

---

# 1 Preface

This section presents a brief description of the product SDF-CONV V3.0A and offers hints on how to use the manual.

## 1.1 Brief product description

The SDF-CONV utility routine is able to perform two different types of procedure conversion:

- conversion of the command language (ISP into SDF commands)
- conversion of the procedure format (non-S into S format)

Since the release of BS2000 V7.5, two command processors have co-existed in BS2000: ISP (Interpretative **S**canner and **P**rocessor) and SDF (**S**ystem **D**ialog **F**acility).

As of BS2000 V10.0, most BS2000 manuals describe commands in the SDF command language (SDF commands).

SDF-CONV is a utility routine for converting ISP commands that are stored in files or library elements into SDF commands. The aim of the conversion is to produce executable procedures in the SDF command language from executable procedures in ISP format. The procedure to be converted is also known as the input procedure, and the converted procedure as the output procedure.

Some ISP commands cannot be converted into SDF commands. Instead, they are taken over unchanged into the output procedure. This ensures that the procedures can be executed.

Any SDF commands that form part of the input procedure are expanded to their full command name.

BS2000 V10.0 introduced a new procedure format co-existing with the format used in previous versions. Procedures of this new format are referred to as "structured procedures" or S procedures, while those of the old format are now referred to as non-S procedures.

As of V3.0, SDF-CONV is able to convert non-S procedures (containing ISP or SDF commands) into S procedures.

SDF-CONV writes the result of the conversion to an output procedure, which is stored as a SAM file or as an element in a PLAM library. A detailed log of the conversion run is output to SYSLST.

## 1.2 Target group and summary of contents

This manual is addressed to all users of BS2000 working with procedures that contain commands in ISP or SDF format.

The manual is organized into the following chapters:

### **Components and installation of SDF-CONV**

This chapter deals with the components of the product SDF-CONV and the functions they perform. It is especially relevant for first-time installation of SDF-CONV, as it describes how to install the subsystem.

### **Operation and execution of SDF-CONV**

This chapter describes the conversion process, all the important statements used when working with SDF-CONV, and the layout of the listings output by SDF-CONV. It concludes with a section on restrictions that apply to the conversion and the reasons for them.

### **Examples**

This chapter gives examples of the conversion of procedures containing ISP commands into procedures containing SDF commands as well as examples of the conversion of non-S procedures into S procedures. The resultant logs and the output procedures are accompanied by comments.

### **Appendix**

The notational conventions or metasyntax (SDF syntax) used in this manual, including the syntax for data types, are listed in the appendix. The appendix also contains tables in which the ISP commands are listed alongside the corresponding (equivalent) SDF commands. Subsequent tables show the ISP commands converted into SDF commands. Finally, a list of the error messages generated by SDF-CONV is given.

In the text, references to other publications are given using abbreviated titles. The full title of each publication referred to is listed in the *References* section at the back of the manual.

The following abbreviations are used in the manual:

DMS	Data Management System
DSSM	Dynamic SubSystem Management
ISAM	Index Sequential Access Method (file format)
ISP	Interpretative Scanner and Processor
MIP	Message Improvement Processing
OSM	Old Source or Macro library (library type)
PLAM	Program Library Access Method (library type)
SAM	Sequential Access Method (file format)
SDF	System Dialog Facility (command processor)

## 1.3 Changes since the last version of the manual

### *Formal changes*

As of V11.0, BS2000 has been renamed "BS2000/OSD" (Open Systems Direction). The name of the basic configuration has been changed from "BS2000-GA V11.0" to "BS2000/OSD-BC V1.0". These changes reflect the new orientation of BS2000 as an open server platform.

All names beginning with an asterisk and containing no other wildcards are interpreted as keywords.

All keyword operands start with an asterisk. Specification of keyword operands without the leading asterisk continues to be supported for compatibility reasons.

The UNCHANGED-COMMANDS operand has been renamed UNCHANGED-CMD.

### *Changes in functionality*

Elements of a PLAM input or output library need no longer be of type J. As of V3.0, SDF-CONV supports all types of element.

As of V3.0, SDF-CONV is able to convert non-S procedures into S procedures (PROCEDURE-FORMAT operand). The user can optionally specify whether the command language of the input procedure is to be retained or to be converted into the SDF command language, whether data lines are to be converted into statement lines and whether block input is to be permitted when writing the converted SDF commands to the output procedure.

The meaning of EXPECT-/PRODUCE-CONTINUATION=\*NEW-MODE has been expanded to include the processing of S procedures since command and statement lines in S procedures may have any length.

The new operand OUTPUT-FORM further improves the control of the handling of optional operands in command records of the input procedure.

Operand	Changes
FROM-FILE / TO-FILE – TYPE	Elements of a PLAM input or output library may now be of any type. – new suboperand for specifying the type of the input/output library elements
EXPECT-CONTINUATION / PRODUCE-CONTINUATION	The meaning of operand value *NEW-MODE has been enlarged to include S procedures.
UNCHANGED-COMMANDS	Renamed UNCHANGED-CMD
INVARIANT-INPUT	The operand has been omitted, its functionality being incorporated in the new operand OUTPUT-FORM. The old operand continues to be supported for compatibility reasons.
OUTPUT-FORM	New operand for the handling of optional operands in the input procedure.
PROCEDURE-FORMAT – CMD-FORMAT – PROGRAM-INPUT – EXCEPT-AFTER-CMD – WARNING – BLOCK-INPUT	New operand for defining the procedure format of the output file: – determines whether the commands of the input procedure are to be converted into SDF commands – determines how data lines are handled during the conversion into an S procedure – permits individual data lines to be excepted from the general handling of data lines – determines whether a warning is to be output whenever a data line is detected – determines the output format of generated SDF commands

Table 1: New and modified operands and operand values for the CONVERT statement

*README file*

Information on any functional changes and additions to the current product version described in this manual can be found in the product-specific README file. You will find the README file on your BS2000 computer under the file name

SYSDOC.SDF-CONV.030.READ-ME.E.

The user ID under which the README file is cataloged can be obtained from your system administrator. You can view the README file using the SHOW-FILE command or an editor, and print it out on a standard printer using the following command:

```
PRINT-FILE FILE-NAME=SYSDOC.SDF-CONV.030.READ-ME.E. ,-  
LAYOUT-CONTROL=PARAMETERS(CONTROL-CHARACTERS=EBCDIC)
```



## 2 Components and installation of SDF-CONV

The utility routine SDF-CONV V3.0A can run with BS2000 V10.0A and SDF V2.0A. The full range of functions supported by SDF-CONV V3.0A is available as of SDF V4.0A. SDF-CONV is supplied as a subsystem but can, for reasons of compatibility, also be started as a simple program. SDF-CONV converts ISP commands which were defined in a BS2000 version  $\leq$  V10.

The SDF-CONV utility routine consists of the following components:

- the module library SYSLNK.SDF-CONV.030 for calling the utility routine as a subsystem by means of the START-SDF-CONV command
- the conversion routine SDF-CONV which is available for reasons of compatibility for calling the utility routine by means of the command START-PROGRAM SDF-CONV
- the syntax files
  - SYSSDF.SDF-CONV.030, (system) syntax file containing the syntax for the SDF-CONV program statements,
  - SYSSDF.SDF-CONV.030.USER.U-CMD, (system) syntax file, and
  - SYSSDF.SDF-CONV.030.USER.S-CMD (group) syntax file containing the command syntax for the commands to be converted, split up according to functional range for system administration (group syntax file) and non-privileged users (system syntax file)
- the declaration files
  - SYSSSD.SDF-CONV.030, subsystem declaration for use in BS2000 V10
  - SYSSCC.SDF-CONV.030, subsystem declaration for use as of BS2000/OSD-BC V1.0
- the message files
  - SYSMSV.SDF-CONV.030 for use up to BS2000/OSD-BC V1.0
  - SYSMES.SDF-CONV.030 for use in BS2000/OSD-BC V2.0
- the REP file SYSREP.SDF-CONV.030 for object corrections.

### Subsystem installation

As of Version V2.0A, SDF-CONV is a BS2000 subsystem and is managed by DSSM. It cannot be loaded by DSSM until it has been declared in the subsystem catalog. The subsystem declarations required to do this are contained in the file SYSSDF.SDF-CONV.030 (for BS2000 V10) or SYSSSC.SDF-CONV.030 (for BS2000/OSD-BC V1.0).

Depending on which version of the operating system is being used, the system administrator has to proceed as follows to install the subsystem:

#### **as of BS2000/OSD-BC V1.0:**

The subsystem declaration SYSSSC.SDF-CONV.030 must be added to the subsystem catalog by means of SCCM. The START-SUBSYSTEM command is used to activate the SDF-CONV subsystem. This means that the message file, the syntax file SYSSDF.SDF-CONV.030 and the REP file also belong to the SDF-CONV subsystem and are activated automatically.

#### **in BS2000 V10:**

The subsystem declaration SYSSSD.SDF-CONV.030 must be added to the subsystem catalog during a UGEN run. The START-SUBSYSTEM command is used to activate the SDF-CONV subsystem. The message file has to be activated by means of the MODIFY-MSG-FILE-ASSIGNMENT command. The syntax file SYSSDF.SDF-CONV.030 is merged into the current system syntax file using the SDF-I utility routine. The REP file is activated automatically.

By default, the two other syntax files (SYSSDF.SDF-CONV.030.USER.U-CMD and SYSSDF.SDF-CONV.030.USER.S-CMD) are stored under the standard system ID "\$". In order to install them under another ID, the preset path names in the operands SYSTEM-SYNTAX-FILE and GROUP-SYNTAX-FILE of the CONVERT statement must be modified accordingly.

These syntax files must not be merged into the current BS2000 system or group syntax files.

Generation of the subsystem catalog and the SCCM routine is described in the manual "System Administrator's Guide" [7], management of group and system syntax files and the SDF-I utility routine are described in the manual "SDF Management" [5], and UGEN is described in the manual "System Installation" [6].

#### *Note:*

If optional REPs are used in the system, preset operand values in commands such as CATALOG can be modified. SDF-CONV ignores these changes unless the system administrator enters them in the syntax files using the SDF-A utility routine.



### 3 Operation and execution of SDF-CONV

The SDF-CONV utility routine is able to perform two different types of procedure conversion:

- conversion of the command language (ISP into SDF commands)
- conversion of the procedure format (non-S into S format)

SDF-CONV converts ISP commands and their operands in files and library elements into SDF commands with the appropriate operands. Commands in SDF command records are expanded to their full command names.

Certain ISP commands cannot be converted into SDF format. Such commands are added unchanged to the output procedure. For example, ISP commands are not converted if the selection of the corresponding SDF command is dependent on the runtime behavior or on procedure parameters or the ISP command is no longer supported by BS2000.

All commands and operand configurations that cannot be converted by SDF-CONV are listed in the section "Restrictions" (as of page 32).

As of V3.0, SDF-CONV can be used to convert non-S procedures (ISP and SDF format) into S procedures.

An S procedure ("structured procedure") is structured in accordance with SDF-P rules. SDF-P is a block-oriented programming language. It enhances the command language used in BS2000 up to now by enabling structured programming similar to higher programming languages (see the manual "SDF-P" [8]).

The software product SDF-P includes the following two components:

- the non-chargeable subsystem SDF-P-BASYS containing the basic SDF-P components
- the chargeable subsystem SDF-P offering the full SDF-P functionality

Only the functionality of the non-chargeable SDF-P-BASYS is taken into account when converting a non-S procedure into an S procedure using SDF-CONV (see manuals "User Commands (SDF Format)" Part1 [1] and Part 2 [2]).

The resulting S procedure is compatible but not optimized in comparison with the control structure mechanisms of an S procedure generated with SDF-P.

The following conversions of commands and procedure formats can be performed using SDF-CONV V3.0:

non-S procedure with ISP commands	→	non-S procedure with SDF commands
non-S procedure with ISP commands	→	S procedure with ISP commands
non-S procedure with ISP commands	→	S procedure with SDF commands
non-S procedure with SDF commands	→	S procedure with SDF commands
S procedure with ISP commands	→	S procedure with SDF commands

SDF-CONV writes the result of the conversion into an output procedure, which is stored as a SAM file or as an element in a PLAM library. A detailed conversion log is output to SYSLST.

### 3.1 Starting and terminating the program

The SDF-CONV utility routine is started using the command

#### START-SDF-CONV

```
MONJV = *NONE / <full-filename 1..54 without-gen-vers>  
,CPU-LIMIT = *JOB-REST / <integer 1..32767>  
,PROGRAM-MODE = *ANY / 24
```

This command calls SDF-CONV as a subsystem. The alternative of starting the SDF-CONV conversion routine by means of the command START-PROGRAM SDF-CONV continues to be supported for reasons of compatibility.

Once the program has been started, the message "CVR0060 SDF-CONV V03.0A40 READY" is displayed.

Every time a conversion run is completed, a short error listing is output (see "Output to SYSOUT", page 30), followed once more by message CVR0060 to indicate that the program is ready and waiting for further statements. This makes it possible to start several conversion runs in succession without SDF-CONV having to be restarted each time. SDF-CONV is terminated by means of the statement

```
//END
```

## 3.2 Statements

### 3.2.1 Overview of all statements

The SDF-CONV utility routine accepts the statements listed in the table below.

Statement	Brief description
CONVERT	starts the conversion of the command language and/or procedure format
END	terminates the called program
MODIFY-SDF-OPTIONS	activates/deactivates a user syntax file and changes the settings of SDF options
REMARK	identifies a given text as a comment
RESTORE-SDF-INPUT	outputs operands and operand values of the most recently accepted statement
SHOW-SDF-OPTIONS	outputs information on the active syntax files and the settings of the SDF options
STEP	defines a restart point for error handling within a sequence of statements
WRITE-TEXT	outputs a text via SYSOUT

Table 2: Statements accepted by SDF-CONV

CONVERT is the only program-specific statement. The remaining statements, which are described briefly as of page 24, are referred to as "SDF standard statements" (see the BS2000 manual "Introductory Guide to the SDF Dialog Interface" [3]).

In expert mode, the CONVERT statement is preset. This means that entry of "?" causes the operand form of the CONVERT statement to be output instead of the statement menu.

### 3.2.2 CONVERT statement

The CONVERT statement converts ISP commands stored in files or library elements into SDF commands.

```

CONVERT

FROM-FILE = <full-filename 1..54> / *LIBRARY-ELEMENT(...)
    *LIBRARY-ELEMENT(...)
        LIBRARY = <full-filename 1..54 without-gen-vers>
        ,ELEMENT = *ALL(...) / <composed-name 1..64>(...) /
            <full-filename 1..64 without-cat-gen-vers with-wild>(...)
            *ALL(...)
                VERSION = *HIGHEST-EXISTING / *ALL / <composed-name 1..24> /
                    <full-filename 1..24 without-cat-gen-vers with-wild>
                <composed-name>(...)
                VERSION = *HIGHEST-EXISTING / *ALL / <composed-name 1..24> /
                    <full-filename 1..24 without-cat-gen-vers with-wild>
                <full-filename>(...)
                VERSION = *HIGHEST-EXISTING / *ALL / <composed-name 1..24> /
                    <full-filename 1..24 without-cat-gen-vers with-wild>
            ,TYPE = ↓ / <alphanum-name 1..8>
    ,EXPECT-CONTINUATION = *STD / *NEW-MODE / *OLD-MODE
    ,PARAMETER-LINES = *CONVERT / *COPY-ONLY
    ,TO-FILE = *STD / <full-filename 1..54> / *LIBRARY-ELEMENT(...)
        *LIBRARY-ELEMENT(...)
            LIBRARY = <full-filename 1..54 without-gen-vers>
            ,ELEMENT = *SAME(...) / <composed-name 1..64>(...)
            *SAME(...)
                VERSION = *HIGHEST-EXISTING / *SAME / <composed-name 1..24>
            <composed-name>(...)
                VERSION = *HIGHEST-EXISTING / *SAME / <composed-name 1..24>
            ,TYPE = *SAME / <alphanum-name 1..8>

```

continued ➡

```

,PRODUCE-CONTINUATION = *STD / *NEW-MODE / *OLD-MODE
,REPLACE-OLD-FILE = *YES / *NO
,SYSTEM-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.U-CMD / <full-filename 1..54>
,GROUP-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.S-CMD / <full-filename 1..54> / *NO
,UNCHANGED-CMD = *NONE / list-poss(30): <structured-name 1..30>
,DOCUMENTATION = *NO / *MINIMUM / *MAXIMUM
,TARGET-VERSION = *CURRENT / *V10 / *V11 / *OSD-V1
,OUTPUT-FORM = *STD / *ACCEPTED-FORM / *INVARIANT-FORM(...)
  *INVARIANT-FORM(...)
    |
    |   CONTEXT = *ALL / list-poss(3): *BATCH / *BATCH-PROC / *DIALOG-PROC
,PROCEDURE-FORMAT = *SAME (...) / *S-PROCEDURE(...)
  *SAME(...)
    |
    |   BLOCK-INPUT = *NOT-ALLOWED / *ALLOWED
  *S-PROCEDURE(...)
    |
    |   CMD-FORMAT = *SDF / *SAME
,PROGRAM-INPUT = *DATA(...) / *STMT(...)
  *DATA(...)
    |
    |   EXCEPT-AFTER-CMD = *NONE / list-poss(30): <text 0..30>
    |   ,WARNING = *NO / *YES
  *STMT(...)
    |
    |   EXCEPT-AFTER-CMD = *NONE / list-poss(30): <text 0..30>
,BLOCK-INPUT = *NOT-ALLOWED / *ALLOWED

```

## Operand description

### FROM-FILE =

Name(s) of the SAM or ISAM file or of the element(s) from a PLAM or OSM library (input procedure) which are to be converted.

The files/library elements may be located on NK4 disks.

### FROM-FILE = <full-filename 1..54>

Name of the SAM or ISAM file to be converted.

**FROM-FILE = \*LIBRARY-ELEMENT(...)**

Name of the library and its element(s).

Elements of an OMS library must be of type S.

**LIBRARY = <full-filename 1..54 without-gen-vers>**

Name of the library containing the element or elements to be converted.

**ELEMENT =**

Specifies the name(s) of the library element or elements.

**ELEMENT = \*ALL(...)**

All elements of the specified library are to be converted. This operand value can only be specified for elements of a PLAM library.

**VERSION =**

Specifies the element version.

**VERSION = \*HIGHEST-EXISTING**

The highest existing version is used by default.

**VERSION = \*ALL**

All versions of all elements are converted. This operand value can only be specified for elements of a PLAM library.

**VERSION = <composed-name 1..24>**

Version number of the element.

**VERSION = <full-filename 1..24 without-cat-gen-vers with-wild>**

All versions which match the specified string are to be converted. This operand value can only be specified for SDF versions  $\geq$  V3.0 and only for elements of a PLAM library.

**ELEMENT = <composed-name 1..64>(…)**

Name of the library element.

If the library is an OSM library, the names of the elements it contains must not be longer than 8 characters. If more than 8 characters are entered for the element name, only the first 8 are used for selecting the element.

**VERSION=**

For possible specifications see the operand ELEMENT = \*ALL(...).

**ELEMENT = <full-filename 1..64 without-cat-gen-vers with-wild>(…)**

All elements in the library which match the specified string are to be converted. This operand value can only be specified with SDF versions  $\geq$  V3.0 and for elements in a PLAM library.

**VERSION=**

For possible specifications see the operand ELEMENT = \*ALL(...).

**TYPE = J / <alphanum-name 1..8>**

Determines the type of the element(s) of the input library.

TYPE=S must always be specified for elements of an OMS library.

**EXPECT-CONTINUATION =**

Position of the continuation character in the input procedure.

**EXPECT-CONTINUATION = \*STD**

The position of the continuation character corresponds to the current SDF setting.

This setting can be displayed by means of the SHOW-SDF-OPTIONS statement and modified by means of MODIFY-SDF-OPTIONS CONTINUATION=...

**EXPECT-CONTINUATION = \*NEW-MODE**

The position of the continuation character is expected to be behind the last character in the line.

Lines in S procedures containing commands and statements may have any length; the position of the continuation character is therefore variable.

In non-S procedures, the continuation character must be in a position preceding column 73.

**EXPECT-CONTINUATION = \*OLD-MODE**

The position of the continuation character is expected to be in column 72 exactly.

**PARAMETER-LINES =**

Determines whether command records which contain procedure parameters or in which job variables are to be replaced are converted.

**PARAMETER-LINES = \*CONVERT**

Command records which contain procedure parameters or in which job variables are to be replaced are converted. If errors occur during conversion, the commands affected are transferred unchanged to the output procedure.

**PARAMETER-LINES = \*COPY-ONLY**

Command records which contain procedure parameters or in which job variables are to be replaced are transferred unchanged to the output procedure. These commands are not provided with comments, regardless of the value set for documentation (DOCUMENTATION operand).

**TO-FILE =**

Name(s) of the SAM file or the element(s) of a PLAM library to which the conversion result (output procedure) is to be written.

**TO-FILE = \*STD**

If no file name or library element is specified, SDF-CONV automatically creates a SAM output file with the name SYSCVR.tsn.yyyy.mm.dd.hh.mm.ss. This name is made up of the prefix SYSCVR, the task sequence number, and the current date and time, and is therefore

unique. The file will not be overwritten by subsequently generated output files (regardless of the setting for the REPLACE-OLD-FILE operand). In guided dialog the generated standard file name is displayed.

**TO-FILE = <full-filename 1..54>**

Name of the output file to which the conversion result (output procedure) is to be written. If there is no file with this name in the catalog, a SAM file with this name is created. Any file with the same name in the catalog must be released for overwriting by means of the operand REPLACE-OLD-FILE=\*YES.

