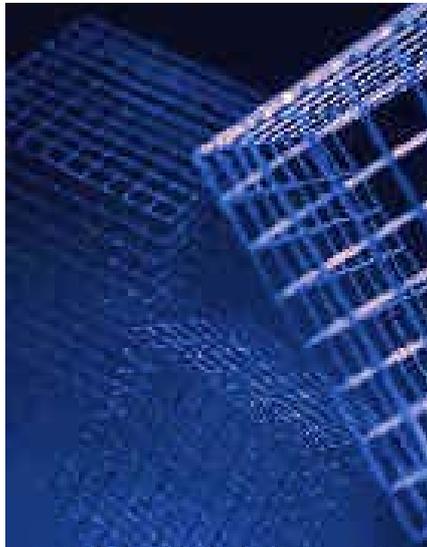


# Loader: SAP DB



Version 7.4



## Copyright

© Copyright 2003 SAP AG.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation.

For more information on the GNU Free Documentaton License see <http://www.gnu.org/copyleft/fdl.html#SEC4>.

## Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

## Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters that appear on the screen. These include field names, screen titles, pushbuttons as well as menu names, paths and options.  Cross-references to other documentation.
<b>Example text</b>	Emphasized words or phrases in body text, titles of graphics and tables.
EXAMPLE TEXT	Names of elements in the system. These include report names, program names, transaction codes, table names, and individual key words of a programming language, when surrounded by body text, for example, SELECT and INCLUDE.
Example text	Screen output. This includes file and directory names and their paths, messages, source code, names of variables and parameters as well as names of installation, upgrade and database tools.
EXAMPLE TEXT	Keys on the keyboard, for example, function keys (such as F2) or the ENTER key.
<b>Example text</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Pointed brackets indicate that you replace these words and characters with appropriate entries.

Loader: SAP DB 7.4 .....	9
Concepts .....	9
Transformation .....	10
System Tables.....	10
SAP DB Loader: Overview.....	11
SAP DB Loader: Commands and SQL Statements.....	11
Commands for Creating a Database Session.....	11
Commands for Loading and Unloading.....	12
Examples of DATALOAD Commands .....	12
SQL Statements .....	13
SAP DB Loader: Call .....	13
Call with the SAP DB Loader CLI (LOADERCLI).....	14
Options.....	15
Command File: -b.....	15
Database Instance: -d.....	16
User Data: -u.....	16
Database Server: -n.....	16
Loader Server: -r.....	17
Maximum Permitted Number of Errors: -E.....	17
Output File: -o[w a].....	17
Substitution Parameter: -p.....	18
InstRoot Directory: -R.....	19
Loader Version: -V.....	19
LOADERCLI Version: -v.....	19
Help: -h.....	19
Command File .....	20
Syntax Rules for Commands Files.....	20
Call with the Perl Interface of the SAP DB Loader.....	21
Perl Classes.....	22
Perl: Loader Class.....	22
Perl: Exception Classes .....	23
Perl: Example 1 .....	23
Perl: Example 2 .....	24
Perl: Example 3 .....	24
Perl: Example 4 .....	25
Perl: Example 5 .....	26
Call with the Python Interface of the SAP DB Loader.....	27
Python Classes.....	29
Python: Loader Class.....	29

Python: Exception Classes .....	30
Python: Example 1 .....	30
Python: Example 2 .....	31
Python: Example 3 .....	31
Python: Example 4 .....	32
Python: Example 5 .....	33
Call with the Java Interface of the SAP DB Loader .....	34
Java Classes .....	35
Java: Loader Class .....	35
Java: Exception Classes .....	36
Syntax Rules for Calling the SAP DB Loader .....	36
SAP DB Loader: Log Functions .....	36
Log File .....	37
System Tables .....	37
TRANSFORMATIONMODEL .....	38
TRANSFORMATIONMONITOR .....	39
TRANSFORMATIONPACKAGES .....	40
TRANSFORMATIONRESOURCES .....	40
TRANSFORMATIONSTATISTICS .....	41
TRANSFORMATIONSTEPS .....	41
Commands .....	42
AUTOCOMMIT Command .....	43
CATALOGEXTRACT Command .....	43
CATALOGLOAD Command .....	44
DATAEXTRACT Command .....	45
DATALOAD Command .....	47
DATAUPDATE Command .....	48
DBEXTRACT Command .....	48
DBLOAD Command .....	50
FASTLOAD Command .....	51
SET Command .....	53
SQLMODE Command .....	53
TABLEEXTRACT Command .....	54
TABLELOAD Command .....	55
USE SERVERDB Command .....	58
USE USER Command .....	58
Syntax Rules for Commands .....	58
Syntax Rules for Setting up a Session .....	59
Syntax Rules for Table Descriptions .....	59
Syntax Rules for Column Descriptions .....	60

Syntax Rules for Describing the Data Stream .....	60
acc_column_spec .....	61
acc_column_spec_mlt .....	62
bool_spec .....	62
cataloginstream_spec .....	62
catalogostream_spec .....	63
code_page_spec .....	63
code_spec .....	65
column_assignment .....	65
column_descriptor .....	66
column_id .....	67
column_id_spec .....	67
column_names .....	69
compare_operator .....	69
condition .....	70
configurationstream_spec .....	70
database_name_statement .....	71
datainstream_spec .....	71
dataostream_spec .....	71
date_spec .....	71
delimiter_spec .....	72
duplicates_clause .....	72
extract_files_spec .....	73
field_format .....	73
field_pos .....	74
format_spec .....	75
generate_spec .....	76
if_condition .....	76
instream_spec .....	77
int_spec .....	77
key_column_spec .....	78
lit_column_spec .....	78
load_column_spec .....	79
load_column_spec_mlt .....	79
longfile_code_spec .....	80
longfile_spec .....	81
longfile_spec_mlt .....	81
mediumtype_spec .....	82
noheader_spec .....	82
null_assign .....	83

null_condition.....	84
null_spec.....	85
number_spec.....	86
numerical_functions.....	86
order_clause.....	87
output_column.....	87
output_column_list.....	88
outstream_spec.....	89
packagestream_spec.....	89
part_spec.....	89
restart.....	90
restore_spec.....	90
round_or_trunc_spec.....	91
scale_spec.....	92
select_expression.....	92
separator_spec.....	93
set_column_spec.....	94
set_column_spec_mlt.....	94
sequence_number.....	94
simple_column_spec.....	95
simple_condition.....	95
sql_mode.....	96
standard_code_spec.....	97
standard_date_mask.....	97
standard_time_mask.....	98
standard_timestamp_mask.....	99
stream_extract.....	99
stream_format_spec.....	99
table_name.....	100
table_spec.....	101
time_spec.....	101
timestamp_spec.....	102
usage_spec.....	102
user_statement.....	103
Keywords.....	103
Keywords A - C.....	104
Keywords D - E.....	104
Keywords F - K.....	105
Keywords L - P.....	105
Keywords R - S.....	106

