5 Discontinued functions (and those to be discontinued)

3.6 Incompatibilities

The compiler options USE-APOSTROPHE, RESET-PERFORM-EXITS and REPLACE-PSEUDOTEXT are reset to their default values by setting one of the following compiler options: FLAG-ABOVE-MINIMUM, FLAG-ABOVE-INTERMEDIATE, FLAG-ALL-SEGMENTATION, FLAG-INTRINSIC-FUNCTIONS, FLAG-NONSTANDARD, FLAG-OBSOLETE, FLAG-REPORT-WRITER or FLAG-SEGMENTATION-ABOVE1. This was pre-viously indicated by a severity I error message. Due to extensive semantic changes caused by resetting the compiler options, the severity code has been increased from I to 2 for the safety of the user!

If a sequence of programs is held in one source, when this source is compiled only one options list is created and no longer an options list for each program as pre-viously. Comment lines between the separate programs were previously ignored. They are now assigned to the subsequent program in each case.
Comment lines after the end of the last program are ignored.

COPY and REPLACE statements are still executed if they are specified in comment lines. Comment entries are obsolete in the COBOL85 standard and are not included in the next standard. Any problems which may occur because of this change can be easily avoided by inserting an asterisk in column 7 of the lines concerned.
The compilation date is no longer listed in the DATE-COMPILED paragraph of the source list. It is included in the list header.

3.7 Restrictions

Alias Catalog System (ACS):

Standard file names (xxxLST.COB85.programname) may n o t be entered as alias file names since ACS does n o t consider alias names with and without a prefixed user ID to be the same.

```
*3              COBOL85 Beautifiers:
*3
*3              - The effect of the LAYOUT PARAMETER's LINE-SIZE and
*3              LOWER-CASE-KEYWORDS options for the PRETTY-PRINTER is not
*3              defined for non-standard printers. The BOLD-FACE=YES option
*3              for the LAYOUT-PARAMETER is not supported for output to SYSLIST.
*3
*3              - In some rare cases, the beautifiers insert blank lines after comment lines.
*3              Using the X'15' character causes the beautifier to truncate the respective line.
*3
*3              - The beautifiers do not support nested programs.
*3
*3              - The beautifiers support the processing of source files with up to
*3              approx. 30,000 lines, including the expanded COPY elements.
*3
*3              - IF the beautifier reads a source file from SYSDTA, SYSCMD is used to form the
*3              filename instead of the source program  filename when the output is directed to
*3              STD-FILES.
*3
*3              - With errors of Error Class 2, the beautifier triggers the SPINOFF mechanism.
```

- With the structurizer "Pretty Print" function, the statements COPY textname RE-PLACING ==pseudotext== BY ==pseudotext== REPLACE ==pseudotext== BY ==pseudotext== are only accepted syntactically. However, the intended text re-placement is not shown in the program text. The non-replaced texts can cause syn-tax errors. If a syntax error is found in the copied area of a COPY statement with pseudo-text assignment, an error message is output and no further syntax errors are reported in this copied area (Manual reference: chapter 5 of the User Guide).
- Nested programs are not correctly processed by the "Beautify" function of the structurizer.

Due to the system-specific differences between POSIX and BS2000, a few lan-guage scope and run behavior points have to be noted when using COBOL pro-grams which use POSIX interfaces (ENABLE-UFS-ACCESS=YES). (Manual refer-ence: section 13.5 of the User Guide).

When using the ENABLE-UFS-ACCESS=YES option, no support is provided for us-ing the STANDARD-2 character set (International Reference Version of the ISO 7-bit code) specified in the ALPHABET clause, in the CODE SET clause. An OPEN of this type is rejected at runtime with FILE STATUS 30.

If access to the POSIX file system has been enabled via the ENABLE-UFS-ACCESS option, the COB9151 and COB9175 messages contain the relevant SIS message numbers instead of the DMS codes for POSIX file access errors. The same applies for any extended file status that may be returned to the COBOL object.
The returned file status may also deviate from the previously known value (Manual reference: section 13.6.3 of the User Guide).

If the ENABLE-UFS-ACCESS=YES option is used, the components then used in-ternally always require that the program runs under ILCS. This is ensured with a COBOL85 main program as of V1.1.
Main programs in other languages must ensure that ILCS is initialized in this case.

Prelinked main modules or link/load modules containing older COBOL85 runtime modules are not supported by the current runtime system. Such main modules have to be relinked. However, this does not restrict the backward compatibility to old CO-BOL85 object modules.

Duplicates of alternate keys are not correctly detected in POSIX with indexed files (file status 02 is not issued).

The LOCKING behavior known from SINIX is modified in POSIX. For example, it is possible to open the same file several times.

Shared file processing (SHARED-UPDATE) is only supported when compiled with COMOPT or RUNTIME-OPTION ENABLE-UFS-ACCESS=NO.

The following restriction applies for UDS/SQL V2.0A support:

- If the input field for FIND or ACCEPT is of type DATABASE-KEY-LONG and
- the subschema is present in the old format (UDS/SQL V1.2), statements following FETCH1 and ACCEPT2 can only be correctly processed if
- the value in the input field can be displayed in a field of type DATABASE-KEY.