**TO-FILE = \*LIBRARY-ELEMENT(...)**

Name of the PLAM library and its element(s).

**LIBRARY = <full-filename 1..54 without-gen-vers>**

Name of the PLAM library which contains the element or elements to be converted.

**ELEMENT =**

Specifies the name(s) of the library element(s).

**ELEMENT = \*SAME(...)**

The name of the library element matches the name of the file or element specified in the FROM-FILE operand.

**VERSION =**

Specifies the version of the element.

**VERSION = \*HIGHEST-EXISTING**

By default the highest available version is used.

**VERSION = \*SAME**

The version of the output library element matches that of the input library element or, if a SAM or ISAM file is to be converted, of the highest available version.

**VERSION = <composed-name 1..24>**

Version number of the element.

**ELEMENT = <composed-name 1..64>(...)**

Name of the library element.

**VERSION =**

For possible specifications see the operand ELEMENT = \*SAME(...).

**TYPE = \*SAME / <alphanum-name 1..8>/**

Determines the type of the element(s) of the output library.

If the name of an SAM or ISAM file is specified for the FROM-FILE operand, the elements of the output library are automatically of type J.



**PRODUCE-CONTINUATION =**

Position of the continuation character in the output procedure.

If the output procedure is to be an S procedure, a variable format is used. Continuation characters are written as described under PRODUCE-CONTINUATION=\*NEW-MODE.

**PRODUCE-CONTINUATION = \*STD**

The position of the continuation character corresponds to the current SDF setting.

This setting can be displayed by means of the SHOW-SDF-OPTIONS statement and modified by means of MODIFY-SDF-OPTIONS CONTINUATION=...

**PRODUCE-CONTINUATION = \*NEW-MODE**

The continuation character is written behind the last character in the line.

**PRODUCE-CONTINUATION = \*OLD-MODE**

The continuation character is written in column 72 exactly.

**REPLACE-OLD-FILE = \*YES / \*NO**

Specifies whether an existing output file or a library element may be overwritten (\*YES) or not (\*NO).

**SYSTEM-SYNTAX-FILE =**

Name of the (system) syntax file used to convert user commands.

**SYSTEM-SYNTAX-FILE = \$.SYSSDF-CONV.030.USER.U-CMD**

Preset (system) syntax file for converting user commands.

**SYSTEM-SYNTAX-FILE = <full-filename 1..54>**

Name of the (system) syntax file to be used to convert user commands.

**GROUP-SYNTAX-FILE =**

Name of the (group) syntax file used for converting privileged commands.

**GROUP-SYNTAX-FILE = \$.SYSSDF-CONV.030.USER.S-CMD**

Preset (group) syntax file used for converting privileged commands.

**GROUP-SYNTAX-FILE = <full-filename 1..54>**

Name of the (group) syntax file to be used for converting privileged commands.

**GROUP-SYNTAX-FILE = \*NO**

The privileged commands are not to be converted. If there is no (group) syntax file for the user, conversion will be aborted unless \*NO is specified.

**UNCHANGED-CMD =**

The conversion of one or more commands can be suppressed. If SDF-CONV encounters such a command, the command record is transferred to the output procedure unchanged. In addition, an error message with the weight NOTE is written to the conversion log. You should bear in mind that the only commands that will be recognized as commands that are to be excluded from conversion are those written exactly as specified in the UNCHANGED-CMD operand.

*Example:*

In order to exclude the ERASE command from conversion, all the various forms used (in the input procedure) must be specified. This could be, for example: ERASE, ERAS, ER.

**UNCHANGED-CMD = \*NONE**

No commands are to be excluded from conversion.

**UNCHANGED-CMD = list-poss(30): <structured-name 1..30>**

The specified commands are excluded from conversion right from the outset.

Up to 30 commands or alternative command forms can be specified.

If the output procedure is to be an S procedure, the PROCEDURE command is converted accordingly even if specified in the UNCHANGED-CMD operand.

**DOCUMENTATION =**

Successful conversion of the command language causes inline comments in commands from the input procedure to be lost, with the exception of those inline comments written before the command name.

The output procedure identified by TO-FILE can now contain additional information in the form of comment lines. The scope of this information depends not only on the value of the DOCUMENTATION operand but also on the value set for the UNCHANGED-CMD operand (see Table 4, "Documentation level", page 29): no additional comment lines are generated for any commands specified with the UNCHANGED-CMD operand.

Procedure parameters contained in the additionally generated comment lines are not replaced.

**DOCUMENTATION = \*NO**

Command records from the input procedure which have been identified as comment lines are not included in the output procedure.

**DOCUMENTATION = \*MINIMUM**

Command records from the input procedure which contain inline comments or which consist of more than one input line are passed to the output procedure for documentation purposes. These records appear (as comment lines) in front of the lines containing the conversion result. If it was not possible to convert the command record because conversion is context-dependent or runtime-dependent, a conversion proposal is written in the comment line. The comment lines have the following format:

```
/"*****" REMARK <command record of input procedure or conversion proposal>
```

This rule does not apply to command records containing any of the following commands: BEGIN-PROCEDURE, PROCEDURE, REMARK (only if in non-S procedures), SET-PROCEDURE-OPTIONS, BEGIN-PARAMETER-DECLARATION, DECLARE-PARAMETER, END-PARAMETER-DECLARATION or their guaranteed abbreviations.

**DOCUMENTATION = \*MAXIMUM**

All command records from the input procedure are also written (as comment lines) in front of the line containing the conversion result. If it was not possible to convert the command record because conversion is context-dependent or runtime-dependent, a conversion proposal is written in the comment line. The comment lines have the following format:

```
/"*****" REMARK <command record of input procedure or conversion proposal>
```

This rule does not apply to command records containing any of the following commands: BEGIN-PROCEDURE, PROCEDURE, REMARK (only if in non-S procedures), SET-PROCEDURE-OPTIONS, BEGIN-PARAMETER-DECLARATION, DECLARE-PARAMETER, END-PARAMETER-DECLARATION or their guaranteed abbreviations.

**TARGET-VERSION =**

Defines which BS2000 version the procedure is to be able to run under.

**TARGET-VERSION = \*CURRENT**

The converted procedure is to be able to run under the BS2000 version which is loaded when conversion is performed.

**TARGET-VERSION = \*V10**

The converted procedure is to be able to run under BS2000 V10. This means that the conversion of the ISP commands ABEND, FILE and LOGOFF for BS2000 V10 is subject to greater restrictions than are necessary for BS2000 V11 and later versions. For further details see "Non-convertible operand configurations and conditional conversions", page 36, and Tables13 and14 in the appendix (as of page 83).

**TARGET-VERSION = \*V11**

The converted procedure is to be able to run in BS2000 V11 (≡ BS2000/OSD-BC V1.0). As of V11, "BS2000 V11.0" has been renamed "BS2000/OSD-BC V1.0". However, the old name remains valid as a synonym (see the list of amendments given on page 3). For this reason, both \*V11 and \*OSD-V1 are permitted as operand values.

**TARGET-VERSION = \*OSD-V1**

The converted procedure is to be able to run in BS2000 V11 ( $\equiv$  BS2000/OSD-BC V1.0). See also the description of TARGET-VERSION = \*V11.

**OUTPUT-FORM =**

Defines the way in which optional operands (i.e. operands with preset values) which are not specified explicitly in the command record are handled.

Regardless of what is specified for OUTPUT-FORM, all specified operands and operand values are converted to the corresponding SDF operands and operand values, provided this is not prevented by prevailing restrictions (e.g. by the UNCHANGED-CMD operand or any of the restrictions listed under "Restrictions", page 32ff).

**OUTPUT-FORM = \*STD**

ISP operands and operand values that are not specified are ignored during conversion, i.e. SDF-CONV only converts explicitly specified ISP operands and operand values into the corresponding SDF formats.

When the converted procedure is executed, all operands which have not been specified are set to their default values.

*Note*

It is possible to modify default values in the syntax files. This may mean that the results of command execution differ.

**OUTPUT-FORM = \*ACCEPTED-FORM**

All specifications of SDF operands and operand values are expanded to their complete names.

This operand value can only be specified with SDF versions  $\geq 4.0$ .

**OUTPUT-FORM = \*INVARIANT-FORM(...)**

ISP operands and operand values that are not specified are replaced by the corresponding SDF forms in the course of conversion and assigned their default values.

In this context, users should bear in mind that not all SDF commands with their operands and default values can be executed independently of the task type and the method of execution (called either directly or within a (second) procedure). In such cases, it is vital to specify the environment (task type/execution type) in which the converted procedure is to run, as errors may result in the program being aborted (context-dependency).

This operand value can only be specified with SDF versions  $\geq 3.0$ .

**CONTEXT =**

This operand allows you to specify an environment in which the converted procedure is to run. All SDF operands and default values which are guaranteed to run in the specified environment are generated.

Furthermore, it is possible to specify up to three different environments for which all matching operands and default values are to be replaced. Operands or default values that are not supported by all specified environments are not preset.

**CONTEXT = \*ALL**

The commands are expanded to include the operands and default values guaranteed to run in all possible environments. In this case, that would be the following environments:

in interactive mode

- guided dialog
- unguided dialog
- procedure called from the terminal (DIALOG-PROC)

in batch mode

- in the ENTER file (BATCH)
- in a procedure called by the ENTER file (BATCH-PROC)

**CONTEXT = \*BATCH**

The output procedure is to be able to run in (simple) batch mode.

**CONTEXT = \*BATCH-PROC**

The output procedure is to be able to run in a procedure called in batch mode.

**CONTEXT = \*DIALOG-PROC**

The output procedure is to be able to run in a procedure called from the terminal in interactive mode.

**PROCEDURE-FORMAT =**

Determines the type of conversion to be applied to the input procedure(s).

**PROCEDURE-FORMAT = \*SAME(...)**

The output procedure is to have the same format as the input procedure.

The commands of input procedures are to be converted into SDF commands.

**BLOCK-INPUT =**

Determines how the SDF commands generated by SDF-CONV are to be written to the output procedure.

S procedures allow more than one command to be written continuously (as a block) into a single command line, individual command records being separated by semicolon, while in non-S procedures, each command must be written into a separate line since there are no command separators.

**BLOCK-INPUT = \*NOT-ALLOWED**

SDF-CONV writes each command into a separate line in the output procedure.

**BLOCK-INPUT = \*ALLOWED**

If SDF-CONV converts single ISP commands into more than one SDF command, these can be written continuously into a single command line.

Any specification of this operand value is ignored during the conversion of commands from non-S procedures.

*Example*

/ISP1	may be converted into /SDF11;SDF12
/ISP2;ISP3;ISP4	into /SDF21;SDF22;SDF23;SDF3;SDF41;SDF42;SDF43

**PROCEDURE-FORMAT = \*S-PROCEDURE(...)**

The input procedure is to be converted into an S procedure. The following options can be specified: conversion of input procedure commands into SDF commands (CMD-FORMAT operand), conversion of data lines into statement lines (PROGRAM-INPUT operand), definition of the output procedure format (BLOCK-INPUT operand).

**CMD-FORMAT =**

Specifies whether the commands of the input procedure are to be converted into SDF commands or not.

**CMD-FORMAT = \*SDF**

The commands of the input procedure are to be converted into SDF commands as specified with the OUTPUT-FORM operand.

If the input procedure is a non-S procedure and the PROCEDURE command is specified with the UNCHANGED-CMD operand, this specification is ignored.

**CMD-FORMAT = \*SAME**

The commands of the input procedure are to be retained unchanged. Nevertheless, the representations of procedure parameters are adapted to S procedure conventions.

*Example*

/FS &FILENAME	is converted into /FS &(FILENAME)
/REMARK replace &LIB	is converted into /REMARK 'replace &&(LIB)'

**PROGRAM-INPUT =**

Determines whether data lines are to be converted into SDF statement lines during conversion of a non-S procedure into an S procedure.

In non-S procedures, statement lines can have the same format as data lines, i.e. they can be entered without leading slashes, while in S procedures, statement lines must always start with two slashes.

In any case, the representations of procedure parameters are adapted to S procedure conventions (see example under CMD-FORMAT=\*SAME).

**PROGRAM-INPUT = \*DATA(...)**

Lines without leading slashes in the input procedure are written to the output procedure unchanged i.e. as data lines.

**EXCEPT-AFTER-CMD =**

Determines whether individual data lines are to be excepted from the general handling of data lines.

**EXCEPT-AFTER-CMD = \*NONE**

All data lines are to be transferred to the output procedure unchanged.

**EXCEPT-AFTER-CMD = list-poss(30): <text 1..30>**

Permits exceptional conversion of individual data lines into SDF statement lines. Any data line following a command line containing at least one of the elements listed will then be converted into an SDF statement line; this exceptional handling will be continued until the next command line containing none of the elements listed.

**WARNING = \*NO / \*YES**

Determines whether a WARNING message is to be issued (\*YES) or not (\*NO) each time SDF-CONV detects a data line during conversion.

**PROGRAM-INPUT = \*STMT(...)**

Lines without leading slashes in the input procedure are converted into SDF statement lines by prefixing two slashes to each of these lines.

**EXCEPT-AFTER-CMD =**

Determines whether individual data lines are to be excepted from the general handling of data lines.

**EXCEPT-AFTER-CMD = \*NONE**

All data lines are to be converted into SDF statement lines.

**EXCEPT-AFTER-CMD = list-poss(30): <text 1..30>**

Permits individual data lines to be excepted from conversion into SDF statement lines i.e. are to be written to the output procedure unchanged. Any data line following a command line containing at least one of the elements listed will then be transferred unchanged; this exceptional handling will be continued until the next command line containing none of the elements listed.

**BLOCK-INPUT =**

Determines how the SDF commands generated by SDF-CONV are to be written to the output procedure.

S procedures allow more than one command to be written continuously (as a block) into a single command line, individual command records being separated by semicolon, while in non-S procedures, each command must be written into a separate line since there are no command separators.

**BLOCK-INPUT = \*NOT-ALLOWED**

SDF-CONV writes each command into a separate line in the output procedure.

**BLOCK-INPUT = \*ALLOWED**

If SDF-CONV converts single ISP commands into more than one SDF command, these can be written continuously into a single command line.

Any specification of this operand value is ignored during the conversion of commands from non-S procedures.

For an example see PROCEDURE-FORMAT=\*SAME(BLOCK-INPUT=\*ALLOWED).

### 3.2.3 SDF standard statements

Programs that read in their statements via the SDF interface have what are called SDF standard statements. These statements are independent of program-specific statements in the way that they function.

For detailed explanations, operand descriptions and examples, see the BS2000 manual "Introductory Guide to the SDF Dialog Interface" [3].

#### END

The END statement terminates the input to the called program.

The statement has no operands and is executed immediately.

<b>END</b>

#### MODIFY-SDF-OPTIONS

The MODIFY-SDF-OPTIONS statement enables the user to activate or deactivate a user syntax file during the program run, and to modify SDF settings. The functionality of the statement is the same as that of the command of the same name. (Test mode, however, can be set only at command level.)

<b>MODIFY-SDF-OPTIONS</b>
<pre> <b>SYNTAX-FILE</b> = *<u>UNCHANGED</u> / *<b>NONE</b> / &lt;full-filename 1..54&gt; <b>,GUIDANCE</b> = *<u>UNCHANGED</u> / *<b>EXPERT</b> / *<b>NO</b> / *<b>MAXIMUM</b> / *<b>MEDIUM</b> / *<b>MINIMUM</b> <b>,LOGGING</b> = *<u>UNCHANGED</u> / *<b>INPUT-FORM</b> / *<b>ACCEPTED-FORM</b> / *<b>INVARIANT-FORM</b> <b>,UTILITY-INTERFACE</b> = *<u>UNCHANGED</u> / *<b>OLD-MODE</b> / *<b>NEW-MODE</b> <b>,PROCEDURE-DIALOGUE</b> = *<u>UNCHANGED</u> / *<b>YES</b> / *<b>NO</b> <b>,CONTINUATION</b> = *<u>UNCHANGED</u> / *<b>OLD-MODE</b> / *<b>NEW-MODE</b> <b>,MENU-LOGGING</b> = *<u>UNCHANGED</u> / *<b>NO</b> / *<b>YES</b> </pre>



**REMARK**

The REMARK statement identifies the specified text as a comment and is significant only for documenting the execution of the job or procedure.

<b>REMARK</b>
<b>TEXT</b> = <cmd-rest 0..1800>

**RESTORE-SDF-INPUT**

The RESTORE-SDF-INPUT statement causes the most recently accepted statement to be redisplayed on the screen. This statement is effective only in a guided or temporarily guided dialog and in the NO mode of an unguided dialog. It cannot be used in procedure mode. In a (temporarily) guided dialog, the operand form is displayed together with all entered or preset operand values. In the NO mode of an unguided dialog, the explicitly entered values are output once again. RESTORE-SDF-INPUT enables the user to correct the contents of a statement or to reuse the same statement with modified operand values.

<b>RESTORE-SDF-INPUT</b>

**SHOW-SDF-OPTIONS**

The SHOW-SDF-OPTIONS statement enables the user to request current information about all activated syntax files and SDF settings for the executing task. The functionality of the statement is the same as that of the command of the same name.

<b>SHOW-SDF-OPTIONS</b>

## STEP

The STEP statement can be used to subdivide a sequence of statements in a procedure file into sections. If an errored statement is encountered, the spin-off mechanism is triggered. This means that all succeeding statements are ignored until a STEP statement is encountered. If no STEP statement is found before the END statement, the program is sent a return code to which it can respond with an abnormal termination. In the event of abnormal program termination, the spin-off mechanism resumes at command level (see the SET-JOB-STEP command in the manual "User Commands (SDF Format), Part 2" [2]).

<b>STEP</b>

## WRITE-TEXT

The WRITE-TEXT statement writes the specified text on SYSOUT.

<b>WRITE-TEXT</b>
<b>TEXT</b> = ' ' / <c-string 1..1024 with-low>

### 3.2.4 Control functions

The following inputs are used for dialog guidance control. They are accepted in the statement part of SDF-CONV and executed immediately.

Functions	Brief description
<statement> ?	SDF-CONV switches to (temporarily) guided dialog
<statement> .. <operand> = ?	SDF-CONV provides information about the function of the operand and the possible operand values
K1 key	returns from (temporarily) guided dialog to unguided dialog
K2 key	switches to command level; the RESUME-PROGRAM command is used to return to SDF-CONV

Table 3: Control functions in the statement part of SDF-CONV

A full overview and a description of all possible inputs in a (temporarily) guided and an unguided dialog are given in the BS2000 manual "Introductory Guide to the SDF Dialog Interface" [3].

## 3.3 Outputs

### 3.3.1 Output procedure

The TO-FILE operand of the CONVERT statement defines the name of the output procedure or, if two or more input procedures stored in library elements are to be converted, the names of the output procedures.

In this context, it is vital to remember that the FROM-FILE and TO-FILE values must always be different. Regardless of the format of the input procedure, a cataloged SAM file or PLAM library element is always created for the output procedure. The operand REPLACE-OLD-FILE = \*YES can be used to specify that an existing file or a library element with the same name can be overwritten. As of SDF-CONV V3.0, no empty files or library elements are generated. This means that in the event of aborting the conversion before writing any command record to the output procedure file specified with TO-FILE, no output procedure is generated.

The output procedure contains the converted commands or the ISP command records taken over unchanged from the input procedure.

If SDF command records form part of the input procedure, the specified commands are expanded to their full command names.

Depending on the values of the DOCUMENTATION and UNCHANGED-CMD operands, the output procedure may also contain additionally added comment lines (see the table on page 29). These additionally inserted comment lines may be useful to the user in the following cases:

1. For manually post-editing the converted procedure (output procedure): certain ISP commands (e.g. JOIN) cannot be converted automatically as they have various SDF equivalents, the appropriate one being determined by the context. In the comment line, SDF-CONV makes a suggestion as to the appropriate conversion. This comment line is written in front of the command line with the unconverted command. The output procedure then contains the following lines:

```
/"*****" REMARK <sdf-cmd = conversion proposal>  
/<isp-cmd>
```

2. For retaining inline comments in an ISP command record: If SDF-CONV encounters a command record with inline comments, the command, the operands and the operand values are converted but the comment is not transferred. Insertion of the ISP command record ensures that the information contained in the comment is not lost and can be incorporated in the command record manually if required. The output procedure contains the following lines:

```
/"*****" REMARK <isp-cmd "with inline comment">  
/sdf-cmd
```

The following table shows the scope of the output procedure depending on the values set for the UNCHANGED-CMD and DOCUMENTATION operands. This is illustrated by way of an example based on two ISP commands:

```
/SYSFILE SYSLST=()           → /ASSIGN-SYSLST TO-FILE=*PRIMARY
/ERASE OTTO "DELETE"        → /DELETE-FILE FILE-NAME=OTTO
```

DOC	UNCHANGED-CMD = *NO <sup>1)</sup>	UNCHANGED-CMDS = isp-cmd <sup>2)</sup>
<b>*NO</b>	/sdf-cmd ... The command is converted.  /ASSIGN-SYSLST TO-FILE=*PRIMARY  /DELETE-FILE FILE-NAME=OTTO	/isp-cmd ... The command is not converted.  /SYSFILE SYSLST=()  /ERASE OTTO "DELETE"
<b>*MIN</b>	/REMARK isp-cmd ... (not always) /sdf-cmd ... The command is converted. If the command is longer than one line or contains an inline comment, SDF-CONV additionally writes this record to an inserted comment line.  /ASSIGN-SYSLST TO-FILE=*PRIMARY  /REMARK ERASE OTTO "DELETE" /DELETE-FILE FILE-NAME=OTTO	/REMARK sdf-cmd ... /isp-cmd ... The command is not converted, but SDF- CONV writes a conversion proposal in an inserted comment line.  /REMARK ASSIGN-SYSLST TO-FILE=*PRIMARY /SYSFILE SYSLST=()  /REMARK DELETE-FILE FILE-NAME=OTTO /ERASE OTTO "DELETE"
<b>*MAX</b>	/REMARK isp-cmd ... /sdf-cmd ... The command is converted. In addition, SDF-CONV writes the command record to an inserted comment line.  /REMARK SYSFILE SYSLST=() /ASSIGN-SYSLST TO-FILE=*PRIMARY  /REMARK ERASE OTTO "DELETE" /DELETE-FILE FILE-NAME=OTTO	/REMARK sdf-cmd ... /isp-cmd ... The command is not converted, but SDF-CONV writes a conversion proposal to an inserted comment line.  /REMARK ASSIGN-SYSLST TO-FILE=*PRIMARY /SYSFILE SYSLST=()  /REMARK DELETE-FILE FILE-NAME=OTTO /ERASE OTTO "DELETE"

Table 4: Documentation level in the output procedure depending on the values set for the UNCHANGED-CMD and DOCUMENTATION operands

- 1) Provided conversion of the ISP command isp-cmd is not subject to any restrictions.
- 2) Or: the ISP command cannot be converted because its meaning is context-dependent or runtime-dependent.