Keywords T - Z .....	106
Terms and Procedures .....	107
Binary Values .....	107
Data .....	108
Application Data.....	108
Metadata .....	108
Data Stream .....	108
Data Format.....	109
Format of the Data Stream .....	109
COMPRESSED.....	109
FORMATTED.....	111
FORMATTED BINARY .....	111
PAGES.....	112
RECORDS .....	112
Direction.....	113
Data Types .....	113
Internal Database Data Types.....	114
External Data Types .....	114
Converting the Data Types.....	114
Unloading LONG Values .....	115
Each LONG Value to Be Unloaded in a Separate LONG Data Stream .....	116
All LONG Values to Be Unloaded in One LONG Data Stream .....	116
HEX Values .....	117
Plain Text Values .....	117
Loading Binary Values .....	118
Loading LONG Values .....	118
Each LONG Value to Be Inserted in a Separate LONG Data Stream .....	119
All LONG Values to Be Inserted in One LONG Data Stream.....	120
Code Attribute for a Data Stream with LONG Values .....	121
Medium.....	122
Metadata.....	122
Metadata of the Application Data .....	122
Metadata of the Database Catalog.....	122
Metadata of the Transformations.....	123
Selecting Data Records.....	123
System Table CODEPAGE.....	124

## Loader: SAP DB 7.4

This documentation describes the SAP DB Loader tool, which can be used as of SAP DB version 7.4.



For general information about the SAP DB database system, see the documentation [The SAP DB Database System](#) and the following Internet address: [www.sapdb.org](http://www.sapdb.org).

- [Architecture of the SAP DB Loader](#)
- [Concepts \[Page 9\]](#)
- [SAP DB Loader: Overview \[Page 11\]](#)
- [SAP DB Loader: Commands and SQL Statements \[Page 11\]](#)
- [SAP DB Loader: Call \[Page 13\]](#)
- [SAP DB Loader: Log Functions \[Page 36\]](#)
- [Commands \[Page 42\]](#)
- [Terms and Procedures \[Page 107\]](#)



### Concepts

The SAP DB Loader uses the concepts of the SAP DB database instance and also its own concepts.

### Concepts of the SAP DB Database

For the entire concepts of the SAP DB database, see the *Reference Manual: SAP DB 7.4*, [Concepts](#) section. In particular, the SAP DB Loader uses the following concepts:

- [Database Catalog](#)
- [Application Data](#)
- [System Tables](#)

### Concepts of the SAP DB Loader

- [Transformation \[Page 10\]](#)
- [System Tables \[Page 37\]](#)

The SAP DB Loader provides functions to load, unload, and format (transform) [data \[Page 108\]](#) between different data sources and data targets using different [data streams \[Page 108\]](#).

The language scope of the SAP DB Loader allows the transformation of the data of all users of a SAP DB database, a user, a table of a user, or a set of qualified columns of a table or multiple tables ([Commands for Loading and Unloading \[Page 12\]](#)).

[Media \[Page 122\]](#) and [database instances](#) are supported as physical data sources and physical data targets.

The Loader uses [metadata \[Page 122\]](#), which can be in different formats, for the logical description of data and its transformation.

When the system tables are filled, this can, among other things, be used for restarting a transformation.



## Transformation

The SAP DB Loader provides functions to load, unload, and format [data \[Page 108\]](#) between different data sources and data targets using different [data streams \[Page 108\]](#).

The action connected to this is known as a transformation and is described using entries in the SAP DB Loader-specific [system tables \[Page 37\]](#).

A transformation is identified using a SAP DB database instance, a transformation type, the desired part of the database catalog, and the data streams.

- SAP DB Loader connection:  
Servernode, Serverdb, Username, Password, SQL Mode, ASCII CodePage, TransactionSize, MaxErrorCount, BackupTool
- SAP DB Database connection:  
Servernode, Serverdb
- Transformations type:  
EXTRACT: Data is to be unloaded.  
LOAD: Data is to be loaded.  
UPDATE Data is to be changed.
- Part of the database catalog:  
You can define the scope of the catalog objects of a SAP DB database involved in a transformation.  
ALL: Entire database catalog  
USER: Part of the database catalog that belongs to the specified user  
TABLE: Part of the database catalog that belongs to the specified table  
You can specify a set of qualified columns of one or more tables.  
You can determine the commands and their runtime information from the [system tables \[Page 37\]](#) for the parts of the database catalog specified using ALL and USER.
- [Data stream \[Page 108\]](#)



## System Tables

During the installation of the database system, the following system tables are created for the SAP DB Loader:

- [TRANSFORMATIONMODEL \[Page 38\]](#)
- [TRANSFORMATIONMONITOR \[Page 39\]](#)
- [TRANSFORMATIONPACKAGES \[Page 40\]](#)
- [TRANSFORMATIONRESOURCES \[Page 40\]](#)
- [TRANSFORMATIONSTATISTICS \[Page 41\]](#)
- [TRANSFORMATIONSTEPS \[Page 41\]](#)

If you perform a load or unload process that spans multiple tables, the system tables are filled with the corresponding information.

The SAP DB Loader logs information about the runtime behavior for load and unload processes in the system tables.

You can also configure a transformation using the system tables.

The SAP DB Loader uses the entries in the system tables to be able to restart commands. You can use this ability to restart with the RESTART option ([restart \[Page 90\]](#)) of the command.



