

# DATA SHEET

## Fujitsu BS2000/OSD ASSEMBH Version 1.3 Software

Issue August 2016

Pages 3

**The Assembler is a machine language translation program for converting source code of the identically named symbolic programming language into corresponding machine code.**

### Product Characteristic

Assembler programs are general-purpose utilities for deployment in all areas of data processing and can be used in batch mode as well as in interactive mode. Since the Assembler is machine-specific by nature (an Assembler language instruction corresponds to each machine instruction), the technical capabilities of the particular CPU involved can be exploited to the maximum extent.

ASSEMBH is a special Assembler language developed for the instruction set of the business server with /390 architecture.

It supports the following instruction sets:

- XS; eXtended System with 31-bit addressing
  - ESA; Enterprise System Architecture for utilizing multiple virtual address spaces for data (data spaces)
- As assembler programs can be interfaced with high-level programming languages, these special features can also be accessed from within other languages as and when required.

Use of the operating system interfaces is enabled via the macro library of the respective BS2000/OSD operating system version.

ASSEMBH offers a variety of functions including the following:

Structured programming facilities corresponding to COLUMBUS, with utility routines for generating listings (structure lists, Nassi-Shneidermann diagrams, etc.).

Facility for symbolic debugging using AID

Use of XS address space to achieve faster assembly speeds

Utilization of multiple virtual address space for data (data spaces) on ESA systems under BS2000/OSD-BC V1.0 or higher.

Extended control options via BS2000/OSD SDF interface.

For reasons of compatibility, the ASSEMB \*COMOPT control option is still supported, but the new services are not.

Monitoring job variables supported for automatic job sequencing.

Extensions to the macro language using clear syntactic and semantic rules (which can lead to compatibility problems).

Support for up to 100 macro libraries.

Support for up to 100 COPY libraries.

Processing of symbolic names with up to 64 characters. Upper- and lower-case characters allowed in source programs (not case-sensitive).

XREF also for COPYs.

Nested macro definitions.

Mixed sequence of positional and keyword operands in macros.

Support for ILCS (Inter-Language Communication Services) in structured programming.

### Functional Description

The language elements are:

- Machine instructions (commands)
- Assembler instructions (statements)
- Macro definitions,
- Macro calls and
- Comments.

Machine instructions enable the programmer to symbolically address the instruction set of the CPU from the Assembler. The Assembler language instructions control the internal assembly process at assembly time and perform auxiliary functions which include:

- Sectioning and linking of programs
- Definition of constants
- Control and assignment of memory addresses
- Recognition of different addressing modes.

Macro calls request the Assembler to call prewritten Assembler language subroutines from libraries, modify them in accordance with the parameters specified in the macro call, and insert the generated machine instructions in place of the macro call.

Comments are used to document the developed program and have no effect on the assembled program.

Macro language

The macro language is part of the Assembler. It enables the programmer to code frequently used Assembler language instruction sequences in the form of macro definitions. Macro definitions consist of:

- Header statement
- Prototype statement
- Model statement and

**■ Trailer statement.**

The model statements are statements that produce the desired code sequence after the assembly. Appropriate modifications can be made in macros as well as in Assembler source code by using special macro language elements such as SET and AIF instructions and variable parameters which may only be assigned current values at runtime.

**Program Description**

Once the Assembler has been loaded, multiple assembly runs can be performed. Only minimal control is required for a standard assembly run, but a number of parameters and options are additionally available to allow explicit control of the assembly process.

The assembler can read source code, macros, and copy elements from BS2000/OSD files or PLAM libraries.

Object can be output either in OM (object module) format to a PLAM library or \*EAM file (OMF) or in LLM format (link-and-load modules) to a PLAM library.

Logging information for listings is output with ASSEMBH in a special intermediate format (CIF) which can be processed further using list editing tools. This intermediate format can be permanently stored in a PLAM library for later processing.

Logs may be optionally output to SYSLST, to a file, or to a PLAM library.

Compatibility restrictions:

The strict syntax and semantic rules may result in structural incompatibilities with earlier Assembler source modules and macros, since ASSEMB wrongly tolerated certain syntax and semantic rule violations.

---

## TECHNICAL DETAILS

---

### ASSEMBH V1.3

---

#### Technical Requirements

##### Hardware

BS2000/OSD Business Server

##### Technical Requirements

##### Software

BS2000/OSD-BC V7.0  
OSD/XC V3.0 or higher  
Optional software requirements:  
AID for symbolic testing

##### Operating Mode

Batch and interactive mode

##### Memory requirements:

From 4 to 50 Mbytes depending on the program size

##### Implementation Language

SPL4, C and Assembler

##### User Interface

German / English

##### Installation

Notes on installation can be found in the relevant Release Notices.

#### Documentation

ASSEMBH (BS2000/OSD), User Guide  
ASSEMBH (BS2000/OSD), Reference Manual  
The manuals are available in German and English.

The documentation is available as online manuals, see <http://manuals.ts.fujitsu.com/mainframes.html>, or in printed form which must be paid for and ordered separately at <http://manuals.ts.fujitsu.com>.

#### Training

See course offer at:  
[https://sp.ts.fujitsu.com/dmsp/docs/ca\\_bs2000-course\\_de.pdf](https://sp.ts.fujitsu.com/dmsp/docs/ca_bs2000-course_de.pdf)

#### User Requirements

Knowledge of the programming language Assembler and of BS2000/OSD

#### Conditions

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

#### Ordering and Delivery

This software product may be obtained from your local Fujitsu Technology Solutions GmbH regional office

---

Information about environmental care, policies, programs and our Environmental Guideline FSC03230:

[ts.fujitsu.com/aboutus](http://ts.fujitsu.com/aboutus)

Take back and Recycling information: [ts.fujitsu.com/recycling](http://ts.fujitsu.com/recycling)

---