### 3.3.2 Output to SYSOUT

Following every conversion run, SDF-CONV outputs an overview of the number of errors, warnings and comments detected in the course of conversion. This overview has the following format (i = number):

```
CVR1000 CONVERSION TERMINATED WITH:  
CVR1001          FAILURES : i  
CVR1002          ERRORS   : i  
CVR1003          WARNINGS: i  
CVR1004          NOTES    : i
```

FAILURES and ERRORS denote fatal (DMS) errors. When such errors occur, the error message is output both to SYSLST (see below) and also to SYSOUT. The SDF-CONV conversion run may be aborted after error messages of this type. SDF-CONV outputs the overview and waits for further statements after issuing the message "CVR0060 SDF-CONV V03.0A40 READY".

Details of the format and contents of the messages can be found under "SDF-CONV messages" in the appendix.

### 3.3.3 Conversion log to SYSLST

In addition to generating the output procedure, SDF-CONV writes a conversion log and outputs it automatically to SYSLST. The log is in three parts:

- Part 1. List of the operand values currently valid for the SDF-CONV run performed.
- Part 2. List of all records in the input file and of all messages relating to errors that occurred during conversion.
- Part 3. List detailing the number of errors and notes (this is the same as the list output to SYSOUT).

The second part of the log records messages in the following cases:

- there is no equivalent SDF command for the ISP command
- the ISP command cannot be unequivocally assigned to an SDF command
- when specified with certain operands, some ISP commands have no SDF equivalent
- a syntax error has been found

Command records containing such errors are not converted; instead they are copied unchanged to the output procedure. If an ERROR or FAILURE is reported before conversion starts, the second part of the log is not generated. The error messages are listed under "SDF-CONV messages" in the appendix.

The system file SYSLST can be assigned to a cataloged file by means of the ASSIGN-SYSLST command or, as of BS2000 V11.0, copied using the COPY-SYSTEM-FILE command.

### 3.3.4 Program monitoring using monitoring job variables

SDF-CONV can supply a program-monitoring job variable with the following values:

'\$T 0000'	Program terminated normally. Conversion without notes.
'\$T 0001'	Program terminated normally. Conversion with notes.
'\$T 1002'	Program terminated normally. Conversion with warnings.
'\$T 1003'	Program terminated normally. Conversion with errors.
'\$A 3004'	Program terminated abnormally due to serious error, or CPU time limit exceeded.

The program-monitoring job variable must be declared in the START-SDF-CONV command (or, for the compatible call, in the START-PROGRAM-SDF-CONV command). For further information see the manual "User Commands (SDF Format), Part 2" [2] or "Job Variables" [4].

## 3.4 Restrictions

SDF-CONV cannot convert all ISP commands and their operands into SDF commands with corresponding operands. Certain commands, operands and operand configurations are excluded from conversion right from the start. Moreover, there are certain so-called "conditional conversions" which are dependent on the operating system version. The conversion of procedure parameters as operands can also be subject to restrictions, as well as the conversion of non-S procedures into S procedures.

The command records concerned are transferred unchanged to the output procedure (SDF-CONV outputs an error message to SYSLST) or, if the condition regarding version dependency was fulfilled, are converted.

### 3.4.1 Basic preconditions and restrictions

- Error-free procedures in ISP format are a basic prerequisite for producing error-free procedures in SDF format.
- SDF-CONV only converts ISP commands that were defined in BS2000 versions  $\leq V10$ .
- The FROM-FILE and TO-FILE operands must always be assigned different values. This applies even if the operand REPLACE-OLD-FILE = \*YES is specified.
- Every command in the input procedure that is not described in one of the syntax files is copied unchanged to the output procedure. An error message is output to SYSLST.
- Card readers ceased to be supported by BS2000 with the release of V10.0. This means that all commands which refer to such devices are copied unchanged to the output procedure.
- Operands to which no explicit values have been assigned can be generated for SDF commands. These operands implicitly receive the default value specified in the operand description of the relevant command (see the manuals "User Commands (SDF Format)" Part 1 [1] and Part 2 [2]).
- Procedure parameters must not be part of a command or operand and must not themselves be either a command or an operand.

#### *Example*

```
/PROC N,(&PARAM1 = 'SHOW', &PARAM2 = 'FILE')  
/...  
/&PARAM1-SDF-OPTIONS  
/CREATE-FILE &PARAM2 = OTTO.DAT  
/...  
/ENDP
```



- Commands containing procedure parameters can be converted to SDF format only if their conversion is not dependent on the parameters. In all other cases no one-to-one assignment can be made, and so the commands are not converted (cf. page 38ff).

*Example*

```
/PROC N,(&USE1)
/...
/RFD USE=&USE1
/...
/ENDP
```

```
&USE1 = NO           is converted in SDF to: /STOP-DISKETTE-INPUT
&USE1 = INPUT       is converted in SDF to: /START-DISKETTE-INPUT
```

- If the input procedure contains the SDF command MODIFY-SDF-OPTIONS, the value of the CONTINUATION operand must not be a procedure parameter.
- If the input procedure contains the SDF command ENDP-RESUME and the output procedure is to be an S procedure, the command is converted into EXIT-PROCEDURE RESUME=YES.  
If the output procedure is to be a non-S procedure, the command is not converted since the procedure's spin-off mechanism is different in this case.
- As of SDF V3.0, all names beginning with an asterisk and containing no other wildcards are interpreted as keywords.
- The old label format (.label) is not converted into the new label format (label:).
- SDF-CONV does not recognize nested procedure parameters or nested replaceable job variables when converting ISP commands from S procedures into SDF commands, so that the command records concerned are not converted.
- Successful conversion of the command language causes inline comments in commands from the input procedure to be lost. This can be avoided by specifying the CONVERT statement operand DOCUMENTATION=\*MIN/\*MAX, which causes inline comments to be transferred to the output procedure.

### 3.4.2 Restrictions on the conversion into S procedures

- ENTER files cannot be converted into S procedures by means of SDF-CONV.
- SDF-CONV reports an error if the name of a procedure parameter exceeds 20 characters. While the names of procedure parameters in non-S procedures may be up to 255 characters long, a maximum length of 20 characters applies to S procedures. This restriction does not apply to positional operands; these are automatically abbreviated by SDF-CONV, if required.
- Command lines in non-S input procedures are only evaluated up to the 72nd character; all characters from column 73 onward are ignored.
- Conversion may affect the logging control. While in non-S procedures, logging is determined exclusively by the called procedure, in S procedures, it is also dependent on the relevant setting for the calling environment.
- No conversion takes place if more than one job variable has to be replaced in any value of any procedure parameter.
- If the first significant character following a command name in a non-S input procedure is an equal sign, SDF-CONV issues a WARNING, except in the event of command records containing the REMARK command.
- SDF-P does not support recursive replacement of procedure parameters.

#### *Example*

1. Non-S procedure:

```
/BEGIN-PROC A,PAR=YES(PROC-PAR=(&TEST=))
/MOD-JV JV(JV-NAME=JOBVAR),VALUE='&&TEST'
/SHO-JV JV(JV-NAME=JOBVAR)
/REMARK &TEST
/WR-TEXT '&(JOBVAR)'
/EXIT-PROC
```

Trace listing of the non-S procedure:

```
/CALL-PROC EXA-NON-S
%/BEGIN-PROC A,PAR=YES(PROC-PAR=(&TEST=))
%/MOD-JV JV(JV-NAME=JOBVAR),VALUE='&&TEST'
%/SHO-JV JV(JV-NAME=JOBVAR)
%&TEST
%/REMARK &TEST
%&TEST=hallo
%/REMARK HALLO
%/WR-TEXT 'HALLO'
HALLO
%/EXIT-PROC
```

The job variable JOBVAR is replaced in two steps.

JOBVAR contains the value of the procedure parameter TEST i.e. HALLO.

## 2. S procedure:

```

/SET-PROC-OPTIONS LOG=YES
/BEGIN-PAR-DEC
/DECLARE-PAR NAME=TEST (INIT=*PROMPT)
/END-PAR-DEC
/MOD-JV JV(JV-NAME=JOBVAR),VALUE='&&TEST'
/SHO-JV JV(JV-NAME=JOBVAR)
/REMARK &TEST
/WR-TEXT '&(JV('JOBVAR'))'

```

## Trace listing of the S procedure:

```

/CALL-PROC EXA-S,LOG=YES
% 1 1 /SET-PROCEDURE-OPTIONS LOGGING-ALLOWED=YES,INTERRUPT-
% ALLOWED=YES,DATA-ESCAPE-CHAR=NONE,SYSTEM-FILE-CONTEXT=OWN,-
% DATA-ERROR-HANDLING=NO
% 1 1 /BEGIN-PARAMETER-DECLARATION
% 1 1 /DECLARE-PARAMETER NAME=TEST,TYPE=ANY,INIT=*PROMPT
% 1 1 /END-PARAMETER-DECLARATION
% 6 1 /MOD-JV JV(JV-NAME=JOBVAR),VALUE='&TEST'
% 7 1 /SHOW-JV JV(JV-NAME=JOBVAR)
%&TEST
%TEST: hallo
% 8 1 /REMARK 'HALLO'
% 9 1 /WRITE-TEXT '&TEST'
&TEST
% 11 1 /IF-BLOCK-ERROR
% 11 1 /END-IF
% 11 1 /EXIT-PROCEDURE

```

The job variable JOBVAR is replaced in a single step.  
JOBVAR contains &TEST.

### 3.4.3 Commands that cannot be converted

ISP command	Explanation
<b>– for non-privileged users</b>	
ACCOUNT	This command has not been available since BS2000 V8.5
AUDIT	No one-to-one assignment is possible (conversion is runtime-dependent)
DATA	The command has no SDF equivalent
DIAG	This is an IDA command, not an ISP command.
DO	The command may collide with global parameters (conversion is context-dependent); a conversion proposal can be found in a comment line of the output procedure, provided DOCUMENTATION = *MIN/*MAX is set
END	This command has no SDF equivalent
PARAMETER	This command has no SDF equivalent
<b>– for system administration</b>	
JOIN	No one-to-one assignment is possible (conversion is context-dependent); a conversion proposal can be found in a comment line of the output procedure of the output procedure, provided DOCUMENTATION = *MIN/*MAX is set
LOADAID	The command has no SDF equivalent
LOADAIDSYS	The command has no SDF equivalent
RCARD	Card readers are no longer supported
SDVC	No one-to-one assignment possible (conversion is context-dependent)

continued ➡

ISP command	Explanation
<b>– for the operator at the console</b>	
AGOGO ASR ASTOP BCACT BCAPPL BCASP BCCONN BCCONP BCDAC BCDISCON BCDISP BCEND BCGEN BCIN BCLOSE BCMAP BCMOD BCMOFF BCMON BCOPTION BCOUT BCSET BCSHOW BCSWP BCTIMES BCXAF BROADCAST CONSOLE DADM DCDIAG DCSTART MESSAGE MRSEND MRSMOD MRSSTART SHUTDOWN TURN	These operator commands can only be entered at the console, not at a terminal.

Table 5: Non-convertible commands for non-privileged and privileged users

### 3.4.4 Non-convertible operand configurations and conditional conversions

The following ISP commands are not converted, or only converted if certain conditions are fulfilled, when the listed operands are specified.

ISP command	Non-convertible operands or operands which restrict conversion	Explanation
<b>– for non-privileged users</b>		
ABEND	dependent on the BS2000 version	the command is not converted unless the operand TARGET-VERSION=*V11/ *OSD-V1 in the CONVERT statement was set during conversion
FILE	Dependent on the BS2000 version  LINK  *DUMMY, LINK, TSET together	the command is not converted unless the operand TARGET-VERSION=*V11/ *OSD-V1 in the CONVERT statement was set during conversion, except when:  the LINK operand was specified and other operands apart from the file name were specified or  these three operands were specified in a FILE command
LOGOFF	dependent on the BS2000 version	the command is not converted unless the operand TARGET-VERSION=*V11/ *OSD-V1 in the CONVERT statement was set during conversion
LOGON	dependent on the BS2000 version	the command is not converted unless the operand TARGET-VERSION=*V11/ *OSD-V1 in the CONVERT statement was set during conversion

continued ➡

ISP command	Non-convertible operands or operands which restrict conversion	Explanation
PRINT	(file1 , file2 , ...)  CONTROL=12LPI PNAME=	Format3 (= multiple files) is converted only if the files are specified with a fully or partially qualified name or with their <eam-file-numbers>; specification of more than one *SYSLSTnn file is not supported  the operand is no longer supported  any hyphen contained in a name is deleted without substitution during conversion into PRINT-FILE SPOOLOUT-NAME=...
PUNCH	(file1 , file2 , ...)  <eam-file-number>  TAPE=POOLER	Format3 (= multiple files) has no SDF equivalent  file names consisting solely of digits are not permitted  pooler tapes are not supported as of BS2000 V11
RESUME	with operands	is an IDA command
SECURE	WORK=...	the operand has no SDF equivalent
SYSFILE	FILE=...	the operand is too complex, one-to-one assignment is not possible; conversion is carried out only if the file name is specified in the following form: FILE=<full-filename-without-gen-vers>
TCHNG	CORR=C'a'	the operand is no longer supported

continued ➡

<b>ISP command</b>	<b>Non-convertible operands or operands which restrict conversion</b>	<b>Explanation</b>
<b>– for system administration</b>		
CANCEL	KILL dependent on the BS2000 version	the operand is not converted unless the operand TARGET-VERSION=*V11/ *OSD-V1 in the CONVERT statement was set during conversion
EXCAT	QUIET	the operand is no longer supported
IMCAT	FORM=CAT  RESET= BUFNUM=	the operand is no longer supported; a conversion proposal can be found in the comment line of the output procedure provided DOCUMENTATION = *MIN/ *MAX is set  the operands are no longer supported
RFD	USERID=userid OWNERID='ownerid'	the keywords must be specified

Table 6: Non-convertible operand configurations and conditional conversions



### 3.4.5 Restrictions on procedure parameters as operands

The following examples illustrate the difficulties that arise when ISP commands containing procedure parameters as operands are to be converted to SDF format.

1. The SDF command is determined by the ISP operand value

```
/CATJV ... ,STATE=UPDATE → /MODIFY-JV-ATTRIBUTES ...
/CATJV ... ,STATE=NEW → /CREATE-JV ...
/CATJV ... ,STATE=&OP1 → /?
```

2. The SDF operand value is determined by the ISP operand value

```
/EXECUTE ... ,SYMTEST=NO → /START-PROGRAM ... ,TEST-OPTIONS=NONE
/EXECUTE ... ,SYMTEST=ALL → /START-PROGRAM ... ,TEST-OPTIONS=AID
/EXECUTE ... ,SYMTEST=&OP2 → /START-PROGRAM ... ,TEST-OPTIONS=?

/GETJV <name> → /SHOW-JV JV-ID=JV-NAME(JV-NAME=<name>)
/GETJV *<name> → /SHOW-JV JV-ID=JV-LINK(JV-LINK=<name>)
/GETJV &OP3 → /SHOW-JV JV-ID=?
```

#### *Note*

Command records containing procedure parameters, or in which job variables are to be replaced, can be excluded from conversion by specifying the operand `PARAMETER-LINES = *COPY-ONLY` in the `CONVERT` statement. Instead of converting the command records concerned, `SDF-CONV` writes them unchanged to the output procedure. No messages are issued in such cases.

The following table provides an overview of the commands that are not converted if the listed positional and/or keyword operands are specified as procedure parameters.

`<i>` means that the *i*-th positional operand must not be a parameter.  
`OPERAND=` means that the positional operand value must not be a parameter.

ISP command	No parameter as positional operand ...	No parameter as a value of the keyword operand ...
<b>– for non-privileged users</b>		
CANCEL	<2> ,<3>	
CATJV		SHARE= , STATE=
COPY		DIALOG= , WRITE=
DELON	<1>	
ENTER		LOG= , TIME=
ERAJV		all operands
ERASE	<1> ,<2>	LIST= , POS=
EXECUTE	<1>	CLASSII= , IDA= , SYMTEST=
FSTATUS	<2>	CRDATE= , EXDATE= , EXTENDS= , FREESIZE= , FROM= , LADATE= , LIST= , PASS= , SHARE= , SIZE= , SORT= , SUPPORT= , STATE= , TYPE= , VTOC=
GETJV	<1>	
HELP		INF=
IMPORT		LIST=
LOAD	<1>	CLASSII= , IDA= , SYMTEST=
LOGOFF	<1> ,<2>	
NCANCEL		all operands
NCOPY		all operands
OPTION		MSG= , TESTPRIV=
PASSWORD		REL=
PRINT	<1> ,<2>	BINARY= , CHARS-POOL= ,CONTROL= , COPIES= , DEFER= , DELETE-FILE= , DEVICE= , DEVIN= , FROM= , LOOP= , ROTATION= , SPACE= , TAPE= , TO= , TRANSLATION-TABLE= , VOLUME=
PRIORITY	<2>	

continued ➡

ISP command	No parameter as positional operand ...	No parameter as a value of the keyword operand ...
PUNCH	<1> ,<2>	ACCESS= , BY-PASS= , DEVICE= , DEVIN= , FDTYPE= , SKEL= , TAPE= , VOLUME= , BLKTYPE= , WRITEPR=
RDTFT	<1>	LINK=
RELEASE	<2>	
RESTART	<2> ,<3>	CHECK=
SECURE		FILE= , VOLUME=
SETJV	<1> ,<2>	
SETSW		INVERT= , OFF= , ON=
SET-SS- OPTIONS		SS-NAME=
SKIP	<1>	
SPARAM		COMPRESS=
SQUC		REL= , SUSP= , TYPE=
STAJV	<1> ,<2>	
STATUS	<1> ,<2>	DISP= , IDENT= , INTYPE= , TIMEREQ= , TYPE=
SYSFILE		SYSDTA= , SYSIPT= , SYSLST= , ..., TASKLIB=
SYSTATUS	<1>	
TCHNG		OFLOW= , READ= , TCHAR=
VERIFY		REPAIR= , SUPPORT=
<b>– for system administration</b>		
CATM		BUFCLS= , STATE= , WAIT=
EXCAT	<2>	
IMCAT		ACNTNUM= , BUFCLS= , FORM=
MSGCONTROL		FILE=
RFD		USE=
SEVER	<2>	
SPMGT		SECONDARY=
STAM	<1>	HOST= , REF=

Table 7: Restrictions affecting procedure parameters as operands



## 4 Examples

This chapter contains examples of the conversion of non-S procedures into S procedures and of ISP commands into SDF commands.

In each of the examples, the input and output procedures as well as the logs produced by SDF-CONV are reproduced and annotated.

Example 1: The procedure DO.REORG reorganizes storage space occupied by ISAM files. The conversion into SDF commands as well as into S procedure format is not subject to any restrictions, SDF-CONV terminates without errors.

Input procedure: DO.REORG  
Logging file: LST.REORG  
Output procedure: SDF.SDFP.DO.REORG

Example 2: The procedure DO.LIB.CONTENTES causes the most important element data to be output for all elements in a library. The conversion of the FILE, PRINT and DO commands for a BS2000 V10 environment is subject to certain restrictions. SDF-CONV issues a NOTE for each. The procedure format (non-S) is retained unchanged.

Input procedure: DO.LIB.CONTENTES  
Logging file: LST.LIB.CONTENTES  
Output procedure: SDF.DO.LIB.CONTENTES

Example 3: The procedure PROC.ASSXT is a small auxiliary procedure that is to speed up processing (system file assignment, assembler call) in the event of frequent access to the ASSEMBH assembler. The procedure was written as a non-S procedure using the SDF command language. No restrictions apply to its conversion into an S procedure. SDF-CONV terminates without errors.

Input procedure PROC.ASSXT  
Logging file: LST.ASSXT  
Output procedure: SDFP.PROC.ASSXT

**Example 1:**

The procedure DO.REORG reorganizes the storage space of an ISAM file. The conversion into SDF commands as well as into S procedure format is not subject to any restrictions, SDF-CONV terminates without errors.