## SAP DB Loader: Overview

### [SAP DB Loader: Commands and SQL Statements \[Page 11\]](#)

- [Commands for Creating a Database Session](#)
- [Commands for Loading and Unloading](#)
- [SQL Statements](#)

### [SAP DB Loader: Call \[Page 13\]](#)

- [Call with the SAP DB Loader CLI \(LOADERCLI\)](#)
- [Call with the Perl Interface of the SAP DB Loader](#)
- [Call with the Python Interface of the SAP DB Loader](#)
- [Call with the Java Interface of the SAP DB Loader](#)

### [SAP DB Loader: Log Functions \[Page 36\]](#)

- [Log File](#)
- [System Tables](#)



## SAP DB Loader: Commands and SQL Statements

- [Commands for creating a database session \[Page 11\]](#)  
The commands and SQL statements for loading and unloading can only be executed after a database session has been successfully created. This ensures that only suitably authorized owners can load and unload.
- [Commands for Loading and Unloading \[Page 12\]](#)  
A significant function of the SAP DB Loader is loading and unloading the database catalog, parts of the database catalog, and application data.
- [SQL statements \[Page 13\]](#)  
The SAP DB Loader can execute the SQL statements required for loading and unloading.



## Commands for Creating a Database Session

The SAP DB Loader provides the following [commands \[Page 42\]](#), which you can use to create a [database session](#):

- [AUTOCOMMIT Command \[Page 43\]](#)
- [SET Command \[Page 53\]](#)
- [SQLMODE Command \[Page 53\]](#)
- [USE SERVERDB Command \[Page 58\]](#)

- [USE USER Command \[Page 58\]](#)

See also:

[SAP DB Loader: Commands and SQL Statements \[Page 11\]](#)



## Commands for Loading and Unloading

The language scope of the SAP DB Loader allows you load or unload, in increasing granularity, a complete database, the objects of a user, individual tables, or a set of columns of a table.

What is to be loaded/unloaded?	SAP DB Loader Command
Database	<a href="#">DBEXTRACT [Page 48]</a> <a href="#">DBLOAD [Page 50]</a>
Database catalog	<a href="#">CATALOGEXTRACT [Page 43]</a> ALL, <a href="#">CATALOGLOAD [Page 44]</a> ALL
Database catalog objects and the application data of a user	CATALOGEXTRACT USER CATALOGLOAD USER  <a href="#">TABLEEXTRACT [Page 54]</a> USER <a href="#">TABLELOAD [Page 55]</a> USER
Tables	CATALOGEXTRACT TABLE CATALOGLOAD TABLE  TABLEEXTRACT TABLE TABLELOAD TABLE  <a href="#">DATAEXTRACT [Page 45]</a> <a href="#">DATALOAD [Page 47]</a>
Set of qualified columns of a table	DATAEXTRACT DATALOAD  <a href="#">FASTLOAD [Page 51]</a>  <a href="#">DATAUPDATE [Page 48]</a>

See also:

[SAP DB Loader: Commands and SQL Statements \[Page 11\]](#)



## Examples of DATALOAD Commands

```

DATALOAD TABLE sqltravel20.reservation IF POS 4 >= '11.08.1999"
  rno      1
  cno      2
  hno      3
  arrival  4
  departure 5
  roomtype 6
INFILE 'reservation.data'
  DATE 'dd.mm.yyyy'

DATALOAD TABLE sqltravel10.room
  IF (POS 41-44 REAL < '400.00')
  AND
    (POS 41-44 REAL >= '50.00')
```

```

nno      01-04 INTEGER
roomtype 09-15
price    41-44 REAL
INFILE 'room.data' FORMATTED

DATALOAD TABLE edemo.products
productid 01-10
productname 11-51          NULL IF POS 10-12 = '  '
unitinstock 52-55 INTEGER NULL IF POS 52-55 INTEGER < '0'
unitprice  60-67 DECIMAL (2) NULL IF POS 60 <> 'X'
                                     OR POS 60-67 DECIMAL < '0'

INFILE 'products.data' FORMATTED

DATALOAD TABLE edemo.suppliers
supplierid 1
phone      2 DEFAULT NULL
address    3 DEFAULT NULL
INFILE 'suppliers.data'
NULL '-    '

DATALOAD TABLE edemo.orders
oderid     1
orderdate  DATE
requireddate 2
INFILE 'orders.data'

```



## SQL Statements

The SAP DB Loader can process the SQL statements required for loading and unloading. DDL statements, in particular, are vital for the loading of data (*Reference Manual: SAP DB 7.4*, [SQL Statements: Overview](#) section).

Database queries, however, return only status messages.

COMMIT and ROLLBACK statements are only active in the mode AUTOCOMMIT OFF ([AUTOCOMMIT Command \[Page 43\]](#)).

**See also:**

[SAP DB Loader: Commands and SQL Statements \[Page 11\]](#)



## SAP DB Loader: Call

The SAP DB Loader processes [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#). Commands and SQL statements can be processed in the background or be embedded in programming languages.

The call [Loader](#) varies, depending on which client of the SAP DB Loader you are using.

- [Call with SAP DB Loader CLI \(LOADERCLI\) \[Page 14\]](#)
- [Call with the Perl Interface of the SAP DB Loader \[Page 21\]](#)
- [Call with the Python Interface of the SAP DB Loader \[Page 27\]](#)
- [Call with the Java Interface of the SAP DB Loader \[Page 34\]](#)

**See also:**

[Syntax Rules for Calling the SAP DB Loader \[Page 36\]](#)



This documentation contains only a short summary of the functions of the script languages that can be used in SAP DB Loader. For more information, see the vendor documentation for Perl, Python, or Java.



## Call with the SAP DB Loader CLI (LOADERCLI)

The [Call of the SAP DB Loader \[Page 13\]](#) can be performed using the SAP DB Loader CLI (LOADERCLI) client.

### Call

```
loadercli [<options>] -b <command_file>
```

You can specify [options \[Page 15\]](#) when calling the SAP DB Loader. The desired [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#) must be stored in a [command file \[Page 20\]](#). When calling the SAP DB Loader, you must specify the command file using the option `-b <command_file>`.

