

COLUMBUS85 – BS2000/OSD Version 1.0

Issue October 2004

Pages 2

Support system for structured programming and the use of decision tables in COBOL85

COLUMBUS85 provides support for program design and generative program creation, based on the principles of structured programming and including the use of decision tables.

According to the principles of structured programming, a program is constructed as a hierarchy of procedures. Procedures have only one entry and one exit, and in turn consist of structure blocks. In terms of internal control flow, there are only three forms of structure block (sequence, selection, repetition). Decision tables are a special form of structure block used for selection. Structure blocks can be strung together and nested inside each other. Structure lists are the best way of showing these forms.

In the COLUMBUS language, forms of structure block can be expressed directly with a number of variants. The internal structure of a structure block includes predefined function words for this purpose. These function words are used to manage the control flow on one hand, and to group instructions on the other. There are also function words to delimit, call, and copy in procedures or code sections.

The function words are the only specific language elements of the COLUMBUS language. They provide a framework, into which instructions and conditions can then be inserted. The instructions may be either further structure blocks, or formulated in any language required. A structure block may also contain a decision table.

COLUMBUS85 processes programs in which the control flow/structure are described by COBOL85 or COLUMBUS instructions.

The program content between the control flow instructions/structure instructions can be formulated using pseudocode in the design phase, or using COBOL85 instructions subsequently in the coding phase. Depending on the combination of control flow form/structure form and program contents form, these are referred to as COBOL85 pseudocode programs, COLUMBUS pseudocode programs, COBOL85 programs or COLUMBUS COBOL programs. Compared to its predecessor (COLUMBUS-COB), the main new feature of COLUMBUS85 is that COBOL85 control flow instructions/structure instructions can be used directly during the program design phase. At the same time, it is still possible to use the previous, language-independent control flow instructions/structure instructions of the COLUMBUS language. In each case, the pseudocode formulated as free text must be converted manually into the corresponding COBOL85 instructions when progressing from design to coding. The COLUMBUS control flow instructions/structure instructions of a COLUMBUS COBOL program are converted into a compilable COBOL85 program using the TRANSLATE-COLUMBUS-SOURCE action. The conversion is not required

if the COBOL85 control flow instructions/structure instructions are used directly.

The individual COLUMBUS85 actions are called and used via an SDF command interface. The following actions are available:

- PRETTY-PRINT
- BEAUTIFY-SOURCE
- PRECOMPILE
- TRANSLATE-COLUMBUS-SOURCE
- EXPAND-COPY

Functional Description

PRETTY-PRINT:

This generates a structure list of the source program. Various options can be set to indicate e.g. the COLUMBUS language delimiters, the type of numbering required, the layout of the structure list, and whether the structure list should include cross-references and / or pass counters. Structure lists can also be generated for pseudocode programs, and allow the design to be checked for structural consistency. Decision tables can also be included. The generation of cross-references and pass counters requires a COBOL85 program that can also contain decision tables.

BEAUTIFY-SOURCE:

This action can be used to show the logical structure of a program (or that of a pseudocode design!). Any decision tables contained in the program are transferred unchanged. There are options to define any delimiters that are not standard values, the numbering format, the layout (lower case, indents, Data Division characteristics), and where to output the result of the Beautifier pass. The BEAUTIFY-SOURCE action can also be started in the EDT.

PRECOMPILE:

This action can be used to convert any remaining decision tables in a COBOL85 program into corresponding COBOL85 instructions. This action is also required to include pass counters in a COBOL85 program to measure the extent of testing. After testing has been completed with the specially prepared program, a structure list with pass counters can be generated using the PRETTY-PRINT action, and based on the data generated during testing. The PRECOMPILE pass offers various control options, such as the output layout, whether to include pass counters, and the output destination. The conversion of decision tables and the inclusion of pass counters can be performed in one call. If the program contains

both decision tables and ESQL instructions, then the COLUMBUS85 PRECOMPILE action should be performed before the generated COBOL85 program (which still contains the ESQL instructions) is precompiled using the ESQL COBOL (BS2000) precompiler.

TRANSLATE-COLUMBUS-SOURCE:

This action is used to convert COLUMBUS control flow instructions into the corresponding COBOL85 structure instructions. This type of conversion is only required if COLUMBUS control flow instructions were used for the program design instead of COBOL85 instructions, as in the case of COLUMBUS-COB, or if there is a requirement to continue using existing COLUMBUS-COB programs. There are options to control which characters will be used as delimiters, how to interpret parameters in the case of @COPY, whether loops can be terminated by an EXIT PERFORM, how to handle the Data Division, how many characters to use when indenting structure blocks, and the destination of output from the action.

The TRANSLATE-COLUMBUS-SOURCE action cannot be used on pseudocode. Decision tables must be converted into COBOL85 instructions with the PRECOMPILE action.

Exceptions for conversion with the TRANSLATE-COLUMBUS-SOURCE action are COLUMBUS @COPY structure instructions with the REPLACING parameter, coroutines, and TOM-GEN macro calls.

EXPAND-COPY:

EXPAND-COPY is used to include files or library elements. This may require a REPLACING in COLUMBUS-COB format. Program Description COLUMBUS85 should be considered as a group of programs that can be called via the same SDF interface. The shared call command is:

```
/START-COLUMBUS85          or:
/COL
```

This call command is followed by details of the source program and the required COLUMBUS85 actions, which in turn are controlled by their respective options. COBOL85 is no longer a software requirement for COLUMBUS85 after revision level COLUMBUS85 V1.0C00. Since then, COLUMBUS85 includes the PrettyPrinter, Beautifier and ErrorText components previously supplied in the COBOL85 package for use with earlier versions of COLUMBUS85. As a result, the COLUMBUS85 product can also be used with the new BS2000 compiler system COBOL2000

(BS2000/OSD) ®. However, the scope of the language is still restricted to the inherent COBOL85 language range. COLUMBUS85 does not include support for new, COBOL2000-specific language elements, in particular those provided for the purposes of object-oriented programming. For a transitional period, existing COLUMBUS-COBOL users upgrading to the new COLUMBUS85 product will receive the files for COLUMBUS-COB V4.2 in addition to the new product COLUMBUS85 (BS2000) V1.0, until the migration to COLUMBUS85 is fully complete.

Technical Data

Hardware:

Business Server with /390 architecture
Business Server with RISC architecture (SR2000)

Memory requirements:

User address space: Basic system configuration
Disk space: Approx. 6,600 PAM pages

Software:

BS2000/OSD-BC V1.0 or later
for Business Servers with /390 architecture
BS2000/OSD-SVP V2.0 or later
for Business Servers with RISC architecture

Operating mode:

Interactive or batch mode

Implementation language:

C and Assembler

Documentation

COLUMBUS85 User Guide

Demands on the User

A knowledge of the COBOL85 programming language and the principles of structured programming.

Training

See course offer at:

<http://www.fujitsu-siemens.com/training>

Conditions

This software product is supplied to the customer under our conditions against a single payment or installments.

Warranty:

Class: A

Delivery format: Machine language

Ordering and Delivery

This software product may be obtained from your local Fujitsu Siemens Computers regional office.