**/SHOW-FILE DO.REORG**

```

/PROC N,(&FNAME),SUBDTA=&
/DCLJV #NEWIMAG
/SETJV #NEWIMAG,X'0C'
/SETSW ON=(4,5), OFF=(15)
/SYSFILE SYSDTA=(SYSCMD)
/WR-TEXT '
/WR-TEXT '*****'
/WR-TEXT '***          Reorganization of ISAM files          ***'
/WR-TEXT '*****'
/WR-TEXT '
/WR-TEXT 'Name of file to be reorganized:'
/REMARK &FNAME
/EXEC $EDT
@GET '&FNAME' N
@CON
@IF N:@IF N D:@SY'COPY &FNAME,&FNAME..BEFORE,SAME'
@CON
@IF N : @IF N D : @SAVE O
@CON
@IF E : @SETSW ON=15
@IF D : @SETSW ON=15
@HALT
/SKIP .ERR, ON=(15)
/TCHNG OFLOW=NO
/WR-TEXT '&(#NEWIMAG)'
/TCHNG OFLOW=ACK
/WR-TEXT ''
/WR-TEXT 'File &FNAME has been reorganized.'
/WR-TEXT ''
/WR-TEXT 'Space required before:'
/FS &FNAME..BEFORE
/WR-TEXT ''
/ERASE &FNAME..BEFORE
/WR-TEXT 'Space required now:'
/FS &FNAME
/SKIP .EOP, OFF=(15)
/.ERR REMARK
/WR-TEXT ''
/WR-TEXT '***** Error during file processing *****'
/WR-TEXT '*** REORGANIZATION HAS NOT BEEN PERFORMED ! ***'
/WR-TEXT ''

```

```

/EOP STEP
/SETSW OFF=(4,5,15)
/SYSFILE SYSDTA=(PRIMARY)
/ENDP

```

**/ASSIGN-SYSLST LST.REORG ----- (1)**

**/START-SDF-CONV ----- (2)**

```

% BLS0517 MODULE 'CVRMAIN' LOADED
% CVR0060 SDF-CONV V03.0A40 READY
% //convert from-file=do.reorg,to-file=sdf.sdfp.do.reorg,doc=*min,-
% //procedure-format=s-proc(prog-inp=*data,block-inp=*allowed)

```

% CVR1000 CONVERSION TERMINATED WITH:

```

% CVR1001          FAILURES      : 0
% CVR1002          ERRORS        : 0
% CVR1003          WARNINGS     : 0
% CVR1004          NOTES         : 0
% //end

```

**/ASSIGN-SYSLST \*PRIMARY ----- (3)**

**/SHOW-FILE LST.REORG ----- (4)**

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 1 ----- (4-1)

% CVR1010 OPTIONS IN EFFECT:

```

% CVR1011          FROM-FILE      = DO.REORG
% CVR1012          EXPECT-CONTINUATION = *NEW-MODE
% CVR1013          PARAMETER-LINES = *CONVERT
% CVR1014          TO-FILE        = SDF.SDFP.DO.REORG
% CVR1015          PRODUCE-CONTINUATION= *NEW-MODE
% CVR1016          REPLACE-OLD-FILE = *YES
% CVR1017          SYSTEM-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.U-CMD
% CVR1018          GROUP-SYNTAX-FILE = $.SYSSDF.SDF-CONV.030.USER.S-CMD
% CVR1019          UNCHANGED-CMD   = *NONE
% CVR1020          DOCUMENTATION   = *MINIMUM
% CVR1021          TARGET-VERSION   = *V11
% CVR1022          OUTPUT-FORM      = *STD
% CVR1023          PROCEDURE-FORMAT = *S-PROCEDURE(CMD-FORMAT=*SDF,-
PROGRAM-INPUT=*DATA(EXCEPT-AFTER-CMD=*NONE,-
WARNING=*NO),BLOCK-INPUT=*ALLOWED)

```

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 2 ----- (4-2)

```

/PROC N,(&FNAME),SUBDTA=&
/DCLJV #NEWIMAG
/SETJV #NEWIMAG,X'0C'
/SETSW ON=(4,5), OFF=(15)
/SYSFILE SYSDTA=(SYSCMD)
/WR-TEXT '

```

```

/WR-TEXT '*****'
/WR-TEXT '***          Reorganization of ISAM files          ***'
/WR-TEXT '*****'
/WR-TEXT '
/WR-TEXT 'Name of the file to be reorganized: '
/REMARK &FNAME
/EXEC $EDT
@GET '&FNAME' N
@CON
@IF N:@IF N D:@SY'COPY &FNAME,&FNAME..BEFORE.SAME'
@CON
@IF N : @IF N D : @SAVE O
@CON
@IF E : @SETSW ON=15
@IF D : @SETSW ON=15
@HALT
/SKIP .ERR, ON=(15)
/TCHNG OFLOW=NO
/WR-TEXT '&(#NEWIMAG)'
/TCHNG OFLOW=ACK
/WR-TEXT ''
/WR-TEXT 'File &FNAME has been reorganized.'
/WR-TEXT ''
/WR-TEXT 'Space required before:'
/FS &FNAME..BEFORE
/WR-TEXT ''
/ERASE &FNAME..BEFORE
/WR-TEXT 'Space required now:'
/FS &FNAME
/SKIP .EOP, OFF=(15)
/.ERR REMARK
/WR-TEXT ''
/WR-TEXT '***** Error during file processing *****'
/WR-TEXT '*** REORGANIZATION HAS NOT BEEN PERFORMED ! ***'
/WR-TEXT ''
/.EOP STEP
/SETSW OFF=(4,5,15)
/SYSFILE SYSDDTA=(PRIMARY)
/END
% CVR1000 CONVERSION TERMINATED WITH: ----- (4-3)
% CVR1001          FAILURES      : 0
% CVR1002          ERRORS        : 0
% CVR1003          WARNINGS      : 0
% CVR1004          NOTES         : 0

/SHOW-FILE SDF.SDFP.DO.REORG ----- (5)

/SET-PROCEDURE-OPTIONS LOGGING-ALLOWED=NO,INTERRUPT-ALLOWED=YES,-

```



```

/      DATA-ESCAPE-CHAR=STD,SYSTEM-FILE-CONTEXT=OWN,DATA-ERROR-HAND-
/      LING=NO;BEGIN-PARAMETER-DECLARATION;DECLARE-PARAMETER NAME=-
/      FNAME,TYPE=ANY,INIT=*PROMPT;END-PARAMETER-DECLARATION ----- (5-1)
/SET-JV-LINK JV-NAME=#NEWIMAG
/MODIFY-JV JV-IDENTIFICATION=JV-NAME(JV-NAME=#NEWIMAG),VALUE=X'0C'
/MODIFY-JOB-SWITCHES ON=(4,5),OFF=15
/ASSIGN-SYSDTA TO=*SYSCMD
/WRITE-TEXT '' ----- (5-2)
/WRITE-TEXT '*****'
/WRITE-TEXT '***      Reorganization of ISAM files      ***'
/WRITE-TEXT '*****'
/WRITE-TEXT ''
/WRITE-TEXT 'Name of the file to be reorganized: '
/REMARK '&(FNAME)'
/START-PROGRAM FROM-FILE=$EDT
@GET '&(FNAME)' N
@CON
@IF N:@IF N D:@SY'COPY &(FNAME),&(FNAME).BEFORE,SAME'
@CON
@IF N : @IF N D : @SAVE O
@CON
@IF E : @SETSW ON=15
@IF D : @SETSW ON=15
@HALT
/SKIP-COMMANDS TO-LABEL=ERR,IF=JOB-SWITCHES(ON=15)
/MODIFY-TERMINAL-OPTIONSOVERFLOW-CONTROL=NO-CONTROL
/WRITE-TEXT '&(JV('#NEWIMAG'))'
/MODIFY-TERMINAL-OPTIONSOVERFLOW-CONTROL=USER-ACKNOWLEDGE
/WRITE-TEXT ''
/WRITE-TEXT 'File &(FNAME) has been reorganized.'
/WRITE-TEXT ''
/WRITE-TEXT 'Space required before:'
/SHOW-FILE-ATTRIBUTES FILE-NAME=&(FNAME).BEFORE,SELECT=BY-ATTRIBUTES
/WRITE-TEXT ''
/DELETE-FILE FILE-NAME=&(FNAME).BEFORE,SELECT=BY-ATTRIBUTES
/WRITE-TEXT 'Space required now:'
/SHOW-FILE-ATTRIBUTES FILE-NAME=&(FNAME),SELECT=BY-ATTRIBUTES
/SKIP-COMMANDS TO-LABEL=EOP,IF=JOB-SWITCHES(OFF=15)
/ERR REMARK
/WRITE-TEXT ''
/WRITE-TEXT '***** Error during file processing *****'
/WRITE-TEXT '** REORGANIZATION HAS NOT BEN PERFORMED ! **'
/WRITE-TEXT ''
/.EOP SET-JOB-STEP
/MODIFY-JOB-SWITCHES OFF=(4,5,15)
/ASSIGN-SYSDTA TO=*PRIMARY
/IF-BLOCK-ERROR;EXIT-PROCEDURE ERROR=YES;END-IF;EXIT-PROCEDURE

```

- (1) The system file SYSLST is assigned the file LST.REORG (as a logging file).
- (2) Start of the SDF-CONV utility routine. The CONVERT statement assigns the input and output procedures as well as defining the documentation level. The input procedure is to be converted into an S procedure. Block input of generated SDF commands is to be allowed. All other operands of the CONVERT statement receive the default value. Once processing has been completed, SDF-CONV reports that the conversion was performed without warnings or errors. This listing is output to SYSOUT. SDF-CONV is terminated with the END statement.
- (3) Resetting the SYSLST system file to its default assignment causes the logging file LST.REORG to be closed.
- (4) Contents of the logging file LST.REORG.
  - (4-1) The operand values set for the SDF-CONV run are listed.
  - (4-2) All command records in the input procedure are listed. If an error occurs during the conversion, the appropriate message is written beneath the command record concerned. This is not the case in the example, i.e. SDF-CONV was able to assign an SDF command to each ISP command as well as converting the input procedure into an S procedure without errors.
  - (4-3) The final part of the log is identical to the listing output to SYSOUT.
- (5) Contents of the output procedure SDFP.DO.REORG. The conversion result is an S procedure.
  - (5-1) The SDF commands for the S procedure generated from the command PROC N,... are represented in block format, i.e. the commands are written continuously, separated by semicolon.
  - (5-2) The WRITE-TEXT command is an SDF command and therefore need not be converted. However, SDF-CONV expands the input command to its full input format.

**Example 2:**

The procedure DO.LIB.CONTENTES lists all the elements contained in a library. The most important data for each element is written to a list. The conversion of the FILE, PRINT and DO commands in this procedure is subject to restrictions. SDF-CONV reports a NOTE for each of these commands. The FILE command is never converted if the operand GENERATED-PROCEDURE=FROM-V10 is specified; the PRINT command is not converted because its first operand is a procedure parameter; conversion of the DO command cannot be guaranteed due to the variety of context-dependent conversion possibilities. SDF-CONV enters conversion proposals for the FILE and DO commands in the output procedure.

**/SHOW-FILE DO.LIB.CONTENTES**

```

/PROC N,(&LIBNAME=,&LISTOUT=,&PRINT=,&INPUT),SUBDTA=&
/DCLJV #NEWIMAG
/SETJV #NEWIMAG,X'0C'
/WR-T '&(#NEWIMAG)'
/WR-T ''
/WR-T ''
/WR-T '*****'
/WR-T '****          Listing of elements stored in a library          ****'
/WR-T '*****'
/WR-T ''
/WR-T 'Name of the library                : LIBNAME'
/REMARK input &LIBNAME
/SYSFILE SYSOUT=*DUMMY
/FS &LIBNAME
/SYSFILE SYSOUT=()
/SKIP .LINK
/STEP "Error with FSTATUS"
/SYSFILE SYSOUT=()
/WR-T ''
/WR-T ''
/WR-T 'Input error !'
/WR-T 'Library &LIBNAME does not exist/cannot be opened'
/SKIP .E
/.LINK REMARK Edit the library
/WR-T 'Name of the output list            : LISTOUT'
/WR-T 'If the output list already exists    '
/WR-T 'the data is added at the end of the file.      '
/WR-T 'Is the list to be printed (Y/N) ?           : PRINT '
/REMARK Input &LISTOUT and &PRINT
/SETSW ON=(1)
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSLST=(&LISTOUT,EXTEND)
/SYSFILE SYSOUT=*DUMMY
/FILE &LIBNAME,LINK=LIB001

```

```

/EXEC $LMS
PAR INFO=SUMMARY
PRT (LST)
LST* (1)
END
/STEP
/SYSFILE SYSOUT=()
/SYSFILE SYSLST=()
/SYSFILE SYSDTA=()
/SETSW OFF=(1)
/REL LIB001
/WR-T ''
/WR-T ''
/WR-T '**** Editing completed.'
/WR-T '**** The result is contained in the list &LISTOUT.'
/SKIP "Branch dependent on the PRINT variables" .&PRINT
/WR-T ''
/.J REMARK Print the list
/PRINT &LISTOUT,PNAME=INF,ENDNO=2048
/WR-T 'List &LISTOUT being printed.'
/.N REMARK
/WR-T 'Edit another library or terminate the procedure ?'
/SKIP "Input: A (edit) or E (terminate)" .&INPUT
/.A REMARK Call the procedure again
/DO DO.LIB.CONTENTS
/STEP
/.E REMARK End
/WR-T ''
/WR-T '**** Procedure terminated'
/ENDP

```

**/ASSIGN-SYSLST LST.LIB.CONTENTS ----- (1)**

**/START-SDF-CONV ----- (2)**

```

% BLS0517 MODULE 'CVRMAIN' LOADED
% CVR0060 SDF-CONV V03.0A40 READY
% //convert from-file=do.lib.contents,to-file=sdf.do.lib.contents,doc=*min,-
% //target-version=*v10

```

CRV1000 CONVERSION TERMINATED WITH:

```

CRV1001      FAILURES: 0
CRV1002      ERROR   : 0
CRV1003      WARNINGS: 0
CRV1004      NOTES   : 3
% //end

```

**/ASSIGN-SYSLST \*PRIMARY ----- (3)**

**/SHOW-FILE LST.LIB.CONTENTS ----- (4)**

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 1 ----- (4-1)

% CVR1010 OPTIONS IN EFFECT:

% CVR1011	FROM-FILE	= DO.LIB.CONTENTS
% CVR1012	EXPECT-CONTINUATION	= *NEW-MODE
% CVR1013	PARAMETER-LINES	= *CONVERT
% CVR1014	TO-FILE	= SDF.DO.LIB.CONTENTS
% CVR1015	PRODUCE-CONTINUATION	= *NEW-MODE
% CVR1016	REPLACE-OLD-FILE	= *YES
% CVR1017	SYSTEM-SYNTAX-FILE	= \$.SYSSDF.SDF-CONV.030.USER.U-CMD
% CVR1018	GROUP-SYNTAX-FILE	= \$.SYSSDF.SDF-CONV.030.USER.S-CMD
% CVR1019	UNCHANGED-CMD	= *NONE
% CVR1020	DOCUMENTATION	= *MINIMUM
% CVR1021	TARGET-VERSION	= *V10
% CVR1022	OUTPUT-FORM	= *STD
% CVR1023	PROCEDURE-FORMAT	= *SAME(BLOCK-INPUT=*NOT-ALLOWED)

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 2 ----- (4-2)

/PROC N,(&LIBNAME=,&LISTOUT=,&PRINT=,&INPUT),SUBDTA=&

/DCLJV #NEWIMAG

/SETJV #NEWIMAG,X'0C'

/WR-T '&(#NEWIMAG)'

/WR-T ''

/WR-T ''

/WR-T '\*\*\*\*\*'

/WR-T '\*\*\*\* Listing of elements stored in a library \*\*\*\*'

/WR-T '\*\*\*\*\*'

/WR-T ''

/WR-T 'Name of the library : LIBNAME'

/REMARK Input &LIBNAME

/SYSFILE SYSOUT=\*DUMMY

/FS &LIBNAME

/SYSFILE SYSOUT=()

/SKIP .LINK

/STEP "Error with FSTATUS"

/SYSFILE SYSOUT=()

/WR-T ''

/WR-T ''

/WR-T 'Input error !'

/WR-T 'Library &LIBNAME does not exist/cannot be opened'

/SKIP .E

/.LINK REMARK Edit the library

/WR-T 'Name of the output list : LISTOUT'

/WR-T 'If the output list already exists, the '

/WR-T 'data is added to the end of the file. '

/WR-T 'Is the list to be printed (Y/N) ? : PRINT '

```

/REMARK Input &LISTOUT and &PRINT
/SETSW ON=(1)
/SYSFILE SYSDTA=(SYSCMD)
/SYSFILE SYSLST=(&LISTOUT,EXTEND)
/SYSFILE SYSOUT=*DUMMY
/FILE &LIBNAME,LINK=LIB001
% CVRF062 COMMAND CAN ONLY BE CONVERTED FOR BS2000 VERSIONS GREATER-
THAN OR EQUAL TO V11 ----- (4-3)
/EXEC $LMS
PAR INFO=SUMMARY
PRT (LST)
LST* (1)
END
/STEP
/SYSFILE SYSOUT=()
/SYSFILE SYSLST=()
/SYSFILE SYSDTA=()
/SETSW OFF=(1)
/REL LIB001
/WR-T ''
/WR-T ''
/WR-T '**** Editing completed.'
/WR-T '**** The result is contained in the list &LISTOUT.'
/SKIP "Branch dependent on the PRINT variable" .&PRINT
/WR-T ''
/J REMARK print the list
/PRINT &LISTOUT,PNAME=INF,ENDNO=2048
% CVRF040 INPUT COMMAND NOT CONVERTED ----- (4-4)
/WR-T 'List &LISTOUT is being printed.'
/.N REMARK
/WR-T 'Edit another library or terminate the procedure ?'
/SKIP "Input: A (edit) or E (terminate)" .&INPUT
/.A REMARK Call the procedure again
/DO DO.LIB.CONTENTS
% CVRF061 COMMAND CAN NEVER BE CONVERTED TO A SECURED SDF FORM;-
ORIGINAL INPUT REMAINS VALID ----- (4-5)
/STEP
/E REMARK End
/WR-T ''
/WR-T '**** Procedure terminated'
/ENDP

% CVR1000 CONVERSION TERMINATED WITH: ----- (4-6)
% CVR1001          FAILURES      : 0
% CVR1002          ERRORS        : 0
% CVR1003          WARNINGS      : 0
% CVR1004          NOTES         : 3

```

**/SHOW-FILE SDF.DO.LIB.CONTENTS ----- (5)**

```

/BEGIN-PROCEDURE PARAMETERS=YES(PROCEDURE-PARAMETERS=(-
/      &LIBNAME=-,
/      &LISTOUT=-,
/      &PRINT=-,
/      &INPUT),ESCAPE-CHARACTER='&'),LOGGING=N
/SET-JV-LINK JV-NAME=#NEWIMAG
/MODIFY-JV JV-IDENTIFICATION=JV-NAME(JV-NAME=#NEWIMAG),VALUE=X'0C'
/WRITE-TEXT '&(#NEWIMAG)' ----- (5-1)
/WRITE-TEXT ' '

```

```

/WRITE-TEXT '*****'
/WRITE-TEXT '****          Listing of elements stored in a library          ****'
/WRITE-TEXT '*****'
/WRITE-TEXT ' '
/WRITE-TEXT 'Name of the library          : LIBNAME'
/REMARK Input &LIBNAME
/ASSIGN-SYSOUT TO-FILE=*DUMMY
/SHOW-FILE-ATTRIBUTES FILE-NAME=&LIBNAME,SELECT=BY-ATTRIBUTES
/ASSIGN-SYSOUT TO-FILE=*PRIMARY
/SKIP-COMMANDS TO-LABEL=LINK
/*****" REMARK '/STEP "Error with FSTATUS" ----- (5-2)

```

```

/SET-JOB-STEP
/ASSIGN-SYSOUT TO-FILE=*PRIMARY
/WRITE-TEXT ' '
/WRITE-TEXT 'Input error !'
/WRITE-TEXT 'Library &LIBNAME does not exist/cannot be opened'
/SKIP-COMMANDS TO-LABEL=E
/.LINK REMARK Edit the library
/WRITE-TEXT 'Name of the output list          : LISTOUT '
/WRITE-TEXT 'If the output list already exists, the          '
/WRITE-TEXT 'data is added to the end of the file.          '
/WRITE-TEXT 'Is the list to be printed out (Y/N) ?          : PRINT'
/REMARK Input &LISTOUT and &PRINT
/MODIFY-JOB-SWITCHES ON=1
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/ASSIGN-SYSLST TO-FILE=&LISTOUT,OPEN-MODE=EXTEND
/ASSIGN-SYSOUT TO-FILE=*DUMMY
/*****" REMARK '/CREATE-FILE FILE-NAME=&LIBNAME,SUPPRESS-ERRORS=-
/      *FILE-EXISTING'
/*****" REMARK 'MODIFY-FILE-ATTRIBUTES FILE-NAME=&LIBNAME,'
/*****" REMARK 'SET-FILE-LINK LINK-NAME=LIB001,FILE-NAME=&LIBNAME'
/FILE &LIBNAME,LINK=LIB001 ----- (5-3)

```

```

/START-PROGRAM FROM-FILE=$LMS
PAR INFO=SUMMARY
PRT (LST)
LST* (1)
END

```

```

/SET-JOB-STEP
/ASSIGN-SYSOUT TO-FILE=*PRIMARY
/ASSIGN-SYSLST TO-FILE=*PRIMARY
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/MODIFY-JOB-SWITCHES OFF=1
/REMOVE-FILE-LINK LINK-NAME=LIB001
/WRITE-TEXT ''
/WRITE-TEXT '**** Editing completed.'
/WRITE-TEXT '**** The result is contained in the list &LISTOUT.'
/*****" REMARK '/SKIP "Branch dependent on the PRINT variable" -
/      .&PRINT' ----- (5-4)
/SKIP-COMMANDS TO-LABEL=&PRINT
/WRITE-TEXT ''
/.J REMARK Print the list
/PRINT &LISTOUT,PNAME=INF,ENDNO=2048 ----- (5-5)
/WRITE-TEXT 'List &LISTOUT is being printed.'
/.N REMARK
/WRITE-TEXT 'Edit another library or terminate the procedure ?'
/*****" REMARK '/SKIP "Input: A (edit) or E (terminate)" -
/      .&INPUT' ----- (5-6)
/SKIP-COMMANDS TO-LABEL=&INPUT
/.A REMARK Call the procedure again
/*****" REMARK '/CALL-PROCEDURE NAME=DO.LIB.CONTENTES' ----- (5-7)
/DO DO.LIB.CONTENTES
/SET-JOB-STEP
/.E REMARK End
/WRITE-TEXT ''
/WRITE-TEXT '**** Procedure terminated'
/EXIT-PROCEDURE

```

- (1) The system file SYSLST is assigned the file LST.LIB.CONTENTES (as the logging file).
- (2) The SDF-CONV utility routine is started. The CONVERT statement assigns the input and output procedures, as well as defining the documentation level and the BS2000 environment under which the output procedure is to run. All other operands of the CONVERT statement are given the default value. Once processing has been completed, SDF-CONV reports that conversion was terminated with three notes (NOTE). This overview is output to SYSOUT. SDF-CONV is terminated with the END statement.
- (3) Resetting the system file SYSLST to its default assignment causes the logging file LST.LIB.CONTENTES to be closed.
- (4) Contents of the logging file LST.LIB.CONTENTES.