### Process Flow

During the creation of a connection to the database instance, the SAP DB Loader first evaluates the specified options. You must specify at least the name of the database instance (Option `-d <database_name>`). The commands and SQL statements in the command file are then executed.



```
loadercli -d [Page 16] demodb -u [Page 16] sqltravel01,travel01 -b  
[Page 15] command.dat -E [Page 17] 20
```

The SAP DB Loader creates a connection to the database instance `demodb` for the user `sqltravel01`. The Loader processes the commands and statements contained in the command file `command.dat`. If errors (return code `<> 0`) occur, the processing of the command file is terminated as soon as 20 error messages occur.

### Result

The SAP DB Loader executes the commands and SQL statements specified in the command file. The SAP DB Loader writes a [log file \[Page 37\]](#)

### Errors

If an error occurs, the Loader terminates the processing of the commands and SQL statements (either immediately, or after the number of errors specified in option `-E` has occurred). It is not possible to react to the errors when using the SAP DB Loader CLI.



To enable you to respond to errors, the SAP DB Loader functions are available as a library for the Perl and Python script languages.

see also:

[Call with the Perl Interface of the SAP DB Loader \[Page 21\]](#)

[Call with the Python Interface of the SAP DB Loader \[Page 27\]](#)



## Options

When [calling the Loader with SAP DB Loader CLI \(LOADERCLI\) \[Page 14\]](#), you can specify options:

```
loadercli [<options>] -b <command_file>
```

### Options

<a href="#">-b &lt;command_file&gt; [Page 15]</a>
<a href="#">-d &lt;database_name&gt; [Page 16]</a>
<a href="#">-u &lt;userid&gt;,&lt;password&gt; [Page 16]</a>
<a href="#">-n &lt;server_node&gt; [Page 16]</a>
<a href="#">-r &lt;server_node&gt; [Page 17]</a>
<a href="#">-E &lt;number&gt; [Page 17]</a>
<a href="#">-o[w a] &lt;file_name&gt; [Page 17]</a>
<a href="#">-p &lt;number&gt;&lt;substitution_string [Page 18]&gt;</a>
<a href="#">-R &lt;directory_name&gt; [Page 19]</a>
<a href="#">-V [Page 19]</a>
<a href="#">-v [Page 19]</a>
<a href="#">-h [Page 19]</a>

## Use

When calling the SAP DB Loader, you usually specify the name of the user (option `-u`) and of the database instance (option `-d`). This creates a connection between the [Loader](#) and the specified database instance for this user.

You must transfer all required [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#) in a [command file \[Page 20\]](#) (option `-b`).



```
loadercli -d demodb -u sqltravel01,travel01 -b command.dat
```

You can also call the SAP DB Loader specifying only the database instance name (option `-d`). You must transfer all other [commandos to create a database session \[Page 11\]](#) in the command file to create the connection between the SAP DB Loader and this or another database instance.



## Command File: -b

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for transferring the name of the [command file \[Page 20\]](#)

### Syntax

```
-b <command_file>
```

<code>&lt;command_file&gt;</code>	Command File
-----------------------------------	--------------

### Use

The contents of the specified command file are transferred to the Loader.



## Database Instance: -d

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for transferring the name of the [database instance](#)

### Syntax

**-d** <database\_name>

<database_name>	Name of the database instance to which all <a href="#">commands [Page 42]</a> refer
-----------------	---

### Use

The SAP DB Loader creates a connection to the specified database instance.



The name of the database instance can also be transferred with the [USE SERVERDB \[Page 58\]](#) or [USE USER \[Page 58\]](#) commands in the [command file \[Page 20\]](#), and can therefore be changed during the database session with the SAP DB Loader.



## User Data: -u

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for transferring user data

### Syntax

**-u** <userid>,<password>

<userid>	User name
<password>	User password

### Use

A [database session](#) is created for the specified user.



User data can also be transferred with the [USE USER \[Page 58\]](#) command in the [command file \[Page 20\]](#).



## Database Server: -n

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for transferring the server name of the [database instance](#)

### Syntax

**-n** <server\_node>

<server_node>	Node name of the server on which the database instance is located
---------------	---

	Default value: Name of the local server
--	---

## Use

The SAP DB Loader creates a connection to the specified database instance.



The host node name can also be transferred with the [USE SERVERDB \[Page 58\]](#) or [USE USER \[Page 58\]](#) commands in the [command file \[Page 20\]](#).



## Loader Server: -r

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for transferring the name of the Loader server

## Syntax

**-r** <server\_node>

<server_node>	Node name of the server, on which the <a href="#">Loader</a> is located, Default value: Name of the local server
---------------	--

## Use

When the SAP DB Loader CLI is called, a connection is created to the Loader on the specified server.



## Maximum Permitted Number of Errors: -E

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for transferring the permissible number of errors

## Syntax

**-E** <number>

<number>	Permitted number of <a href="#">commands [Page 42]</a> with errors
----------	--

## Use

You use this option to specify after how many commands with errors the SAP DB Loader stops processing the [command file \[Page 20\]](#).



## Output File: -o[w|a]

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for specifying the output file

## Syntax

**-o[w|a]** <file\_name>

<file_name>	Name of the file to which the output of the SAP DB Loader is to be written
-------------	--

	Default value: Screen
--	-----------------------

## Use

The output of the SAP DB Loader is written to the specified file.

### Option `-ow`

The file is opened and any existing entries are deleted.

### Option `-oa`

The file is opened and the outputs are appended to the end of the file without any existing entries being deleted.



This option exclusively refers to output from the SAP DB Loader. The program still writes a [log file \[Page 37\]](#) irrespectively.



## Substitution Parameter: `-p`

[Option \[Page 15\]](#) for [calling the SAP DB Loader with LOADERCLI \[Page 14\]](#) for transferring a substitution parameter

## Syntax

`-p<number> <substitution_value>`

<code>&lt;number&gt;</code>	Defines the number <i>n</i> of the placeholder you want to substitute in the <a href="#">command file [Page 20]</a> , where $1 \leq n \leq 9$ .
<code>&lt;substitution_value&gt;</code>	Value that substitutes the placeholder in the command file

## Use

Use this option to specify values for placeholders in the command file. These values can be any character strings. You can specify nine different parameters (and placeholders) for each command file.

In the command file, enter the placeholder in the form `&<number>`.