If it is not possible to ensure that the value in the input field complies with these re-quirements, the problem can be avoided as follows:
- Compile the subschema with the DDL compiler, without the SUBSCHEMA FORM IS OLD option and
- then compile the COBOL program concerned.

*2 Note on files with the COBOL RELATIVE file format:
*2
*2 Relative files can be created in DMS file format with FCB-TYPE=ISAM, or with
*2 FCBTYPE=PAM for compatibility. In the latter case, The PAMKEY provided by the
*2 system is mandatory to ensure that the file is correctly processed! Such files may
*2 not be stored on NONKEY disks. Mechanical conversion with conversion
*2 programs such as PAMCONV is not meaningful. These tools do not know the
*2 COBOL85-specific file structure and would destroy the files. The only way to store
*2 such relative files on NONKEYdisks is by converting them with a COBOL program
*2 that reads the existing files with FCBTYPE=PAM and creates a copy with
*2 FCBTYPE=ISAM. A suitable program is provided in section 5.1.
*2 It is generally not necessary to adapt the COBOL program which converts the files
*2 *). Such a conversion provides a good reason for using the enhanced
*2 options of relative files on ISAM.

*1 The following note only applies to the RISC variant.
*1
*1 In rare cases, the RISC code generation is aborted with the following message
*1   S.E. (ERROR '!!!0000' DURING GENERATION OF
*1   SELECTED DESTINATION CODE)
*1 or
*1   S.E. (FEHLER '!!!0000' BEI DER GENERIERUNG DES
*1   AUSGEWAEHLTEN DESTINATION CODE)
*1 S.E. means System Error. No RISC code was generated. In this case please gen-
*1 erate a /390 object (see section 2.5).
*1 This object can be linked with RISC objects.

---

*2 *) Unfavorable blocking may occur with record sizes close to 2K and, if a BLOCK
*2   CONTAINS clause is specified, compilation errors may occur under certain
*2   conditions.

3.8   Procedure in the event of errors

If errors occur, please send the documentation listed below to your local regional service office:

	Compile errorr	Runtime error
SOURCE, DIAG and COMOPT list	X	X
Run log (MSG=FH)	X	X
Source (incl. COPY elements and COSSD (COBOL sub-schema directory) if necessary)	X	X
Link list		X
Input/output files		X
Expected result		X
Brief description of sequence		X

A detailed description of the error condition, indicating whether and how the error can be reproduced is required for diagnostic purposes.

4 Hardware support

*5   COBOL85 V2.3A is executable on models with /390, RISC and X86 system archi-
*5   tecture (no 25-bit systems) that are supported by the operating system versions
*5   listed in section 3.2.

5 Appendix

5.1 Converting relative files with FCBTYPE=PAM

As explained in section 3.7, relative files with FCBTYPE=PAM must be converted with a COBOL program to relative files with FCBTYPE=ISAM before they can be stored on NONKEY disks. It is generally not necessary to adapt the COBOL program which converts the files.

The following program demonstrates how such a conversion coulbe implemented.

```
        IDENTIFICATION DIVISION.
        PROGRAM-ID.            RELCONV.

        ENVIRONMENT DIVISION.
        CONFIGURATION SECTION.
        INPUT-OUTPUT SECTION.
        FILE-CONTROL.

   *      Clauses of the existing file
   *      ORGANIZATION RELATIVE
   *      ACCESS     SEQUENTIAL
   *      RELATIVE KEY ...
        SELECT RELPAM
          ORGANIZATION        IS RELATIVE
          ACCESS          IS SEQUENTIAL
          RELATIVE KEY      IS REL-KEY
          ASSIGN          TO "INFILE".

   *      Clauses of the file to be created
   *      ORGANIZATION RELATIVE
   *      ACCESS     RANDOM
   *      RELATIVE KEY ...
        SELECT RELISAM
          ORGANIZATION        IS RELATIVE
          ACCESS          IS RANDOM
          RELATIVE KEY      IS REL-KEY
          ASSIGN          TO "OUTFILE".

        DATA DIVISION.
        FILE SECTION.

   *      Clauses and record description of the
   *      existing file
        FD RELPAM.
        01  REC-PAM.
           ...

   *      Clauses and record description of the
   *      file to be created
        FD RELISAM.
        01  REC-ISAM.


            ...
```

```
*2                      WORKING-STORAGE SECTION.
*2                      01  REL-KEY          ...
*2                      01  IO-STATUS        PIC X(3) VALUE SPACS.
*2                       88 EOF              VALUE "EOF".
*2                       88 NOT-EOF           VALUE SPACES.
*2
*2                      PROCEDURE DIVISION.
*2                        OPEN INPUT RELPAM, OUTPUT RELISAM
*2                        READ RELPAM
*2                         AT END SET EOF TO TRUE
*2                        END-READ
*2                        PERFORM UNTIL EOF
*2                         WRITE REC-ISAM FROM REC-PAM
*2            *     INVALID KEY should not happen!
*2                          INVALID KEY ...
*2                         END-WRITE
*2                         READ RELPAM
*2                          AT END SET EOF TO TRUE
*2                         END-READ
*2                        END-PERFORM
*2                        CLOSE RELPAM, RELISAM
*2          STOP RUN.
```