- (4-1) The operand values set for the SDF-CONV run are listed.
- (4-2) All command records in the input procedure are output.
- (4-3) SDF-CONV issues a NOTE pointing out that the FILE command could not be converted as this is not possible with BS2000 V10. For further details see "Restrictions", page 32.
- (4-4) SDF-CONV issues a NOTE pointing out that the PRINT command could not be converted as its first positional operand is a procedure parameter. For further details see "Restrictions", page 32.
- (4-5) SDF-CONV issues a NOTE pointing out that the DO command could not be converted for reasons of context-dependency. For further details see "Restrictions", page 32.
- (4-6) The last part of the log is identical with the output to SYSOUT.
- (5) Contents of the output procedure SDF.DO.LIB.CONTENTES.
- (5-1) The WRITE-TEXT command is an SDF command and therefore need not be converted. However, SDF-CONV expands the input command to its full input format.
- (5-2) The command record with the inline comment is written to a comment line preceding the command line with the conversion result. This ensures that the information in the inline comment is not lost during conversion.  
(In this case, DOC=\*MIN/\*MAX must be set.)
- (5-3) The command record containing the unconverted FILE command is taken over from the input procedure. At the same time, however, SDF-CONV suggests what the conversion of this FILE command could look like. This proposal (FILE is implemented by way of three SDF commands, with the result that more than one line is needed) is written as a comment line preceding the unconverted command record. (In this case, DOC=\*MIN/\*MAX must be set.)
- (5-4) see (5-2).
- (5-5) The command record with the unconverted PRINT command is taken over from the input procedure.
- (5-6) see (5-2).
- (5-7) The command record with the unconverted DO command is taken over from the input procedure. At the same time, however, SDF-CONV suggests what the conversion of this DO command could look like. This proposal is written as a comment line preceding the unconverted command record.  
(In this case, DOC=\*MIN/\*MAX must be set.)

### Example 3

The procedure PROC.ASSXT is a small auxiliary procedure that is to speed up processing (system file assignment, assembler call) in the event of frequent access to the ASSEMBH assembler. The procedure was written as a non-S procedure using the SDF command language. No restrictions apply to its conversion into an S procedure.

#### /SHOW-FILE PROC.ASSXT

```

/BEGIN-PROC A,P-P=(&FILE),ESC-CHAR='&'
/REMARK **** Call ASSEMBH ****
/DEL-FI LST.ASSEMBH
/SET-JOB-STEP
/ASS-SYSDTA *SYSCMD
/DEL-SYS-FI "from previous ASSEMBH calls" *OMF
/START-PROG $ASSEMBH
//COMPILE SOURCE=&FILE,LISTING=PAR(OUTPUT=LST.ASSEMBH),-
//TEST-SUPPORT=YES
//END
/SET-JOB-STEP
/ASS-SYSDTA *P
/END-PROC

```

#### /ASSIGN-SYSLST LST.ASSXT ----- (1)

#### /START-SDF-CONV ----- (2)

```

% BLS0517 MODULE 'CVRMAIN' LOADED
% CVR0060 SDF-CONV V03.0A40 READY
% //convert from-file=proc.assxt,to-file=sdfp.proc.assxt,doc=*min,-
% //procedure-format=s-proc(prog-inp=*data)

% CVR1000 CONVERSION TERMINATED WITH:
% CVR1001          FAILURES    : 0
% CVR1002          ERRORS      : 0
% CVR1003          WARNINGS    : 0
% CVR1004          NOTES       : 0
% //end

```

#### /ASSIGN-SYSLST \*PRIMARY ----- (3)

**/SHOW-FILE LST.ASSXT ----- (4)**

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 1 ----- (4-1)

% CVR1010 OPTIONS IN EFFECT:

```
% CVR1011      FROM-FILE              = PROC.ASSXT
% CVR1012      EXPECT-CONTINUATION    = *NEW-MODE
% CVR1013      PARAMETER-LINES       = *CONVERT
% CVR1014      TO-FILE                = SDFP.PROC.ASSXT
% CVR1015      PRODUCE-CONTINUATION= *NEW-MODE
% CVR1016      REPLACE-OLD-FILE      = *YES
% CVR1017      SYSTEM-SYNTAX-FILE    = $.SYSSDF.SDF-CONV.030.USER.U-CMD
% CVR1018      GROUP-SYNTAX-FILE     = $.SYSSDF.SDF-CONV.030.USER.S-CMD
% CVR1019      UNCHANGED-CMD        = *NONE
% CVR1020      DOCUMENTATION         = *MINIMUM
% CVR1021      TARGET-VERSION        = *V11
% CVR1022      OUTPUT-FORM           = *STD
% CVR1023      PROCEDURE-FORMAT      = *S-PROCEDURE(CMD-FORMAT=*SDF,-
PROGRAM-INPUT=*DATA(EXCEPT-AFTER-CMD=*NONE,-
WARNING=*NO),BLOCK-INPUT=*NOT-ALLOWED)
```

\*\*\*\*\* BS2000 COMMAND CONVERTER : SDF-CONV V3 \*\*\*\*\* page 2 ----- (4-2)

```
/BEGIN-PROC A,P-P=(&FILE),ESC-CHAR='&'
/REMARK **** Call ASSEMBH ****
/DEL-FI LST.ASEMBH
/SET-JOB-STEP
/ASS-SYSDTA *SYSCMD
/DEL-SYS-FI "from previous ASSEMBH calls" *OMF
/START-PROG $ASSEMBH
//COMPILE SOURCE=&FILE,LISTING=PAR(OUTPUT=LST.ASEMBH),-
//TEST-SUPPORT=YES
//END
/SET-JOB-STEP
/ASS-SYSDTA *P
/END-PROC
```

% CVR1000 CONVERSION TERMINATED WITH:----- (4-3)

```
% CVR1001      FAILURES      : 0
% CVR1002      ERRORS        : 0
% CVR1003      WARNINGS     : 0
% CVR1004      NOTES         : 0
```

**/SHOW-FILE SDFP.PROC.ASSXT ----- (5)**

```
/SET-PROCEDURE-OPTIONS LOGGING-ALLOWED=YES,INTERRUPT-ALLOWED=YES,-
/   DATA-ESCAPE-CHAR=STD,SYSTEM-FILE-CONTEXT=OWN,DATA-ERROR-HAND-
/   LING=NO
/BEGIN-PARAMETER-DECLARATION
```

```

/DECLARE-PARAMETER NAME=FILE,TYPE=ANY,INIT=*PROMPT
/END-PARAMETER-DECLARATION
/REMARK '**** Call ASSEMBH ****'
/DELETE-FILE LST.ASEMBH
/SET-JOB-STEP
/ASSIGN-SYSDTA *SYSCMD
/*****" REMARK '/DEL-SYS-FI "from previous ASSEMBH calls" *OMF' ----- (5-1)
/DELETE-SYSTEM-FILE "from previous ASSEMBH calls" *OMF
/START-PROGRAM $ASSEMBH
//COMPILE SOURCE=&(FILE),LISTING=PAR(OUTPUT=LST.ASEMBH),TEST-SUPPORT=-
//      YES
//END
/SET-JOB-STEP
/ASSIGN-SYSDTA *P
/IF-BLOCK-ERROR;END-IF
/END-PROCEDURE

```

- (1) The system file SYSLST is assigned the file LST.ASSXT (as the logging file).
- (2) The SDF-CONV utility routine is started. The CONVERT statement assigns the input and output procedures, as well as defining the documentation level. The procedure is to be converted into an S procedure. All other operands of the CONVERT statement retain their default value. Once processing has been completed, SDF-CONV reports that conversion was terminated without warnings or errors. This overview is output to SYSOUT. SDF-CONV is terminated with the END statement.
- (3) Resetting the system file SYSLST to its default assignment causes the logging file LST.ASSXT to be closed.
- (4) Contents of the logging file LST.ASSXT.
  - (4-1) The operand values set for the SDF-CONV run are listed.
  - (4-2) All command records in the input procedure are output.
  - (4-3) The last part of the log is identical with the output to SYSOUT.
- (5) Contents of the output procedure SDFP.PROC.ASSXT.
  - (5-1) The command record with the inline comment is written to a comment line preceding the command line with the conversion result. This ensures that the information in the inline comment is not lost during conversion. (In this case, DOC=\*MIN/\*MAX must be set.)

## 5 Appendix

The appendix contains the following tables and descriptions:

- SDF syntax description
- assignment of ISP commands to SDF commands, subdivided into user commands and system administration commands
- conversion of ISP commands by SDF-CONV, subdivided into user commands and system administration commands
- SDF-CONV messages

## 5.1 SDF syntax description

Figure 1 shows a sample command syntax description as used in manuals. The command format consists of a field containing the command name. This is followed by a list of all the operands, together with the permissible operand values. Structure-initiating operand values and the operands dependent on them are also listed.

<b>MODIFY-SDF-OPTIONS</b>
<b>SYNTAX-FILE = <u>UNCHANGED</u> / *NONE / &lt;full-filename 1..54&gt;</b> <b>,GUIDANCE = <u>UNCHANGED</u> / EXPERT / NO / MAXIMUM / MEDIUM / MINIMUM</b> <b>,LOGGING = <u>UNCHANGED</u> / INPUT-FORM / ACCEPTED-FORM / INVARIANT-FORM</b> <b>,UTILITY-INTERFACE = <u>UNCHANGED</u> / OLD-MODE / NEW-MODE</b> <b>,PROCEDURE-DIALOGUE = <u>UNCHANGED</u> / YES / NO</b> <b>,CONTINUATION = <u>UNCHANGED</u> / OLD-MODE / NEW-MODE</b> <b>,MENU-LOGGING = <u>UNCHANGED</u> / NO / YES</b> <b>,MODE = <u>UNCHANGED</u> / EXECUTION / TEST</b> <b>,TEST-PROGRAM-NAME = *<u>UNCHANGED</u> / *NONE / &lt;structured-name 1..30&gt;</b>

Figure 1: Syntax description of the user command MODIFY-SDF-OPTIONS

This syntax description is valid for SDF Version 3.0A. The syntax of the SDF command/statement language is explained in the following three tables.

### Notational conventions (Table 8):

Certain characters and representations are used in the command/statement formats; their meaning is explained in Table 8.

### Data types (Table 9):

Variable operand values are represented in SDF by data types. Each data type represents a specific range of values. The number of data types is limited to those described in Table 9. The description of the data types is valid for the entire set of commands and statements. Therefore only deviations (if any) from Table 9 are explained in the relevant operand descriptions.

**Suffixes for data types (Table 10):**

Data type suffixes define additional rules for data type input. They can be used to limit or extend the range of values. The following abbreviations are used in this manual for data type suffixes:

cat-id	cat
completion	compl
correction-state	corr
generation	gen
lower-case	low
odd-possible	odd
separators	sep
underscore	under
user-id	user
user-interface	user-int
version	vers
wildcards	wild

The description of the data type *integer* in Table 10 also contains a number of units in italics; these are not part of the syntax, but are intended merely to make the table easier to read. The description of the data type suffixes is valid for the entire set of commands and statements. Therefore only deviations (if any) from Table 10 are explained in the relevant operand descriptions.

## Notational conventions

Representation	Meaning	Examples
UPPERCASE	Uppercase designates keywords. Some keywords begin with *	<b>MODE = <u>UNCHANGED</u></b>
<b>UPPERCASE LETTERS</b> in boldface	Uppercase letters printed in boldface denote guaranteed/ suggested abbreviations of keywords.	<b>NAME = *ALL</b> <b>CHECK-VERSION = *<u>NO</u></b>
=	An equals sign associates an operand name with the corresponding operand values.	<b>SCOPE = <u>TEMPORARY</u></b>
< >	Angle brackets designate variables whose set of possible values is described using data types and suffixes (see Tables 9 and 10).	<b>NAME = *<u>NONE</u> / &lt;name 1..8&gt;</b>
<u>underscore</u>	An underscore designates the default value for an operand.	<b>REPLACE-PRODUCT = <u>YES</u> / NO</b>
/	A slash separates alternative operand values.	<b>MODE = <u>UPDATE</u> / READ</b>
(...)	Parentheses designate operand values which introduce a structure.	<b>TYPE = GROUP(...)</b>
[ ]	Square brackets designate optional operand values which introduce a structure. The structure which follows can be specified without the introductory operand value.	<b>LOGGING=[PARAMETERS](...)</b>
indentation	Indentation indicates dependence on the next higher-level operand.	<b>OBJECT = DOMAIN(...)</b> <b>DOMAIN(...)</b>   <b>NAME = *<u>ALL</u></b>

continued ➡



Representation	Meaning	Examples
<p> </p> <p>,</p> <p>list-poss(n):</p>	<p>A vertical bar designates operands which belong together in a structure. Its extent shows the beginning and end of a structure. Within one structure there may be other structures. The number of vert. bars in front of an operand shows the structure's depth.</p> <p>A comma is shown before additional operands at the same structure level.</p> <p>A list can be formed from the operand values which follow list-poss. If (n) is specified, the list can consist of a maximum of n elements. If the list contains more than one element, the list must be enclosed in parentheses.</p>	<p><b>SUPPORT = TAPE(...)</b></p> <p><b>TAPE(...)</b></p> <p>  <b>VOLUME = *<u>ANY</u>(...)</b></p> <p>  <b>*<u>ANY</u>(...)</b></p> <p>    ...</p> <p><b>,SIZE = MINIMUM</b></p> <p><b>,FORM = UNGUIDED</b></p> <p>list-poss: <b>SAM / ISAM</b></p> <p>list-poss(40): &lt;structured-name 1..30&gt;</p> <p>list-poss(256): <b>*OMF / *SYSLST(...)</b> / &lt;full-filename 1..54&gt;</p>

Table 8: Notational conventions

## Data types

Data type	Character set	Comments
alphanum-name	A...Z 0...9 \$, #, @	
cat-id	A...Z 0...9	Maximum of 4 characters
command-rest	all characters	
composed-name	A...Z 0...9 \$, #, @ hyphen period	Alphanumeric string which can be subdivided into a number of substrings separated by periods and hyphens.
c-string	EBCDIC characters	Must be enclosed in single quotes; the letter C can be prefixed; single quotes within a c-string must be doubled.
date	0...9 structure indicator: hyphen	Input format: yyyy-mm-dd  yyyy:      year; optionally 2 or 4 digits mm:        month dd:        day
device	A...Z 0...9 hyphen	A string which is up to 8 characters long and corresponds to a device available on the system. In guided dialog, SDF displays the permissible operand values. Notes on the possible devices can be found in the operand descriptions.
fixed	+, - 0...9 period	Input format: [sign][digits].[digits]  [sign]:     + or - [digits]:   0...9  Must contain at least one digit, and may contain up to 10 characters (0...9, period) besides sign prefix.

continued ➡

Data type	Character set	Comments
full-filename	A...Z 0...9 \$, #, @ hyphen period	<p>Input format:</p> <pre> :cat:\$user. {   file   file(num)   group   group { (*abs)           (+rel)           (-rel)         } } </pre> <p>:cat: Optional specification of catalog ID; character set limited to A...Z and 0...9; max. 4 characters; enclosed in colons; default is the catalog ID associated with the user ID in the user catalog entry.</p> <p>\$user. Optional specification of user ID; character set limited to A...Z, 0...9, \$, #, @; max. 8 characters; \$ and period must be entered; default value is the user's own ID.</p> <p>\$. (exception) System default ID</p> <p>file File or job variable name; final character may not be a hyphen or period; max. 41 characters; must contain at least A...Z.</p> <p>#file (exception) @file (exception) # or @ as first character identifies - depending on how the system was generated - temporary files and job variables.</p>

continued ➡

Data type	Character set	Comments
full-filename (cont.)		<p>file(num) Tape file name num: version number; character set is A...Z, 0...9, \$, #, @. Parentheses must be entered.</p> <p>group Name of a file generation group (character set: see under "file")</p> <p>group <math>\left\{ \begin{array}{l} (*abs) \\ (+rel) \\ (-rel) \end{array} \right\}</math></p> <p>(*abs) Absolute generation number (1-9999); and parentheses must be entered.</p> <p>(+rel) (-rel) Relative generation number (0-99); sign and parentheses must be entered.</p>
integer	0...9, +, -	+ or - can be initial character only.
name	A...Z 0...9 \$, #, @	May not begin with 0...9
partial-filename	A...Z 0...9 \$, #, @ hyphen period	<p>Input format: :cat:\$user.partname.</p> <p>:cat: see full-filename \$user. see full-filename</p> <p>partname Optional specification of the first part of a name held in common by files and file generation groups. The last character of partname must be a period. At least one of the parts (:cat:, \$user. or partname) must be specified.</p>

continued ➡



## Suffixes for data types

Suffix	Meaning
x..y <i>unit</i>	<p>a) for data type "integer": interval specification</p> <p>x     minimum value permissible for integer. x is a whole number which may be preceded by a sign.</p> <p>y     maximum value permissible for integer. y is a whole number which may be preceded by a sign.</p> <p><i>unit</i> for integer only: additional units. The following are used:</p> <p><i>days</i>     <i>byte</i> <i>hours</i>     <i>2Kbyte</i> <i>minutes</i>   <i>4Kbyte</i> <i>seconds</i>   <i>Mbyte</i></p> <p>b) for the other data types: length specification</p> <p>x     minimum length for the operand value; x is a whole number.</p> <p>y     maximum length for the operand value; y is a whole number.</p> <p>x=y   The operand value must have precisely the length x.</p>
with	Expands the possible values for a data type.
-compl	For specifications of data type "date", SDF transforms 2-digit year values entered in the form yy-mm-dd as follows:
	<p>20yy-mm-dd   if yy &lt; 60 19yy-mm-dd   if yy ≥ ≥ ≥ ≥ 60</p>
-low	Upper and lowercase are differentiated.
-under	Permits underscores '_' for data type "name".

continued ➡

Suffix	Meaning												
with (cont.)													
-wild(n)	Parts of a name may be replaced by the following wildcards. n indicates the maximum entry length when wildcards are used.												
	<table border="1"> <thead> <tr> <th>Wildcard</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>*</td> <td>Replaces any string, including an empty string. A leading * followed by other characters must be doubled unless the string contains at least one further wildcard.</td> </tr> <tr> <td>/</td> <td>Replaces one and only one arbitrary character.</td> </tr> <tr> <td>&lt;s<sub>x</sub>:s<sub>y</sub>&gt;</td> <td>Replaces a string meeting the criteria below: <ul style="list-style-type: none"> <li>– it is at least as long as the shortest string (s<sub>x</sub> or s<sub>y</sub>)</li> <li>– it is at most as long as the longest string (s<sub>x</sub> or s<sub>y</sub>)</li> <li>– it lies in the alphabetical sequence between s<sub>x</sub> and s<sub>y</sub>; numbers are sorted after letters (A...Z 0...9)</li> <li>– s<sub>x</sub> may also be an empty string, which stands at the beginning of the alphabetical sequence.</li> <li>– s<sub>y</sub> may also be an empty string, which is in the highest position in the alphabetic collating sequence (contains only the characters X' FF' )</li> </ul> </td> </tr> <tr> <td>&lt;s<sub>1</sub>,...&gt;</td> <td>Replaces all strings which match one of the character combinations indicated by s. s can also be an empty string. Any of the strings designated by s can also be a range "s<sub>x</sub>:s<sub>y</sub>" (see above).</td> </tr> <tr> <td>-s</td> <td>Replaces all strings which do not match the specified string s. The minus sign may only stand at the beginning of the string s. Within the data types full-filename and partial-filename the negated string -s may occur once only, i.e. -s can replace one of the three name parts cat, user or file</td> </tr> </tbody> </table>	Wildcard	Meaning	*	Replaces any string, including an empty string. A leading * followed by other characters must be doubled unless the string contains at least one further wildcard.	/	Replaces one and only one arbitrary character.	<s <sub>x</sub> :s <sub>y</sub> >	Replaces a string meeting the criteria below: <ul style="list-style-type: none"> <li>– it is at least as long as the shortest string (s<sub>x</sub> or s<sub>y</sub>)</li> <li>– it is at most as long as the longest string (s<sub>x</sub> or s<sub>y</sub>)</li> <li>– it lies in the alphabetical sequence between s<sub>x</sub> and s<sub>y</sub>; numbers are sorted after letters (A...Z 0...9)</li> <li>– s<sub>x</sub> may also be an empty string, which stands at the beginning of the alphabetical sequence.</li> <li>– s<sub>y</sub> may also be an empty string, which is in the highest position in the alphabetic collating sequence (contains only the characters X' FF' )</li> </ul>	<s <sub>1</sub> ,...>	Replaces all strings which match one of the character combinations indicated by s. s can also be an empty string. Any of the strings designated by s can also be a range "s <sub>x</sub> :s <sub>y</sub> " (see above).	-s	Replaces all strings which do not match the specified string s. The minus sign may only stand at the beginning of the string s. Within the data types full-filename and partial-filename the negated string -s may occur once only, i.e. -s can replace one of the three name parts cat, user or file
Wildcard	Meaning												
*	Replaces any string, including an empty string. A leading * followed by other characters must be doubled unless the string contains at least one further wildcard.												
/	Replaces one and only one arbitrary character.												
<s <sub>x</sub> :s <sub>y</sub> >	Replaces a string meeting the criteria below: <ul style="list-style-type: none"> <li>– it is at least as long as the shortest string (s<sub>x</sub> or s<sub>y</sub>)</li> <li>– it is at most as long as the longest string (s<sub>x</sub> or s<sub>y</sub>)</li> <li>– it lies in the alphabetical sequence between s<sub>x</sub> and s<sub>y</sub>; numbers are sorted after letters (A...Z 0...9)</li> <li>– s<sub>x</sub> may also be an empty string, which stands at the beginning of the alphabetical sequence.</li> <li>– s<sub>y</sub> may also be an empty string, which is in the highest position in the alphabetic collating sequence (contains only the characters X' FF' )</li> </ul>												
<s <sub>1</sub> ,...>	Replaces all strings which match one of the character combinations indicated by s. s can also be an empty string. Any of the strings designated by s can also be a range "s <sub>x</sub> :s <sub>y</sub> " (see above).												
-s	Replaces all strings which do not match the specified string s. The minus sign may only stand at the beginning of the string s. Within the data types full-filename and partial-filename the negated string -s may occur once only, i.e. -s can replace one of the three name parts cat, user or file												
	Wildcards are not permitted in generation and version specifications of file names. Wildcards may be used in user IDs by the system administration only.												

continued ➡

Suffix	Meaning
without	Restricts the possible values for a data type.
-cat	The specification of a catalog ID is not permitted.
-corr	Specifications of data type product-version may not contain the correction status. The entry format is: ['] [V] [n] n.na [']
-gen	The specification of a file generation, or a file generation group, is not permitted.
-man	Specifications of data type product-version may not contain the release status or correction status. The entry form is: ['] [V] [n] n.n [']
-odd	The data type x-text permits only an even number of characters.
-sep	For data type "text", the specification of the following delimiters is not permitted: ; = ( ) < > ? (semicolon, equals sign, left and right parentheses, left and right angle brackets, and blanks)
-user	The specification of a user ID is not permitted.
-vers	The specification of the version (see file(num)) is not permitted for tape files.