Entry in the command file:

```
DATAEXTRACT FOR DATALOAD TABLE customer OUTFILE '&1' OUTFILE
'customer.data'
DATAEXTRACT FOR FASTLOAD TABLE article OUTFILE '&1' APPEND
OUTFILE 'article.data'
```

You can specify the name of the command file with the call of the SAP DB Loader CLI, using the option `-p` :

```
loadercli ... -p1 command.dat
```



## InstRoot Directory: -R

[Option \[Page 15\]](#) for [calling the Loader with LOADERCLI \[Page 14\]](#) for setting up the connection to the [Loader](#)

### Syntax

**-R** <dependent\_path>

<dependent_path>	Name of the directory (InstRoot) on the server of the Loader that contains the programs that are dependent on the version of the database software Default value: none
------------------	---

### Use

If multiple versions of the Loader are installed on the server, you can start a specific version of the Loader. If you do not specify a directory name, the Loader for the newest version of the database software is started.



## Loader Version: -V

[Option \[Page 15\]](#) when [calling the Loader with LOADERCLI \[Page 14\]](#) to determine the version of the server of the SAP DB Loader ([Loader](#))

### Syntax

**-V**



## LOADERCLI Version: -v

[Option \[Page 15\]](#) when [calling the Loader with LOADERCLI \[Page 14\]](#) to determine the version of the client of the SAP DB Loader (SAP DB Loader CLI (LOADERCLI))

### Syntax

**-v**



## Help: -h

[Option \[Page 15\]](#) for displaying information (help) for all available options when [calling the Loader with LOADERCLI \[Page 14\]](#)

### Syntax

**-h**

## Command File

When [calling the SAP DB Loader with the LOADERCLI \[Page 14\]](#), you must store the [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#) that are to be evaluated in a special file, the command file.

When calling the SAP DB Loader, by specifying the [option -b \[Page 15\]](#), you pass the name of the command file to the [Loader](#).



```
loadercli -d demodb -u sqltravel01,travel01 -n PCnew -b
command.dat
```

The commands and SQL statements are in the file `command.dat`.

## Use

The commands and SQL statements in the command file that you specify are individually processed by the SAP DB Loader in the specified order.

**See also:**

[Syntax Rules for Command Files \[Page 20\]](#)

## Syntax Rules for Commands Files

Note the following syntax rules for [command files \[Page 20\]](#).

### General Syntax Rules when Calling the SAP DB Loader

[Syntax Rules for Calling the SAP DB Loader \[Page 36\]](#)

### Delimiting Lines Between Commands

Individual commands in a command file are separated by a line, at the beginning of which there is a double forward slash `//`.

### Comments

After a double forward slash `//` at the start of the line, you can enter comments. The comments are ignored by SAP DB Loader during the processing of the command file.



```
CREATE TABLE customer
(cno          FIXED(4) CONSTRAINT cno BETWEEN 1 AND 9999  KEY,
salutn       CHAR(7)  CONSTRAINT salutn IN ('Mr', 'Ms', 'compan
y'),
name         CHAR(10) NOT NULL,
zip          CHAR(5)  CONSTRAINT SUBSTR(zip_dom,1,5) LIKE '(0-
9)(0-9)(0-9)(0-9)(0-9)',
address      CHAR(25) NOT NULL)
//
FASTLOAD TABLE customer
cno          1
name         2
zip          3
address      4
INFILE 'customer.data'
//create index
```

```

CREATE INDEX customer_index ON customer (name)
//
DATAEXTRACT cno, name, zip, address from customer
OUTFIELDS
  cno          1
  name         2
  zip          3
  address      4
OUTFILE 'newcustomer.data'

```



## Call with the Perl Interface of the SAP DB Loader

You can [call the SAP DB Loader \[Page 13\]](#) through the processing of a Perl script.

### Prerequisites

Perl is installed on the host.

The following Perl modules are delivered with the SAP DB Loader program to allow you to create Perl scripts that can be processed by the [Loader](#).

- Microsoft Windows 2000: loaderaperl.dll, loader.pm, instperl.pl
- UNIX: loadercperl.so, loader.pm, instperl.pl
- HP: loadercperl.sl, loader.pm, instperl.pl



Open the `instperl.pl` file. When you do this, the files are copied to the relevant directories.

### Call

```
perl <perl_script_file> [<argument> ...]
```

You can optionally specify arguments `<argument>`.

#### Frequently used arguments:

<user_name>	User name
<password>	Password
<database_name>	Name of database instance
<data_path>	Directory in which the data file is stored
<server_node>	Name of the server on which the Loader is located



```
perl sample samplename secret TST C:\data\sapdb
```

The Perl script `sample` is called. The logon to the Loader is performed for the user `samplename` on the database instance `TST`. The data file is in the directory `C:\data\sapdb`.

### Template for Background Files

The following section contains examples of background files in Perl. Their equivalents in the [Python \[Page 27\]](#) script language are also provided for comparison.

Example 1		
-----------	--	--

Build a Perl module with reference to the SAP DB Perl Libraries, parse the call arguments	<a href="#">Perl [Page 23]</a>	<a href="#">Python [Page 30]</a>
Example 2 Create a database session with the Loader Log on to the database instance Log off	<a href="#">Perl [Page 24]</a>	<a href="#">Python [Page 31]</a>
Example 3 Create a database session Log on to the database instance Query whether a table exists by querying the error code Create a table without querying the error code Log off	<a href="#">Perl [Page 24]</a>	<a href="#">Python [Page 31]</a>
Example 4 Create a database session Log on to the database instance Create a table and query the error code Use <a href="#">command(s) [Page 42]</a> to load data into the table and query the error code Log off	<a href="#">Perl [Page 25]</a>	<a href="#">Python [Page 32]</a>
Example 5 Create a database session Log on to the database instance Use <a href="#">command(s)</a> to load data into table and intercept exceptions Log off	<a href="#">Perl [Page 26]</a>	<a href="#">Python [Page 33]</a>

[Perl Classes \[Page 22\]](#)



## Perl Classes

Use the following classes to [call the SAP DB Loader with Perl scripts \[Page 21\]](#):

- [Class Loader \[Page 22\]](#)
- [Exception Classes \[Page 23\]](#)



## Perl: Loader Class

### Constructor: Loader (<server\_node>, <database\_name>)

<server_node>	Server node name
<database_name>	Name of database instance

Creation of a connection to the [Loader](#)

If the server node name and name of the database instance are specified, the system assumes that the database instance, data, and Loader are located on a remote server.