Table 10: Suffixes for data types



## 5.2 Assignment of ISP to SDF commands

### 5.2.1 User commands

ISP command	SDF command
ABEND	EXIT-JOB
ABORT	CANCEL-PROCEDURE
ADD-ISAM-POOL-LINK	ADD-ISAM-POOL-LINK
APPLICATION	SET-DCAM-APPLICATION-LINK
AUDIT	RESUME-HARDWARE-AUDIT
AUDIT	SHOW-HARDWARE-AUDIT
AUDIT	START-HARDWARE-AUDIT
AUDIT	STOP-HARDWARE-AUDIT
BCNTRL	MODIFY-MSG-OPTIONS
BREAK	HOLD-PROGRAM
CALL	CALL-PROCEDURE
CANCEL	CANCEL-JOB
CANCEL-FILE-TRANSFER (NCANCEL)	CANCEL-FILE-TRANSFER
CATALOG	CREATE-FILE
CATALOG	CREATE-FILE-GROUP
CATALOG	MODIFY-FILE-ATTRIBUTES
CATALOG	MODIFY-FILE-GROUP-ATTRIBUTES
CATJV	CREATE-JV
CATJV	MODIFY-JV-ATTRIBUTES
CHANGE	CHANGE-FILE-LINK
CONNECTION	REMOVE-DCAM-CONNECTION-LINK
CONNECTION	SET-DCAM-CONNECTION-LINK
COPY	COPY-FILE
CREATE-ISAM-POOL	CREATE-ISAM-POOL
DCLJV	SET-JV-LINK
DELETE-ISAM-POOL	DELETE-ISAM-POOL

continued ➡

<b>ISP command</b>	<b>SDF command</b>
DELON	REMOVE-CJC-ACTION
DROP	UNLOCK-FILE-LINK
ENDON	END-CJC-ACTION
ENDP	END-PROCEDURE
ENDP	EXIT-PROCEDURE
ENTER	ENTER-JOB
EOF	EOF
ERAJV	DELETE-JV
ERASE	DELETE-FILE
ERASE	DELETE-FILE-GENERATION
ERASE	DELETE-FILE-GROUP
ERASE	DELETE-SYSTEM-FILE
ERASE	EXPORT-FILE
ESCAPE	HOLD-PROCEDURE
EXECUTE	START-PROGRAM
FILE	CREATE-FILE
FILE	CREATE-FILE-GENERATION
FILE	CREATE-TAPE-SET
FILE	DELETE-TAPE-SET
FILE	DELETE-TAPE-SET
FILE	EXTEND-TAPE-SET
FILE	IMPORT-FILE
FILE	MODIFY-FILE-ATTRIBUTES
FILE	MODIFY-FILE-GENERATION-SUPPORT
FILE	SET-FILE-LINK
INTR	SEND-MESSAGE
LOAD	LOAD-PROGRAM
LOGOFF	EXIT-JOB
LOGOFF	LOGOFF
LOGON	SET-LOGON-PARAMETERS

continued ➡

<b>ISP command</b>	<b>SDF command</b>
MODIFY-JOB	MODIFY-JOB
MODIFY-JV-CONDITIONALLY	MODIFY-JV-CONDITIONALLY
MODIFY-MSG-ATTRIBUTES	MODIFY-MSG-ATTRIBUTES
MRSSTA	SHOW-MRS-CONNECTIONS
MSGCONTROL	MODIFY-MSG-FILE-ASSIGNMENT
NSTATUS	SHOW-FILE-TRANSFER
ON	ADD-CJC-ACTION
OPTION	MODIFY-JOB-OPTIONS
OPTION	MODIFY-TEST-OPTIONS
PARAMETER	PARAMETER
PASSWORD	ADD-PASSWORD
PASSWORD	REMOVE-PASSWORD
PAUSE	SEND-MESSAGE
PRINT	PRINT-FILE
PRINT	WRITE-SPOOL-TAPE
PRIORITY	CHANGE-TASK-PRIORITY
PRIORITY	MODIFY-JOB
PROCEDURE	BEGIN-PROCEDURE
PSWORD	MODIFY-USER-PROTECTION
PUNCH	WRITE-DISKETTE
PUNCH	WRITE-POOLER-TAPE
RDIR	REDIRECT-REMOTE-OUTPUT
RDTFT	SHOW-FILE-LINK
RELEASE	REMOVE-FILE-LINK
RELEASE-SUBSYSTEM-SPACE	RELEASE-SUBSYSTEM-SPACE
REMARK	REMARK
REMOVE-ISAM-POOL-LINK	REMOVE-ISAM-POOL-LINK
REMOVE-JV-LINK	REMOVE-JV-LINK
REMOVE-SPOOL-CHARACTER-SET	REMOVE-SPOOL-CHARACTER-SET
REMOVE-SPOOL-DEVICE	REMOVE-SPOOL-DEVICE

continued ➡

<b>ISP command</b>	<b>SDF command</b>
REMOVE-SPOOL-FORM	REMOVE-SPOOL-FORM
RESTART	RESTART-PROGRAM
RESUME	RESUME-PROGRAM
RFAEND	REMOVE-RFA-CONNECTION
RFASTART	SET-RFA-CONNECTION
RTI	RESUME-PROCEDURE
SDVC	MODIFY-PRINTER-OUTPUT-STATUS
SDVC	START-PRINTER-OUTPUT
SDVC	STOP-PRINTER-OUTPUT
SECURE-RESOURCE-ALLOCATION	SECURE-RESOURCE-ALLOCATION
SHOW-DISK-DEFAULTS	SHOW-DISK-DEFAULTS
SHOW-DISK-STATUS	SHOW-DISK-STATUS
SHOW-FILE	SHOW-FILE
SHOW-FILE-TRANSFER	SHOW-FILE-TRANSFER
SHOW-JOB-CLASS	SHOW-JOB-CLASS
SHOW-JV-LINK	SHOW-JV-LINK
SHOW-MOUNT-PARAMETER	SHOW-MOUNT-PARAMETER
SHOW-MSG-DEFAULTS	SHOW-MSG-FILE-ASSIGNMENT
SHOW-PCS-OPTION	SHOW-PCS-OPTION
SHOW-RESOURCE-ALLOCATION	SHOW-RESOURCE-ALLOCATION
SHOW-SPOOL-CHARACTER-SETS	SHOW-SPOOL-CHARACTER-SETS
SHOW-SPOOL-DEVICES	SHOW-SPOOL-DEVICES
SHOW-SPOOL-FORMS	SHOW-SPOOL-FORMS
SHOW-SPOOL-PARAMETERS	SHOW-SPOOL-PARAMETERS
SHOW-SS-STATUS	SHOW-SUBSYSTEM-STATUS
SHOW-TAPE-STATUS	SHOW-TAPE-STATUS
SHOW-USER-ATTRIBUTES	SHOW-USER-ATTRIBUTES
SKIP	SKIP-COMMANDS
SKIPJV	SKIP-COMMANDS
SKIPUS	SKIP-COMMANDS

continued ➡

<b>ISP command</b>	<b>SDF command</b>
SPARAM	MODIFY-SPOOLOUT-OPTIONS
SQUC	HOLD-SPOOLOUT
SQUC	RESUME-SPOOLOUT
STAJV	SHOW-JV-ATTRIBUTES
STAM	SHOW-MASTER-CATALOG-ENTRY
STATUS	SHOW-JOB-STATUS
STATUS	SHOW-SYSTEM-STATUS
STATUS	SHOW-USER-STATUS
STEP	SET-JOB-STEP
SYSFILE	ASSIGN-SYSDTA
SYSFILE	ASSIGN-SYSIPT
SYSFILE	ASSIGN-SYSLST
SYSFILE	ASSIGN-SYSOPT
SYSFILE	ASSIGN-SYSOUT
SYSFILE	REMOVE-TASKLIB
SYSFILE	SET-TASKLIB
SYSTATUS	SHOW-SYSTEM-FILE-ASSIGNMENTS
TCHNG	MODIFY-TERMINAL-OPTIONS
TRANSFER-FILE	TRANSFER-FILE
TYPE	SEND-MESSAGE
VERIFY	CHECK-FILE-CONSISTENCY
VERIFY	REMOVE-FILE-ALLOCATION-LOCKS
VERIFY	REPAIR-DISK-FILES
WAIT	WAIT-EVENT
WHEN	WAIT-EVENT
WRITE-ACCOUNTING-RECORD	WRITE-ACCOUNTING-RECORD

Table 11: Assignment of ISP to SDF commands

## 5.2.2 System administration commands

ISP command	SDF command
ADD-SPOOL-DEVICE	ADD-SPOOL-DEVICE
ADD-SPOOL-FORM	ADD-SPOOL-FORM
ADD-SUBSYSTEM	ADD-SUBSYSTEM
AUDIT	HOLD-HARDWARE-AUDIT
AUDIT	RESUME-HARDWARE-AUDIT
AUDIT	SHOW-HARDWARE-AUDIT
AUDIT	START-HARDWARE-AUDIT
AUDIT	STOP-HARDWARE-AUDIT
BIAS	MODIFY-SYSTEM-BIAS
CATALOG	CREATE-FILE
CATALOG	CREATE-FILE-GENERATION
CATALOG	CREATE-FILE-GROUP
CATALOG	MODIFY-FILE-ATTRIBUTES
CATALOG	MODIFY-FILE-GENERATION-SUPPORT
CATALOG	MODIFY-FILE-GROUP-ATTRIBUTES
CATEGORY	MODIFY-TASK-CATEGERIES
CATM	ADD-MASTER-CATALOG-ENTRY
CATM	MODIFY-MASTER-CATALOG-ENTRY
CHANGE-ACCOUNTING-FILE	CHANGE-ACCOUNTING-FILE
CHANGE-CONSLOG	CHANGE-CONSLOG-FILE
CHANGE-SERSLOG	CHANGE-SERSLOG-FILE
ERAM	REMOVE-MASTER-CATALOG-ENTRY
ERASE	DELETE-FILE
ERASE	DELETE-FILE-GENERATION
ERASE	DELETE-FILE-GROUP
ERASE	EXPORT-FILE
EXCAT	EXPORT-CATALOG
EXCAT	EXPORT-PUBLIC-VOLUME-SET
EXCAT	FORCE-CATALOG-EXPORT

continued ➡

<b>ISP command</b>	<b>SDF command</b>
FILE	CREATE-FILE
FILE	CREATE-FILE-GENERATION
FILE	CREATE-TAPE-SET, EXTEND-TAPE-SET
FILE	IMPORT-FILE
FILE	MODIFY-FILE-ATTRIBUTES
FILE	MODIFY-FILE-GENERATION-SUPPORT
FILE	SET-FILE-LINK
HOLD-JOB	HOLD-JOB
HOLD-JOB-CLASS	HOLD-JOB-CLASS
HOLD-PCS	HOLD-PCS
HOLD-SS	HOLD-SUBSYSTEM
IMCAT	IMPORT-PUBLIC-VOLUME-SET
IMPORT	CHECK-IMPORT-DISK-FILE
IMPORT	IMPORT-FILE
JOIN	ADD-USER
JOIN	MODIFY-USER
LOADAID	LOADAID
MODIFY-ACCOUNTING-PARAMETERS	MODIFY-ACCOUNTING-PARAMETERS
MODIFY-JOB-CLASS	MODIFY-JOB-CLASS
MODIFY-JOB-STREAM	MODIFY-JOB-STREAM
MODIFY-PCS-OPTION	MODIFY-PCS-OPTIONS
MODIFY-SPOOL-FORM	MODIFY-SPOOL-FORM
MODIFY-SPOOL-PARAMETERS	MODIFY-SPOOL-PARAMETERS
MSGCONTROL	MODIFY-MSG-FILE-ASSIGNMENT
MSGCONTROL	RESET-MSG-FILE-ASSIGNMENT
NCHOLD	HOLD-TASK
NCREL	RESUME-TASK
RDIR	REDIRECT-REMOTE-OUTPUT
RELEASE-JOB	RESUME-JOB
RELEASE-JOB-CLASS	RESUME-JOB-CLASS

continued ➡

<b>ISP command</b>	<b>SDF command</b>
RELEASE-JOB-STREAM	RESUME-JOB-STREAM
REMOVE-SPOOL-CHARACTER-SET	REMOVE-SPOOL-CHARACTER-SET
REMOVE-SPOOL-DEVICE	REMOVE-SPOOL-DEVICE
REMOVE-SPOOL-FORM	REMOVE-SPOOL-FORM
RESUME-PCS	RESUME-PCS
RESUME-SS	RESUME-SUBSYSTEM
RFD	START-DISKETTE-INPUT
RFD	STOP-DISKETTE-INPUT
SDVC	MODIFY-DISKETTE-OUTPUT-STATUS
SDVC	MODIFY-PRINTER-OUTPUT-STATUS
SDVC	MODIFY-TAPE-OUTPUT-STATUS
SDVC	START-DISKETTE-OUTPUT
SDVC	START-PRINTER-OUTPUT
SDVC	START-TAPE-OUTPUT
SDVC	START-TAPE-REPLAY
SDVC	STOP-DISKETTE-OUTPUT
SDVC	STOP-PRINTER-OUTPUT
SDVC	STOP-TAPE-OUTPUT
SDVC	STOP-TAPE-REPLAY
SET-DSSM-OPTIONS	SET-DSSM-OPTIONS
SET-PUBSET-ATTRIBUTES	SET-PUBSET-ATTRIBUTES
SET-REPLUG-READ-MARK	SET-REPLUG-READ-MARK
SEVER	LOCK-USER
SEVER	REMOVE-USER
SEVER	UNLOCK-USER
SHARE	ADD-SHARED-PROGRAM
SHOW-ACCOUNTING-STATUS	SHOW-ACCOUNTING-STATUS
SHOW-CONSLOG	SHOW-CONSLOG-ATTRIBUTES
SHOW-DAB	SHOW-DAB
SHOW-DISK-STATUS	SHOW-DISK-STATUS

continued ➡



<b>ISP command</b>	<b>SDF command</b>
SHOW-JOB-STREAM	SHOW-JOB-STREAM
SHOW-PCS-OPTION	SHOW-PCS-OPTIONS
SHOW-PUBSET-ATTRIBUTES	SHOW-PUBSET-ATTRIBUTES
SHOW-RESOURCE-REQUESTS	SHOW-RESOURCE-REQUESTS
SHOW-SERSLOG	SHOW-SERSLOG-STATUS
SHOW-SS-STATUS	SHOW-SUBSYSTEM-STATUS
SHOW-TRACE-STATUS	SHOW-TRACE-STATUS
SHOW-USER-ATTRIBUTES	SHOW-USER-ATTRIBUTES
SPMGT	SET-SPACE-SATURATION-LEVEL
SQUC	HOLD-SPOOLOUT
STAM	SHOW-MASTER-CATALOG-ENTRY
START-ACCOUNTING	START-ACCOUNTING
START-DAB	START-DAB
START-JOB-STREAM	START-JOB-STREAM
START-PCS	START-PCS
START-SERSLOG	START-SERSLOG
START-SS	START-SUBSYSTEM
START-TRACE	START-TRACE
STATUS	SHOW-JOB-STATUS
STATUS	SHOW-SYSTEM-STATUS
STATUS	SHOW-USER-STATUS
STOP-ACCOUNTING	STOP-ACCOUNTING
STOP-DAB	STOP-DAB
STOP-JOB-STREAM	STOP-JOB-STREAM
STOP-PCS	STOP-PCS
STOP-SERSLOG	STOP-SERSLOG
STOP-SS	STOP-SUBSYSTEM
STOP-TRACE	STOP-TRACE

Table 12: Assignment of ISP to SDF commands for system administration

## 5.3 Conversion of ISP commands by SDF-CONV

ISP commands for conversion to SDF format are classified into user commands and system administration commands. The tables list the ISP commands in alphabetical order, along with the equivalent SDF commands generated by SDF-CONV. In cases where no conversion is performed by SDF-CONV, the reason is outlined. Certain operand configurations cannot be converted. The commands concerned are identified by a reference to the page in the section "Restrictions", page 32ff, where the exceptions are explained.

### 5.3.1 User commands

ISP command	Converted to SDF command	Reason why not converted or notes
ABEND for BS2000 V11 for BS2000 V10	EXIT-JOB -----	(See page 38)  No SDF equivalent
ABORT	CANCEL-PROCEDURE	
ACCOUNT	-----	No one-to-one assignment possible
APPLICATION LINK=... (as only operand) other configurations	REMOVE-DCAM-APPLICATION-LINK  SET-DCAM-APPLICATION-LINK	
AUDIT	-----	Conversion is runtime-dependent
BCNTRL	MODIFY-MSG-OPTIONS	
BREAK	HOLD-PROGRAM	
CALL	CALL-PROCEDURE	
CANCEL	CANCEL-JOB	(See page 42)
CATALOG	CREATE-FILE CREATE-FILE-GENERATION CREATE-FILE-GROUP IMPORT-FILE MODIFY-FILE-ATTRIBUTES MODIFY-FILE-GROUP-ATTRIBUTES	(See page 8)
CATJV STATE=NEW STATE=UPDATE	CREATE-JV MODIFY-JV-ATTRIBUTES	(See page 42)
CHANGE	CHANGE-FILE-LINK	
CONNECTION LINK=... (as only operand) other configurations	REMOVE-DCAM-CONNECTION-LINK  SET-DCAM-CONNECTION-LINK	
COPY	COPY-FILE	(See page 42)
DATA	-----	No SDF equivalent
DCLJV	SET-JV-LINK	
DELON	REMOVE-CJC-ACTION	(See page 42)

continued ➡

ISP command	Converted to SDF command	Reason why not converted or notes
DIAG	-----	Not an ISP command
DO	-----	Conversion is context-dependent; no automatic conversion, but conversion (CALL-PROCEDURE) in the output procedure (See page 36)
DROP	UNLOCK-FILE-LINK	
END	-----	No SDF equivalent
ENDON	END-CJC-ACTION	
ENDP	EXIT-PROCEDURE	
ENTER	ENTER-JOB	(See page 42)
EOF	EOF	
ERAJV	DELETE-JV	(See page 42)
ERASE only file name * file or *SYSfile POS=... VOLUME=...	DELETE-FILE DELETE-SYSTEM-FILE DELETE-FILE-GENERATION EXPORT-FILE	(See page 42)
ESCAPE (without operands)	HOLD-PROCEDURE	
EXECUTE	START-PROGRAM	(See page 42)
FILE for BS2000 V11	CREATE-FILE CREATE-FILE-GENERATION CREATE-TAPE-SET DELETE-TAPE-SET EXTEND-TAPE-SET IMPORT-FILE MODIFY-FILE-ATTRIBUTES MODIFY-FILE-GENERATION- SUPPORT SET-FILE-LINK	(See page 38)
for BS2000 V10	-----	No automatic conversion; conversion proposal is written to a comment line

continued ➡

ISP command	Converted to SDF command	Reason why not converted or notes
FSTATUS VTOC=YES	SHOW-FILE-ATTRIBUTES IMPORT-FILE SUPPORT=BY-FILE-NAME	(See page 42)
GETJV	SHOW-JV	(See page 42)
GETUS	SHOW-USER-SWITCHES	
HELP INF=CD or INF=C	HELP-MESSAGE-INFORMATION HELP-MESSAGE-INFORMATION INFORMATION=MAXIMUM	(See page 42) There is no SDF equivalent for these operand values, but they are converted nevertheless
HOLD	LOCK-FILE-LINK	
IMPORT	IMPORT-FILE SUPPORT=DISK(...)	(See page 42)
INTR	SEND-MESSAGE TO=PROGRAM	
JAC	WRITE-ACCOUNTING-RECORD USER-DATA=(...)	
LOAD	LOAD-PROGRAM	(See page 42)
LOGOFF for BS2000 V11 for BS2000 V10	EXIT-JOB -----	(See pages 38 and 42)  No SDF equivalent
LOGON for BS2000 V11 for BS2000 V10	SET-LOGON-PARAMETERS -----	Is not converted
MARGIN	MODIFY-TERMINAL-OPTIONS	
MRSSTA	SHOW-MRS-CONNECTIONS	
NCANCEL	CANCEL-FILE-TRANSFER	(See page 42)
NCOPY	TRANSFER-FILE	(See page 42)
NSTATUS	SHOW-FILE-TRANSFER	
ON	ADD-CJC-ACTION	
OPTION MSG=... MAXLST=... MAXOPT=... DUMP=... TESTPRIV=...	MODIFY-JOB-OPTIONS MODIFY-JOB-OPTIONS MODIFY-JOB-OPTIONS MODIFY-TEST-OPTIONS MODIFY-TEST-OPTIONS	(See page 42)

continued ➡

ISP command	Converted to SDF command	Reason why not converted or notes
PARAMETER	-----	No SDF equivalent
PASSWORD password only REL=YES	ADD-PASSWORD REMOVE-PASSWORD	(See page 42)
PAUSE	SEND-MESSAGE TO=OPERATOR (WAIT-RESPONSE=YES)	
PRINT TAPE=NO TAPE≠NO	PRINT-FILE WRITE-SPOOL-TAPE	(See pages 38 and 42)
PRIORITY <1..9> <30..255> (<30..255>,EXPRESS)	MODIFY-JOB CHANGE-TASK-PRIORITY MODIFY-JOB ..(START=IMMEDIATE) CHANGE-TASK-PRIORITY	(See page 42)
PROCEDURE	BEGIN-PROCEDURE	
PSWORD	MODIFY-USER-PROTECTION	
PUNCH DEVICE=CENTRAL, TAPE=NO DEVICE=CENTRAL, TAPE=POOLER DEVICE=DISKETTE  DEVIN=... FORM=... HREC=... TRLNUM=..., HDRNUM=...	PUNCH-FILE  -----  WRITE-DISKETTE  WRITE-DISKETTE WRITE-DISKETTE WRITE-DISKETTE TRAILER-EXIT-NUMBER	(See pages 39 and 43)  Pooler tapes are not supported as of BS2000 V11  The operands refer to card readers, which are no longer supported; however, they are still converted
RDTFT	SHOW-FILE-LINK	(See page 43)
RELEASE	REMOVE-FILE-LINK	(See page 43)
REMARK	REMARK	
RESTART	RESTART-PROGRAM	(See page 43)
RESUME with operands without operands	----- RESUME-PROGRAM	Not an ISP command (see page 39)
RFAEND	REMOVE-RFA-CONNECTION	

continued ➡

<b>ISP command</b>	<b>Converted to SDF command</b>	<b>Reason why not converted or notes</b>
RFASTART	SET-RFA-CONNECTION	
RTI	RESUME-PROCEDURE	
SECURE	SECURE-RESOURCE-ALLOCATION	(See pages 39 and 43)
SETJV	MODIFY-JV	(See page 43)
SETSW with operands without operands	MODIFY-JOB-SWITCHES SHOW-JOB-SWITCHES	(See page 43)
SETUS	MODIFY-USER-SWITCHES	
SET-SS-OPTIONS	SET-SUBSYSTEM-OPTIONS	(See page 43)
SKIP	SKIP-COMMANDS IF=NO-CONDITION	(See page 43)
SKIPJV	SKIP-COMMANDS IF=JOB-SWITCHES(...)	
SKIPUS	SKIP-COMMANDS IF=USER-SWITCHES(...)	
SPARAM	MODIFY-SPOOLOUT-OPTIONS	(See page 43)
SQUC REL=	HOLD-SPOOLOUT RESUME-SPOOLOUT	(See page 43)
STAJV	SHOW-JV-ATTRIBUTES	(See page 43)
STAM	SHOW-MASTER-CATALOG-ENTRY	
STATUS ENVIR / LIST / PROG BIAS / CATEGORY / SUMMARY other operands	SHOW-USER-STATUS SHOW-SYSTEM-STATUS SHOW-JOB-STATUS	(See page 43)
STEP	SET-JOB-STEP	

continued ➡

ISP command	Converted to SDF command	Reason why not converted or notes
SYSFILE SYSIPT=... SYSOPT=... TASKLIB=... TASKLIB=(NO)  for BS2000 V11 SYSDTA=... SYSLST=... SYSLSTn=...  SYSOUT=...  for BS2000 V10 SYSDTA=... SYSLST=... SYSLSTn=...  SYSOUT=...	ASSIGN-SYSIPT ASSIGN-SYSOPT SET-TASKLIB REMOVE-TASKLIB  ASSIGN-SYSDTA TO= ASSIGN-SYSLST TO= ASSIGN-SYSLST TO=*SYSLST-NUMBER(n) ASSIGN-SYSOUT TO=  ASSIGN-SYSDTA TO-FILE= ASSIGN-SYSLST TO-FILE= ASSIGN-SYSLST TO-FILE=*SYSLST-NUMBER(n) ASSIGN-SYSOUT TO-FILE=	(See pages 39 and 43)
SYSTATUS	SHOW-SYSTEM-FILE- ASSIGNMENTS	(See page 43)
TCHNG	MODIFY-TERMINAL-OPTIONS	(See pages 39 and 43)
TYPE	SEND-MESSAGE TO=OPERATOR (WAIT-RESPONSE=NO)	
VERIFY REPAIR=YES  REPAIR=ABS  REPAIR=NO	REPAIR-DISK-FILES FILE-STATUS=OPEN(...) REPAIR-DISK-FILES FILE-STATUS=ANY(...) REMOVE-FILE-ALLOCATION-LOCKS	(See page 43)
WAIT	WAIT-EVENT UNTIL=JV(...)	
WHEN	WAIT-EVENT UNTIL=USER-SWITCHES(...)	

Table 13: Conversion of ISP user commands by SDF-CONV



### 5.3.2 System administration commands

ISP command	Converted to SDF command	Reason why not converted or notes
BIAS	MODIFY-SYSTEM-BIAS	
CANCEL , KILL for BS2000 V11 for BS2000 V10	FORCE-JOB-CANCEL -----	(See page 39)
CATEGORY	MODIFY-TASK-CATEGORIES	
CATM STATE=NEW STATE=UPDATE	ADD-MASTER-CATALOG-ENTRY MODIFY-MASTER-CATALOG-ENTRY	(See page 43)
CREATE-SS	START-SUBSYSTEM	
DELETE-SS	STOP-SUBSYSTEM	
ERAM	REMOVE-MASTER-CATALOG-ENTRY	
EXCAT CANCEL FORCE END	CANCEL-CATALOG-EXPORT FORCE-CATALOG-EXPORT EXPORT-PUBLIC-VOLUME-SET	(See pages 39 and 43)
HOLD-SS	HOLD-SUBSYSTEM	
IMCAT	IMPORT-PUBLIC-VOLUME-SET IMPORT-CATALOG	(See pages 39 and 43)
JOIN	-----	Conversion is context-dependent; no automatic conversion, but conversion proposal (ADD-USER or MODIFY-USER) in the comment line (see page 36)
LOADAID	-----	No SDF equivalent
LOADAIDSYS	-----	No SDF equivalent
MSGCONTROL FILE=STD FILE≠STD	RESET-MSG-FILE-ASSIGNMENT MODIFY-MSG-FILE-ASSIGNMENT	(See page 43)
NCHOLD	HOLD-TASK	
NCREL	RESUME-TASK	
RCARD	-----	Card readers are no longer supported
RDIR	REDIRECT-REMOTE-OUTPUT	

continued ➡

ISP command	Converted to SDF command	Reason why not converted or notes
RELEASE-JOB	RESUME-JOB	
RELEASE-JOB-CLASS	RESUME-JOB-CLASS	
RELEASE-JOB-STREAM	RESUME-JOB-STREAM	
RESUME-SS	RESUME-SUBSYSTEM	
RFD USE=INPUT USE=NO	START-DISKETTE-INPUT STOP-DISKETTE-INPUT	(See pages 39 and 43)
SDVC	-----	No one-to-one assignment possible
SEVER RESET REMOVE SET	UNLOCK-USER REMOVE-USER LOCK-USER	(See page 43)
SHARE	ADD-SHARED-PROGRAM	
SHOW-SS-STATUS	SHOW-SUBSYSTEM-STATUS	
SPMGT	SET-SPACE-SATURATION-LEVEL	(See page 43)
SQUC	HOLD-SPOOLOUT	(See page 43)
STAM	SHOW-MASTER-CATALOG-ENTRY	(See page 43)
START-SS	START-SUBSYSTEM	
STATUS BIAS / CATEGORY / SUMMARY ENVIR / LIST / PROG other operands	SHOW-SYSTEM-STATUS  SHOW-USER-STATUS SHOW-JOB-STATUS	
STOP-SS	STOP-SUBSYSTEM	

Table 14: Conversion of ISP commands for system administration by SDF-CONV

## 5.4 SDF-CONV messages

SDF-CONV distinguishes between NOTE, WARNING, ERROR and FAILURE.

If an ERROR or FAILURE occurs, the error message is output to SYSOUT as well as to SYSLST. The SDF-CONV conversion run is then aborted.

If a NOTE or WARNING occurs, the error message appears only in the log (output to SYSLST). After the error message has been issued, SDF-CONV processes the next command line of the input procedure.

The SDF-CONV messages have the following format: CVRxxxx <message-text>

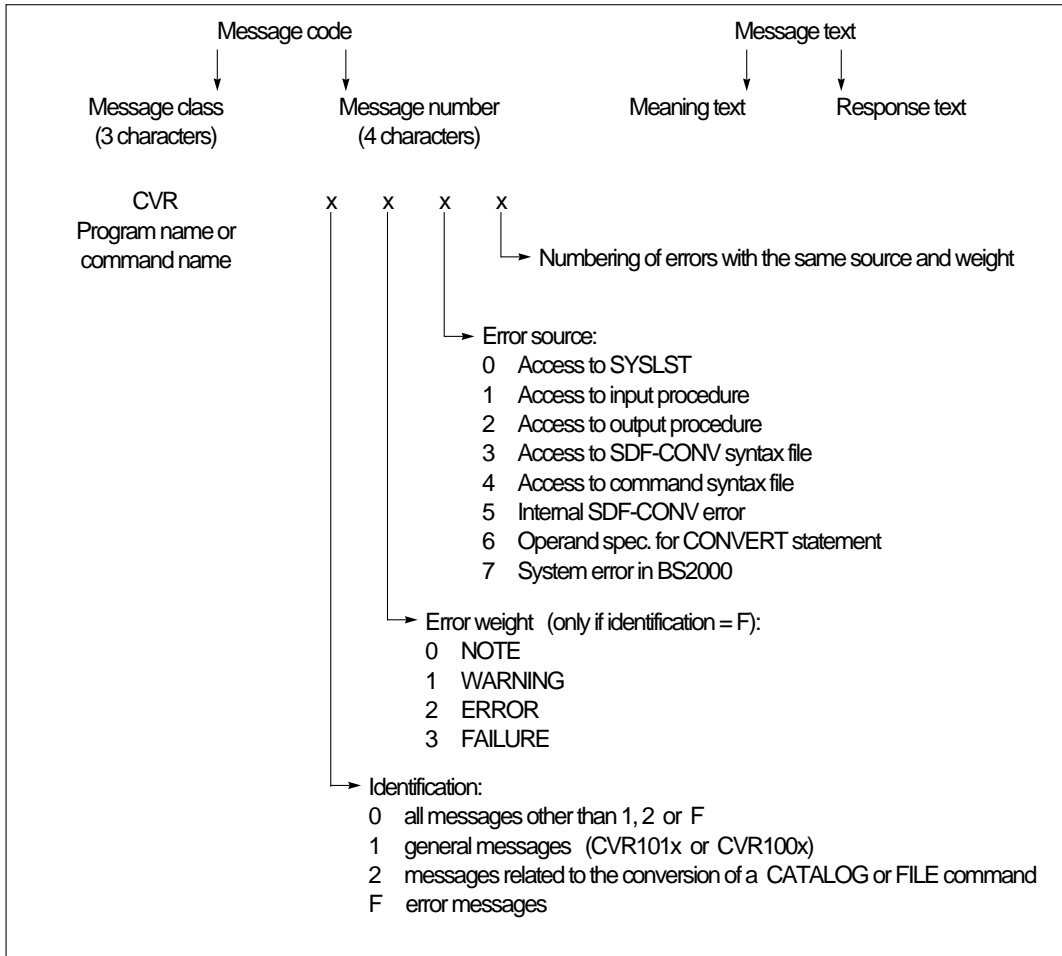


Figure 2: Structure of SDF-CONV messages

In a few cases, a message will be output followed by one or more other error messages that are intended to provide a more precise description of the cause of the error or its impact. Messages of this explanatory nature can have the format "CMDxxxx <message-text>".

More detailed information concerning the cause of the error or possible remedial action can be obtained with the aid of the error code (CVRxxx or CMDxxxx) and the HELP-MSG-INFORMATION command (see the manual "User Commands (SDF Format)" [1]).

### 5.4.1 List of messages (sorted according to message code)

CVRF010 NO INPUT MATCHES THE PATTERN

#### Meaning

The specified wild-cards pattern doesn't match any existing element name/version of the input library

CVRF040 INPUT COMMAND NOT CONVERTED

#### Meaning

The command contains incompatible operands or illegal values.

#### Response

For more detailed information about the restrictions see the BS2000 manual 'SDF-CONV'.

CVRF041 INPUT COMMAND CANNOT BE IDENTIFIED

#### Meaning

The description of this command does not exist in the syntax files from SDF-CONV.

#### Response

Verify if this is an officially documented BS2000 command.

CVRF042 WRONG VALUE SPECIFIED FOR '(&00)' OPERAND

#### Meaning

Input command not converted.

A converted SDF form can be proposed in comments; this is possible if DOCUMENTATION=\*MIN/\*MAX is specified.

Note: the annotated SDF form is a basis for conversion; it is not the converted form

Example: with /FILE, (&00)=DEVICE means that either a not allowed device name was specified or 'SUPPORT=(' (no device) was generated (use DOC=\*MAX)

CVRF043 NO COMMENT LINE CAN BE GENERATED ALTHOUGH REQUESTED THROUGH 'DOCUMENTATION' OPERAND

#### Meaning

The input was not correctly converted from its description in the syntax file because it contains unknown operands.

CVRF060 COMMAND NAME ALREADY SPECIFIED IN 'UNCHANGED-CMDS' OPERAND

CVRF061 COMMAND CAN NEVER BE CONVERTED TO A SECURED SDF FORM; ORIGINAL INPUT REMAINS VALID

**Meaning**

For the conversion of this command, there exists a restriction. However, a converted SDF form can be proposed in comments; this is possible if DOCUMENTATION=\*MIN/\*MAX is specified.

Note: The annotated SDF form is a basis for conversion, it is not the converted form.

CVRF062 COMMAND CAN ONLY BE CONVERTED FOR BS2000 VERSIONS GREATER THAN OR EQUAL TO V11

CVRF063 COMMAND CAN ONLY BE CONVERTED FOR BS2000 VERSIONS LOWER THAN V11

**Meaning**

For the conversion of this command, there exists a restriction. However, a converted SDF form can be proposed in comments; this is possible if DOCUMENTATION=\*MIN/\*MAX is specified.

Note: The annotated SDF form is a basis for conversion, it is not the converted form.

CVRF064 DATA ESCAPE CHARACTER NOT FOLLOWED BY A RIGHT (&00)

**Meaning**

The data escape character generally introduces a procedure or a job variable parameter; in case no such forms can be found, the user is warned and the processing continues in the same record, with next data escape character occurrence.

CVRF100 DMS ERROR '(&00)'. SYSTEM FILE \*SYSLST CANNOT BE OPENED. IN SYSTEM MODE:  
/HELP-MSG DMS(&00)

**Meaning**

Error when opening system file SYSLST. For more detailed information about the DMS error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): DMS error code.

CVRF101 DMS ERROR '(&00)'. SYSTEM FILE \*SYSLST CANNOT BE CLOSED. IN SYSTEM MODE:  
/HELP-MSG DMS(&00)

**Meaning**

Error when closing system file SYSLST. For more detailed information about the DMS error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): DMS error code.

CVRF102 DMS ERROR '(&00)'. WRITING TO SYSTEM FILE \*SYSLST NOT POSSIBLE. IN SYSTEM-MODE: /HELP-MSG DMS(&00)

**Meaning**

For more detailed information about the DMS error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): DMS error code.

CVRF140 INPUT COMMAND IS CONVERTED TO AN EMPTY STRING; ORIGINAL INPUT IS ASSUMED IN OUTPUT FILE

**Meaning**

The input was correctly converted from its description in the syntax file. However, additional handling (mandatory to generate the executable SDF form) reduced it to an empty string. This may happen in some command configurations.

**Response**

Contact the system administrator.

CVRF141 INPUT COMMAND NOT CONVERTED

**Meaning**

The input was not correctly converted from its description in the syntax file because it contains unknown operands. For more detailed information about the restrictions see the BS2000 manual 'SDF-CONV'.

**Response**

If, however, the input procedure runs correctly and no operands are in the restriction list, then contact the system administrator.

CVRF142 WRONG VALUE SPECIFIED FOR OPERAND '(&00)'

**Meaning**

Conversion is not possible because an ambiguous value was specified. For more detailed information about the restrictions see the BS2000 manual 'SDF-CONV'.

CVRF160 USER SYNTAX FILE CANNOT BE DEACTIVATED: RESULT OF CONVERSION MAY BE CORRUPTED

**Meaning**

During conversion of the commands, there is a risk of the definitions in the user syntax file being found. This is why the user syntax file must not be active.

**Response**

Verify whether the /MODIFY-SDF-OPTIONS \*DEAC command works.

CVRF161 THERE ARE NO MORE STATEMENTS TO BE READ: //END IS ASSUMED

**Meaning**

During reading of the SDF-CONV statements, end of file was reached. After termination of the last statement, an END statement is assumed.

CVRF162 DATA PRESERVED FROM UPGRADE TO STATEMENT

**Meaning**

Data lines can be generated as statement lines in S-procedures with PROGRAM-INPUT=\*STMTS(...). Data lines can be excepted from this handling if EXCEPT-AFTER-CMD is specified.

CVRF21A NO SUPPORT OF SEVERAL JOB VARIABLES AS PROCEDURE PARAMETER DEFAULT VALUE

CVRF21B CONTROL-CHARACTERS CANNOT BE ACCEPTED

**Meaning**

Meaningfull command and statement lines cannot contain characters which have no text representation. The use of internal control-characters in SDF-CONV resulted from this observation.

CVRF210 INPUT FILE CANNOT BE OPENED DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF211 INPUT FILE CANNOT BE CLOSED DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF212 INPUT FILE CATALOGED BUT IT HAS WRONG FCBTYPE

**Meaning**

SDF-CONV only supports SAM- and ISAM-files and PLAM- and OSM- libraries.

CVRF213 RECORD CANNOT BE READ FROM INPUT FILE DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF214 INPUT CANNOT BE FOUND DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF215 INPUT LIBRARY IS OF DMS DATA TYPE ISAM BUT IS NOT STRUCTURED AS AN OSM-LIBRARY

CVRF216 PROCESSING OF MULTI-ELEMENTS IS NOT ALLOWED WITH OLD LIBRARY FORMAT 'ISAMLIB'

**Meaning**

OSM-libraries are supported for compatibility reasons; the new library format PLAM should be preferred. Conversion is possible, but the element name must be specified explicitly.

CVRF217 FIRST PROCEDURE LINE IS NOT A COMMAND

**Meaning**

valid procedures must start with a command

CVRF218 (&00) PROCEDURE CANNOT BE CONVERTED TO THE S-PROCEDURE FORMAT

**Meaning**

The JMS -which supports the batch mode- doesn't still support S-PROCedure features. The migration from /[BEGIN-]PROCEDURE to /SET-PROCEDURE-OPTIONS cannot be successfully performed.

CVRF219 NO POSITIONING ON INPUT FILE FIRST RECORD DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system modus or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): Four-digit message number.

CVRF220 OUTPUT FILE CANNOT BE OPENED DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.