If only the name of the database instance is specified, the system assumes that the database instance, data, and Loader are located on the local server. The suitable Loader is determined from the release of the specified database instance.

If neither the server node name nor the name of the database instance is specified, the system establishes a connection to the newest Loader on the local system.

The session is closed again when the object is deleted with `undef $session`.



```
$session = loader::Loader ('p12345', 'mydb')
```

## Method: `cmd (<command_string>)`

An [SQL statement \[Page 13\]](#) or Loader [command \[Page 42\]](#) is executed.

The script is terminated if the command fails.



```
$output = $session->cmd ("DATALOAD TABLE customer".
                        "cno      1-4".
                        "name    6-12".
                        "zip      14-18".
                        "city     20-31".
                        "INFILE '$data_path\customer.dat' ")
```

## Method: `sql (<command_string>)`

An SQL statement or a Loader command is executed.

If a command fails, execution of the script is terminated.

If an SQL statement fails, a return code is output.



```
$result = $session->sql ('EXISTS TABLE MYTABLE')
```



## Perl: Exception Classes

An exception is a string beginning with either `CommunicationError` or `LoaderServerError`.



```
LoaderServerError: 25011 SQL error -3005 = Invalid SQL
Statement
```

Example of use in a script:

```
eval {$session->cmd ('complete nonsense;')} ;
if ($?) { print "command failed: $@\n"; }
```

Print output:

```
command failed: LoaderServerError: 25011 SQL error -3005 =
Invalid SQL Statement
```



## Perl: Example 1

Example of [Calling the Loader with Perl Script \[Page 21\]](#)

Create a Perl module with reference to the SAP DB Perl Libraries  
Parse the call arguments:

```
# Reference to the SAP DB Perl Library
# -----

use SAP::DBTECH::loader;

# Parse the call arguments
# -----

$userid = $ARGV[0];
$password = $ARGV[1];
$database_name = $ARGV[2];
$data_path = $ARGV[3];
$server_node = "localhost";
```



## Perl: Example 2

Example of [Calling the Loader with Perl Script \[Page 21\]](#)

Create a database session with the SAP DB Loader  
Log on to the database instance  
Log off

```
# Reference to the SAP DB Perl Library
# -----

use SAP::DBTECH::loader;

# Parse the call arguments
# -----

$userid = $ARGV[0];
$password = $ARGV[1];
$database_name = $ARGV[2];
$data_path = $ARGV[3];
$server_node = "localhost";

# Create a database session for the Loader
# -----

$session = loader::Loader ($server_node, $database_name);

# Log on to the database instance
# -----

$session->cmd("use user $userid $password;");

# Log off
# -----

undef $session
```



## Perl: Example 3

Example of [Calling the Loader with Perl Script \[Page 21\]](#)

Create a database session  
Log on to the database instance  
Query whether table exists by querying the error code

Create a table without querying error code  
Log off

```
# Reference to the SAP DB Perl Library
# -----

use SAP::DBTECH::loader;

# Parse the call arguments
# -----

$userid = $ARGV[0];
$password = $ARGV[1];
$database_name = $ARGV[2];
$data_path = $ARGV[3];
$server_node = "localhost";

# Create a database session for the Loader
# -----

$session = loader::Loader ($server_node, $database_name);

# Log on to the database instance
# -----

$session->cmd("use user $userid $password;");

# Query whether the table exists by querying the error code
# The sql method is used to do this
# -----

$rc = $session->sql('EXISTS TABLE CUSTOMER')

If $rc!=0

# Create the table CUSTOMER
# -----

$session->cmd ( 'CREATE TABLE customer ( '.
               'cno                FIXED(4), '.
               'name                CHAR(10) ASCII, '.
               'zip                 CHAR(5)  ASCII, '.
               'city                CHAR(12) ASCII, '.
               'PRIMARY KEY (cno) ')

# End the database session
# -----

undef $session
```



## Perl: Example 4

Example of [Calling the Loader with Perl Script \[Page 21\]](#)

Create a database session  
Log on to the database instance  
Create a table and query the error code  
Use a [Loader command \[Page 42\]](#) to load data into the table and query the error code  
Log off

```
# Reference to the SAP DB Perl Library
# -----

use SAP::DBTECH::loader;
```

```

# Parse the call arguments
# -----

$userid = $ARGV[0];
$password = $ARGV[1];
$database_name = $ARGV[2];
$data_path = $ARGV[3];
$server_node = "localhost";

# Create a database session for the Loader
# -----

$session = loader::Loader ($server_node, $database_name);

# Log on to the database instance
# -----

$session->cmd("use user $userid $password;");

$rc = $session->sql('EXISTS TABLE CUSTOMER')

If $rc!=0

# Create the table CUSTOMER
# -----

$session->cmd ( 'CREATE TABLE customer ( '.
                'cno          FIXED(4), '.
                'name         CHAR(10) ASCII, '.
                'zip          CHAR(5)  ASCII, '.
                'city         CHAR(12) ASCII, '.
                'PRIMARY KEY (cno) ')

print $rc

If $rc==0

# Load the table CUSTOMER
# -----

$loadrc = $session->cmd ("DATALOAD TABLE customer ".
                        "cno      1-4".
                        "name    6-12".
                        "zip     14-18".
                        "city    20-31".
                        "INFILE $data_path\customer.dat" )

print $loadrc

# End the database session
# -----

undef $session

```



## Perl: Example 5

Example of [Calling the Loader with Perl Script \[Page 21\]](#)



If a command is changed at runtime, for example, because user inputs are part of the command, syntax errors can easily occur.

The SQL method only provides SQL error codes. You need exceptions to intercept syntax errors in the [DATALOAD command \[Page 47\]](#).

Create a database session  
Log on to the database instance

Use a Loader [command \[Page 42\]](#) to load data into the table and intercept exceptions  
Log off

An incorrect data path in the following example can lead to a syntax error or data access error.

```
# Reference to the SAP DB Perl Library
# -----

use SAP::DBTECH::loader;

# Parse the call arguments
# -----

$userid = $ARGV[0];
$password = $ARGV[1];
$database_name = $ARGV[2];
$data_path = $ARGV[3];
$server_node = "localhost";

# Create a database session for the Loader
# -----

$session = loader::Loader ($server_node, $database_name);

# Log on to the database instance
# -----

$session->cmd("use user $userid $password;");

# Example of Exception Handling
# -----

eval{
    $loadrc=$session->sql ("DATALOAD TABLE customer ".
        "cno      1-4".
        "name    6-12".
        "zip      14-18".
        "city     20-31".
        "INFILE $data_path\customer.dat" );
    print "$loadrc\n";}
if ($@){
    print "command failed: $@\n";
}

# End the database session
# -----

undef $session
```



## Call with the Python Interface of the SAP DB Loader

You can [call the SAP DB Loader \[Page 13\]](#) through the processing of a Python script.

### Prerequisites

The following Python modules are delivered with the SAP DB Loader program to allow you to create Python scripts that can be processed by the [Loader](#).

- Microsoft Windows 2000: loader.pyd
- UNIX: loadermodule.so

- HP: loadermodule.sl



These are the modules you need to use the SAP DB Loader functions. You do not need a complete PYTHON installation.

If you already have a PYTHON installation, add %INSTROOT%\misc to the PYTHONPATH variable.

The SAP DB Loader supports PYTHON as of version 1.5.2.

## Call

**x\_python** <python\_script\_file> [<argument>...]

You can optionally specify arguments <argument>.

### Frequently used arguments:

<user_name> <password>	User name Password
<database_name>	Name of database instance
<data_path>	Directory in which the data file is stored
<server_node>	Name of the server on which the Loader is located



**X\_python sample samplename secret TST C:\data\sapdb**

The Python script **sample** is called. The logon to the Loader is performed for the user **samplename** on the database instance **TST**. The data file is in the directory **C:\data\sapdb**.

## Template for Background Files

The following section contains examples of background files in Python. Their equivalents in the [Perl \[Page 21\]](#) script language are also provided for comparison.

Example 1 Build a Python module with reference to the SAP DB Python Libraries, parse the call arguments	<a href="#">Python [Page 30]</a>	<a href="#">Perl [Page 23]</a>
Example 2 Create a database session with the Loader Log on to the database instance Log off	<a href="#">Python [Page 31]</a>	<a href="#">Perl [Page 24]</a>
Example 3 Set up a database session Log on to the database instance Query whether a table exists by querying the error code Create a table without querying the error code Log off	<a href="#">Python [Page 31]</a>	<a href="#">Perl [Page 24]</a>
Example 4 Create a database session Log on to the database instance Create a table and query the error code Use <a href="#">command(s) [Page 42]</a> to load data into the table and query the	<a href="#">Python [Page 31]</a>	<a href="#">Perl [Page 24]</a>

error code Log off	<a href="#">32]</a>	<a href="#">25]</a>
Example 5 Create a database session Log on to the database instance Use command(s) to load data into table and intercept exceptions Log off	<a href="#">Python [Page 33]</a>	<a href="#">Perl [Page 26]</a>

[Python Classes \[Page 29\]](#)

## Python Classes

Use the following classes to [Call the SAP DB Loader with Python Scripts \[Page 27\]](#):

- [Class Loader \[Page 29\]](#)
- [Exception Classes \[Page 30\]](#)

## Python: Loader Class

### Constructor: Loader (<server\_node>, <database\_name>)

<server_node>	Server node name
<database_name>	Name of database instance

Creation of a connection to the [Loader](#)

If the server node name and name of the database instance are specified, the system assumes that the database instance, data, and Loader are located on a remote server.

If only the name of the database instance is specified, the system assumes that the database instance, data, and Loader are located on the local server. The suitable Loader is determined from the release of the specified database instance.

If neither the server node name nor the name of the database instance is specified, the system establishes a connection to the newest Loader version on the local system.

The session is closed again when the object is deleted with `del session`.



```
session = loader.Loader ('p12345', 'mydb')
```

### Method: cmd (<command\_string>)

An [SQL statement \[Page 13\]](#) or Loader [command \[Page 42\]](#) is executed.

The script is terminated if the command fails.



```
output = session.cmd ("""DATALOAD TABLE customer
                        cno          1-4
                        name         6-12
                        zip           14-18
                        city          20-31
                        INFILE '%s\customer.dat' """
                        %data_path)
```

## Method: sql (<command\_string>)

An SQL statement or a Loader command is executed.

If a command fails, execution of the script is terminated.

If an SQL statement fails, a return code is output.



```
result = session.sql ('EXISTS TABLE MYTABLE')
```

## Python: Exception Classes

### Class: CommunicationError

The creation of the connection to the SAP DB Loader failed.

The value of the exception is an instance with the following attributes:

- errorCode
- message

### Class: LoaderServError

The command failed.

The value of the exception is an instance with the following attributes:

- errorCode (Loader error number)
- message (Loader error text)
- sqlCode (SQL error number if an SQL command fails, otherwise 0)
- sqlMessage (SQL error text if an SQL command fails, otherwise 0)

## Python: Example 1

Example of [calling the Loader with Python Script \[Page 27\]](#)

Build a Python module with reference to the SAP DB Python Libraries  
Parse the call arguments

```
# Reference to the Python Libraries
# -----

import sys
import loader

# Parse the call arguments
# -----

userid = sys.argv [1]
password = sys.argv [2]
database_name = sys.argv [3]
data_path = sys.argv[4]
server_node = ''
```



## Python: Example 2

Example of [calling the Loader with Python Script \[Page 27\]](#)

Create a database session with the SAP DB Loader  
 Log on to the database instance  
 Log off

```
# Reference to the Python Libraries
# -----

import sys
import loader

# Parse the call arguments
# -----

userid = sys.argv [1]
password = sys.argv [2]
database_name = sys.argv [3]
data_path = sys.argv[4]
server_node = ''

# Create a database session for the Loader
# -----

session = loader.Loader (server_node, database_name)

# Log on to the database instance
# -----

session.cmd ('use user %s %s;' % (userid, password))

# Log off
# -----

del session
```



## Python: Example 3

Example of [calling the Loader with Python Script \[Page 27\]](#)