CVRF221 OUTPUT FILE CANNOT BE CLOSED DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF222 OUTPUT FILE CANNOT BE DELETED DUE TO '(&00)' ERROR '(&01)'

**Meaning**

Output file already exists and REPLACE-OLD-FILE=\*YES was specified. It must be erased before processing can continue. For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF223 OUTPUT FILE ALREADY CATALOGED

**Meaning**

Output file is already cataloged but REPLACE-OLD-FILE=\*NO was specified. So it cannot be overwritten: processing ends.

CVRF224 WRITING TO OUTPUT FILE NOT POSSIBLE DUE TO '(&00)' ERROR '(&01)'

**Meaning**

For more detailed information about the error code enter /HELP-MSG in system mode or see the BS2000 manual 'System Messages'.

(&00): PLAM or DMS error

(&01): four-digit message number.

CVRF225 SPECIFIED OUTPUT LIBRARY MUST BE OF TYPE PLAM

CVRF230 NO STATEMENT CAN BE READ: //END IS ASSUMED

**Meaning**

Sdf-conv uses the sdf interface which means that the program name and its statements description MUST be available in the syntax file hierarchy of your task

**Response**

Contact the system administrator

CVRF260 (&00) FORM NOT GENERATED AS REQUESTED

**Meaning**

In case of conversion of batch procedures to an invariant-form, the operand context must be set to a compatible value

CVRF261 INHOMOGENEITY BETWEEN 'FROM-FILE' AND 'TO-FILE' OPERAND VALUES

**Meaning**

When FROM-FILE designates several elements/versions to be converted, TO-FILE must allow to designate at least the same number of target elements/versions.

CVRF262 THE OPERAND VALUE FROM 'PROCEDURE-PARAMETERS' IS ERRONEOUS

**Meaning**

The value -when specified- must be bordered with parenthesis.

CVRF280 '(&00)' IS NOT A VALID DATA ESCAPE CHARACTER; NO REPLACEMENT

CVRF281 THE PROCEDURE PARAMETER '(&00)' IS LONGER THAN THE S-PROC ALLOWED LENGTH; PROCESSING CONTINUES HOWEVER

**Meaning**

Only 20 characters can be used to name an S-PROC parameter

**Response**

Use shortest names but pay attention not to rename them to already existing procedure variable names

CVRF282 '(&00)' IS AN INVALID PROCEDURE PARAMETER FORM

**Meaning**

NOT-S procedure parameters must start with an '&' followed by a procedure parameter name

**Response**

Your input procedure is erroneous: correct it

CVRF283 PROCEDURE/JOB VARIABLE PARAMETERS CANNOT BE REPLACED

**Meaning**

each data escape character is replaced in turn by an unambiguous digit value. Unfortunately, the input record already contains all of them.

CVRF340 SYNTAX FILE CONTAINING DESCRIPTION OF ISP COMMANDS TO BE CONVERTED CANNOT BE OPENED BY SDF

CVRF341 SYNTAX FILE CONTAINING DESCRIPTION OF ISP COMMANDS TO BE CONVERTED CANNOT BE CLOSED BY SDF

CVRF342 STANDARD SYSTEM SYNTAX FILE CANNOT BE OPENED BY SDF

**Meaning**

Conversion to an INVARIANT/ACCEPTED form is based on the definition of the command in the system syntax file.

**Response**

Verify whether the SDF version is at least V2.

CVRF343 STANDARD SYSTEM SYNTAX FILE CANNOT BE CLOSED BY SDF

**Meaning**

Conversion to an INVARIANT/ACCEPTED form is based on the definition of the command in the system syntax file.

**Response**

Verify whether the SDF version is at least V2.

CVRF350 INTERNAL ERROR: '(&00)'

CVRF360 'STD' ENTRY FROM 'EXPECT-CONTINUATION' OR 'PRODUCE-CONTINUATION' OPERANDS CANNOT BE ASSOCIATED WITH MEANINGFUL VALUE: PROCESSING ENDS

**Meaning**

In procedures, commands may be split over several lines, provided the continuation character '-' is specified at the end of each line. Its position may vary from system to system and can be read interactively with the /SHOW-SDF-OPTIONS command or via the CMDSTA SDF subprogram call. This subprogram call currently fails.

**Response**

Contact the system administrator

CVRF361 NO MORE MEMORY SPACE CAN BE ALLOCATED. CONTACT THE SYSTEM ADMINISTRATOR

**Meaning**

The memory space limit defined by the system administrator has been reached. No more memory space is available.

CVRF362 REQUESTED MEMORY AREA IS NOT SUFFICIENT

**Meaning**

A buffer of 16 KB is reserved to contain the input command line. As long as a continuation character is found, the contents are appended to this buffer.

**Response**

A command line with such a size is suspicious; verify whether this command really works.

CVRF363 RECORDS OF ILLEGAL LENGTH ARE GENERATED: PROCESSING ENDS

**Meaning**

Commands written to the output procedure must be split into records whose length is lower than or equal to 72. The current record processing fails because subrecords of length  $\leq 0$  or greater than the record from which they were extracted are generated.

**Response**

First check the corresponding input command line or contact the system administrator.

CVRF370 AT LEAST SDF V3 MUST BE INSTALLED

**Meaning**

- The data type FULL-FILENAME with wildcards can only be evaluated with the SDF V3 builtin function CMDVAL.
- The INVARIANT-INPUT form can only be generated as of SDF V3.

**Response**

SDF V3 can be installed as of BS2000 V10.  
Contact the system administrator.

CVR0060 SDF-CONV (&00) READY

**Meaning**

The program is ready to perform (another) conversion and is waiting for (further) statements.

CVR0061 CPU TIME LIMIT REACHED DURING PROGRAM EXECUTION

CVR1000 CONVERSION TERMINATED WITH:

**Meaning**

Every conversion ends with a detailed error log.  
CVR1000 introduces the messages CVR1001 through CVR1004.

CVR1001 FAILURES: (&00)

**Meaning**

FAILURES identify serious errors. When such errors occur, the error message is additionally output to SYSOUT as well as to SYSLST. The SDF-CONV conversion run is aborted following this type of error message.

CVR1002 ERRORS : (&00)

**Meaning**

ERRORS identify serious errors. When such errors occur, the error message is additionally output to SYSOUT as well as to SYSLST. The SDF-CONV conversion run is aborted following this type of error message.

CVR1003 WARNINGS: (&00)

**Meaning**

See message CVR1000.

CVR1004 NOTES : (&00)

**Meaning**

See message CVR1000.

CVR1010    OPTIONS IN EFFECT:

**Meaning**

The operand values used for conversion are logged to \*SYSLST ahead of each conversion log. CVR1010 introduces the messages CVR1011 through CVR1023.

CVR1011    FROM-FILE = (&00)

**Meaning**

See message CVR1010.

CVR1012    EXPECT-CONTINUATION = (&00)-MODE

**Meaning**

See message CVR1010.

CVR1013    PARAMETER-LINES = (&00)

**Meaning**

See message CVR1010.

CVR1014    TO-FILE = (&00)

**Meaning**

See message CVR1010.

CVR1015    PRODUCE-CONTINUATION = (&00)-MODE

**Meaning**

See message CVR1010.

CVR1016    REPLACE-OLD-FILE = (&00)

**Meaning**

See message CVR1010.

CVR1017    SYSTEM-SYNTAX-FILE = (&00)

**Meaning**

See message CVR1010.

CVR1018    GROUP-SYNTAX-FILE = (&00)

**Meaning**

See message CVR1010.

CVR1019    UNCHANGED-CMDS = (&00)

**Meaning**

See message CVR1010.

CVR102A PROCESSING OF LIBRARY '(&00)' COMPLETED

**Meaning**

(&00): library name.

CVR1020 DOCUMENTATION = (&00)

**Meaning**

See message CVR1010.

CVR1021 TARGET-VERSION = (&00)

**Meaning**

See message CVR1010.

CVR1022 OUTPUT-FORM = (&00)

**Meaning**

See message CVR1010.

CVR1023 PROCEDURE-FORMAT = (&00)

**Meaning**

See message CVR1010

CVR1029 PROCESSING OF LIBRARY ELEMENT '(&00)' OF VERSION '(&01)'

CVR2000 INPUT CATALOG COMMAND CAN NEVER BE EXECUTED BECAUSE BOTH BASIC-ACL AND GEN (OR DISP, FIRST, BASE) ARE SPECIFIED. COMMAND CANNOT BE CONVERTED

CVR2001 INPUT CATALOG COMMAND CAN ONLY BE EXECUTED ON NON-FGG FILES, PROVIDED NEW FILE NAME (RENAMING) IS SPECIFIED. IN THIS CASE, GENERATION OPERANDS ARE IGNORED AND COMMAND IS CONVERTED TO /MODIFY-FILE-ATTR

# List of Tables

Table 1:	New and modified operands and operand values for the CONVERT statement . . . .	4
Table 2:	Statements accepted by SDF-CONV . . . . .	11
Table 3:	Control functions in the statement part of SDF-CONV . . . . .	27
Table 4:	Documentation level in the output procedure depending on the values set for the UNCHANGED-CMD and DOCUMENTATION operands . . . . .	29
Table 5:	Non-convertible commands for non-privileged and privileged users . . . . .	36
Table 6:	Non-convertible operand configurations and conditional conversions. . . . .	38
Table 7:	Restrictions affecting procedure parameters as operands . . . . .	42
Table 8:	Notational conventions. . . . .	65
Table 9:	Data types . . . . .	69
Table 10:	Suffixes for data types . . . . .	72
Table 11:	Assignment of ISP to SDF commands. . . . .	77
Table 12:	Assignment of ISP to SDF commands for system administration . . . . .	81
Table 13:	Conversion of ISP user commands by SDF-CONV . . . . .	88
Table 14:	Conversion of ISP commands for system administration by SDF-CONV . . . . .	90





## References

- [1] **BS2000/OSD-BC V1.0**  
User Commands (SDF Format), Volume 1  
User Guide

*Target group*

Nonprivileged BS2000/OSD users (privilege STD-PROCESSING)

*Contents*

This manual contains BS2000/OSD commands A-Q available to the nonprivileged user in the basic configuration of BS2000/OSD V1.0. The user is given hints on command input in interactive and batch mode. Commands R-Z can be found in Volume 2 (order no. U21070-J-Z125-1-7600). Further products discussed include:

- SDF V3.0A
- SDF-P-BASYS V1.0B
- SPOOL V2.7A
- RSO V2.2A
- JV V11.0A
- RFA V11.0A
- FT V5.0A

- [2] **BS2000/OSD-BC V1.0**  
User Commands (SDF Format), Volume 2  
User Guide

*Target group*

Nonprivileged BS2000/OSD users (privilege STD-PROCESSING)

*Contents*

This manual contains BS2000/OSD commands R-Z available to the nonprivileged user in the basic configuration of BS2000/OSD-BC V1.0. The appendix includes notes on SDF-P-BASYS. Commands A-Q can be found in Volume 1 (order no. U2338-J-Z125-11-7600). Further products discussed include:

- SDF V3.0A
- SDF-P-BASYS V1.0B
- SPOOL V2.7A
- RSO V2.2A
- JV V11.0A
- RFA V11.0A
- FT V5.0A

- [3] **SDF V3.0**  
(BS2000/OSD)  
Introductory Guide to the SDF Dialog Interface  
User Guide
- Target group*  
BS2000/OSD users
- Contents*  
This manual describes the interactive input of commands and statements in SDF format. The input modes in guided and unguided dialog are explained in detail. Practical use is facilitated by a number of examples. SDF syntax files are discussed.
- [4] **JV V11.0A**  
(BS2000/OSD)  
Job Variables  
User Guide
- Target group*  
BS2000/OSD users (both privileged and nonprivileged)
- Contents*  
The manual describes the use of the software product JV (Job Variables). It contains descriptions of the commands and macros for JV management and conditional job control.
- [5] **SDF V3.0A**  
(BS2000/OSD)  
SDF Management  
User Guide
- Target group*  
This manual is intended for the system administration and experienced BS2000 users.
- Contents*  
It describes how SDF is installed and administered using SDF commands and the SDF-I and SDF-U utility routines. A full description of the SDF-I and SDF-U statements is included.

**[6] BS2000/OSD-BC V1.0**

System Installation  
User Guide

*Target group*

BS2000/OSD system administration

*Contents*

This manual describes

- the generation of the hardware and software configuration with UGEN
- the following installation services:
  - disk organization with MPVS
  - program system SIR
  - volume installation with SIR
  - configuration update (CONFUPD)
  - utility routine IOCFCOPY

**[7] BS2000/OSD-BC V1.0**

System Administrator's Guide  
User Guide

*Target group*

BS2000/OSD system administration

*Contents*

The manual describes the measures to be taken by the system administration for the management and control of the operating system as well as the commands required for this purpose. The new edition contains a number of new functions and responsibilities, in particular for handling the caching media in BS2000/OSD.

The manual contains the following chapters:

- System administration
- System control and optimization
- Data security
- Data saving
- Automation of system operation
- Commands

**[8] SDF-P V1.0B**  
(BS2000/OSD)

User Guide

*Target group*

BS2000/OSD users and system administrators

*Contents*

The software product SDF-P is a procedure language expanding the BS2000/OSD command language into a programming language. SDF-P supports structured programming. SDF-P V1.0B can be used in BS2000 V10.0A and BS2000/OSD-BC V1.0.

### Ordering manuals

The manuals listed above and the corresponding order numbers can be found in the Siemens Nixdorf *List of Publications*. New publications are described in the *Druckschriften-Neuerscheinungen (New Publications)*.

You can arrange to have both of these sent to you regularly by having your name placed on the appropriate mailing list. Please apply to your local office, where you can also order the manuals.

# Index

## A

- abbreviations 3
- alphanum-name (data type) 66
- assignment
  - input procedure 13
  - of output procedure 28
  - output procedure 12, 15
- assignment ISP - SDF
  - system administration commands 78
  - user commands 73

## C

- card reader 32
- cat (suffix for data types) 72
- cat-id (data type) 66
- command-rest (data type) 66
- commands
  - syntax description 62
  - to be excluded from conversion 12, 18
- commands for system administration
  - assignment ISP - SDF 78
  - conversion ISP - SDF by SDF-CONV 89
- comment lines
  - output procedure 12, 18
- compl (suffix for data types) 70
- components of SDF-CONV 7
- composed-name (data type) 66
- context dependency
  - output procedure 12
- continuation character
  - input procedure 15
  - output procedure 12, 17
- control of dialog guidance 27
- conversion ISP - SDF by SDF-CONV
  - system administration commands 89
  - user commands 83

- conversion log
  - output to SYSOUT 30
- conversion of command language (ISP to SDF) 9
  - examples 45
- conversion of procedure format (non-S to S) 9
  - examples 45
- conversion proposal
  - output procedure 19, 28
- conversion restrictions 32
  - non-convertible commands and operands 36
  - procedure parameters as operands 41
- conversion result
  - output procedure 28
  - SYSOUT 30
- CONVERT statement 12
- corr (suffix for data types) 72
- c-string (data type) 66

**D**

- data lines in input procedure 22, 23
- data types 66
- date (data type) 66
- device (data type) 66
- dialog guidance control 27
- documentation
  - output procedure 12
- documentation level
  - output procedure 18, 28

**E**

- error messages
  - SDF-CONV 91
- examples 45

**F**

- fixed (data type) 66
- full-filename (data type) 67

**G**

- gen (suffix for data types) 72

**H**

- handling of optional operands 20

**I**

- inline comments 28
- input procedure
  - assignment to file or library element 13
  - inline comments 28
- input procedure (procedure to be converted)
  - handling of data lines 22, 23
  - procedure format 21
- installation of SDF-CONV 8
- integer (data type) 68
- ISP commands
  - assignment to SDF commands 73
  - conversion restrictions 32
  - conversion to SDF commands 83
  - non-convertible 36

**J**

- job variable replacement
  - conversion setting 15

**L**

- low (suffix for data types) 70

**M**

- man (suffix for data types) 72
- message file 91
  - installation 8
- MODIFY-SDF-OPTIONS statement 24
- monitoring job variables 31

**N**

- name (data type) 68
- non-S procedures 9

**O**

- odd (suffix for data types) 72
- operating system dependency 19
- operating system version dependency 12
- optional operands in conversion 20
- output
  - into PLAM library (output procedure) 28
  - into SAM file (output procedure) 28
  - monitoring job variables for program monitoring 31
  - of messages during data line conversion 23

- to SYSOUT (overview) 30
- output procedure
  - assignment to file or library element 12, 15
  - comment lines 12
  - context dependency 12
  - conversion proposal 19, 28
  - documentation 12
  - documentation level 18, 28
  - version dependency 12
- output procedure (converted procedure)
  - assignment to file or library element 28
  - contents 28
  - definition of command format 22
  - definition of procedure format 21, 22
  - output format 21, 23

## P

- partial-filename (data type) 68
- procedure file
  - as input procedure 13
  - as output procedure 12, 15
- procedure parameters
  - conversion setting 15
  - restrictions 34
- procedure parameters as operands 41
- product-version (data type) 69
- program monitoring
  - monitoring job variables 31

## R

- REMARK statement 25
- REP file
  - installation 8
  - optional REPs 8
- RESTORE-SDF-INPUT statement 25
- restrictions
  - conversion 32
- restrictions on conversion into S procedures 34

## S

- S procedures 9
  - restrictions on conversion 34
- SDF
  - dialog guidance control 27



- syntax description 62
- SDF commands
  - assignment to ISP commands 73
  - conversion from ISP commands 83
- SDF standard statements
  - END 24
  - MODIFY-SDF-OPTIONS 24
  - overview 11
  - REMARK 25
  - RESTORE-SDF-INPUT 25
  - SHOW-SDF-OPTIONS 25
  - STEP 26
  - WRITE-TEXT 26
- SDF-CONV
  - brief description 1
  - components 7
  - functions 7
  - installation 8
  - operation 9
  - overview of possible conversions 10
- SDF-CONV messages 91
- sep (suffix for data types) 72
- SHOW-SDF-OPTIONS statement 25
- starting SDF-CONV 10
- statement
  - CONVERT 12
- statements
  - overview 11
  - SDF standard statements 24
  - syntax description 62
- STEP statement 26
- structured-name (data type) 69
- subsystem SDF-CONV 7
- suffixes for data types 70
- syntax files
  - assignment 12, 17
  - installation 8
- SYSOUT output 30
- system administration commands
  - conversion restrictions 32

## T

- terminating SDF-CONV 10
- text (data type) 69

time (data type) 69

### U

under (suffix for data types) 70

user (suffix for data types) 72

user commands

assignment ISP - SDF 73

conversion ISP - SDF by SDF-CONV 83

conversion restrictions 32

### V

vers (suffix for data types) 72

version dependency

operating system 12

vsn (data type) 69

### W

wild(n) (suffix for data types) 71

with (suffix for data type) 70

without (suffix for data types) 72

WRITE-TEXT statement 26

### X

x-string (data type) 69

x-text (data type) 69

# Contents

<b>1</b>	<b>Preface</b> .....	<b>1</b>
1.1	Brief product description .....	1
1.2	Target group and summary of contents .....	2
1.3	Changes since the last version of the manual .....	3
<b>2</b>	<b>Components and installation of SDF-CONV</b> .....	<b>7</b>
<b>3</b>	<b>Operation and execution of SDF-CONV</b> .....	<b>9</b>
3.1	Starting and terminating the program .....	10
3.2	Statements .....	11
3.2.1	Overview of all statements .....	11
3.2.2	CONVERT statement .....	12
3.2.3	SDF standard statements .....	24
3.2.4	Control functions .....	27
3.3	Outputs .....	28
3.3.1	Output procedure .....	28
3.3.2	Output to SYSOUT .....	30
3.3.3	Conversion log to SYSLST .....	30
3.3.4	Program monitoring using monitoring job variables .....	31
3.4	Restrictions .....	32
3.4.1	Basic preconditions and restrictions .....	32
3.4.2	Restrictions on the conversion into S procedures .....	34
3.4.3	Commands that cannot be converted .....	36
3.4.4	Non-convertible operand configurations and conditional conversions .....	38
3.4.5	Restrictions on procedure parameters as operands .....	41
<b>4</b>	<b>Examples</b> .....	<b>45</b>
<b>5</b>	<b>Appendix</b> .....	<b>61</b>
5.1	SDF syntax description .....	62
5.2	Assignment of ISP to SDF commands .....	73
5.2.1	User commands .....	73
5.2.2	System administration commands .....	78
5.3	Conversion of ISP commands by SDF-CONV .....	82
5.3.1	User commands .....	83
5.3.2	System administration commands .....	89

**Contents**

---

5.4 SDF-CONV messages ..... 91

5.4.1 List of messages (sorted according to message code) ..... 92

**List of Tables** ..... **103**

**References** ..... **105**

**Index** ..... **109**

---

# SDF-CONV V3.0A (BS2000/OSD)

## User Guide

### *Target group*

All BS2000 users

### *Contents*

The procedure format and command language of procedures can be converted as follows:

- from ISP to SDF command language
- from non-S to S procedure format
- simultaneous conversion of command language and procedure format.

The complete functionality of SDF-CONV is available as of BS2000 V10.0A and SDF V3.0A.

**Edition: February 1994**

**File: SDF\_CONV.PDF**

BS2000 is a registered trademark of Siemens Nixdorf Informationssysteme AG.

Copyright © Siemens Nixdorf Informationssysteme AG, 1994.

All rights, including rights of translation, reproduction by printing, copying or similar methods, even of parts, are reserved.

Offenders will be liable for damages.

All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Delivery subject to availability; right of technical modifications reserved.



## Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

The Internet pages of Fujitsu Technology Solutions are available at

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

and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

## Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@[ts.fujitsu.com](mailto:ts.fujitsu.com).

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter

[http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009