Create a database session  
 Log on to the database instance  
 Query whether table exists by querying the error code  
 Create a table without querying error code  
 Log off

```
# Reference to the Python Libraries
# -----

import sys
import loader

# Parse the call arguments
# -----

userid = sys.argv [1]
password = sys.argv [2]
database_name = sys.argv [3]
data_path = sys.argv[4]
server_node = ''
```



```

session = loader.Loader (server_node, database_name)
# Log on to the database instance
# -----
session.cmd ('use user %s %s;' % (userid, password))
# Query whether the table exists by querying the error code
# The sql method is used to do this
# -----
rc = session.sql("EXISTS TABLE CUSTOMER")
If rc!=0
    # Then branch of the If statement must be indented in Python
    # Create the table CUSTOMER
    # -----
    rc = session.sql( """CREATE TABLE customer (
                          cno                  FIXED(4),
                          name                  CHAR(10) ASCII,
                          zip                   CHAR(5)  ASCII,
                          city                  CHAR(12) ASCII,
                          PRIMARY KEY (cno) """)
print rc
If rc==0
    # Then branch of the If statement must be indented in Python
    # Load the table CUSTOMER
    # -----
    loadrc = session.sql ("""DATALOAD TABLE customer
                          cno                  1-4
                          name                  6-12
                          zip                   14-18
                          city                  20-31
                          INFILE %s\customer.dat""" %data_path )
    print loadrc
session.cmd ("COMMIT")
# End the database session
# -----
del session

```



## Python: Example 5

Example of [calling the Loader with Python Script \[Page 27\]](#)



If a command is changed at runtime, for example, because user inputs are part of the command, syntax errors can easily occur.

The SQL method only provides SQL error codes. You need exceptions to intercept syntax errors or other Loader errors in the [commands \[Page 42\]](#).

Create a database session

Log on to the database instance

Use Loader command(s) to load data into table and intercept exceptions

Log off

```

# Reference to the Python Libraries
# -----

import sys
import loader

# Parse the call arguments
# -----

userid = sys.argv [1]
password = sys.argv [2]
database_name = sys.argv [3]
data_path = sys.argv[4]
server_node = ''

# Create a database session for the Loader
# -----

session = loader.Loader (server_node, database_name)

# Log on to the database instance
# -----

session.cmd ('use user %s %s;' % (userid, password))

# Example of Exception Handling
# -----

try:

    loadrc = session.sql ("""DATALOAD TABLE customer
                                cno          1-4
                                name         6-12
                                zip          14-18
                                city         20-31
                                INFILE %s\customer.dat""" %data_path )

    print "'%s'" % loadrc

except loader.LoaderServerError, err:
print 'command failed:', err

# End the database session
# -----

del session

```



## Call with the Java Interface of the SAP DB Loader

You can [call the SAP DB Loader \[Page 13\]](#) using Java.

### Prerequisites

Java is installed on the server.

The following Java modules are delivered with the SAP DB Loader program:

`com.sap.dbtech.powertoys.Loader`

`com.sap.dbtech.powertoys.LoaderException`

For a description of the modules, see [Java Classes \[Page 35\]](#).



## Java Classes

Use the following classes to [Call the SAP DB Loader with Java \[Page 34\]](#):

- [Loader Class \[Page 35\]](#)
- [Exception Classes \[Page 36\]](#)



## Java: Loader Class

The following Java module is delivered with the SAP DB Loader program:

```
com.sap.dbtech.powertoys.Loader
```

```
public class Loader extends java.lang.Object
```

With the Loader class, you can create new [Loader](#) objects instances and execute [SQL statements \[Page 13\]](#) or Loader [commands \[Page 42\]](#).

### Constructor: Loader

```
public Loader(java.util.Properties properties) throws RTEException
```

A new Loader object instance is created. The Loader class uses the following properties:

- `host`: The Loader is started on this host.
- `dbname`: The Loader is started for the specified database instance.
- `dbroot`: The Loader is started for this version/in this directory.

### Method: cmd

```
public java.lang.String cmd(java.lang.String cmdString) throws  
RTEException, LoaderException
```

An SQL statement or a Loader command (`cmdString`) is executed.

### Method: dbLoader

```
public static Loader dbloader(java.lang.String host, java.lang.String  
dbname) throws RTEException
```

A new Loader object instance is created in which the host name (`host`) and the name of the database instance (`dbname`) are specified.

### Method: dbrootLoader

```
public static Loader dbrootloader(java.lang.String host,  
java.lang.String dbroot) throws RTEException
```

A new Loader object instance is created in which the host name (`host`) and the `dbroot` directory are specified.

### Method: finalize

```
public void finalize() throws RTEException
```

This method overwrites the method `finalize()` of the class `java.lang.Object`.

### Method: release

```
public void release() throws RTEException
```

The Loader session is ended.



## Java: Exception Classes

The following Java module is delivered with the SAP DB Loader program:

```
com.sap.dbtech.powertoys.LoaderException
```

```
public class LoaderException extends java.lang.
```

If a Loader command fails, the [Loader](#) exception is output.

### Method: `getErrorCode`

```
public int getErrorCode()
```

The error code set by the Loader is returned.

### Method: `getErrorID`

```
public java.lang.String getErrorID()
```

The error number set by the Loader is returned.

### Method: `toString`

```
public java.lang.String toString()
```

The exception is output as a string object.



## Syntax Rules for Calling the SAP DB Loader

Not the following general syntax rules when [calling the SAP DB Loader \[Page 13\]](#).

### Upper- and Lowercase Characters

A distinction is not made between upper- and lowercase characters. The SAP DB Loader converts all names to uppercase characters internally, unless you place character strings in double quotation marks.

### Keywords

Both the [keywords \[Page 103\]](#) of the SAP DB Loader and [SQL keywords](#) can be used to denote database objects if they are placed in single or double quotation marks.



## SAP DB Loader: Log Functions

The SAP DB Loader logs its actions in a log file and in system tables.

- [Log File \[Page 37\]](#)
- [System Tables \[Page 37\]](#)

Unloading or loading a database takes some time, depending on the volume of data. The SAP DB Loader can restart the action if it is interrupted. To do this, the information in the system tables is evaluated.



## Log File

During the start of the SAP DB Loader, a log file is created or updated. The name of this log file is `loader.prt`. The file is stored in the [run directory](#) (`...\wrk`) of the database instance.

The following information is logged in the log file:

- Database name and user name
- Start and end of a load or unload action
- All executed SQL statements
- All executed load commands
  - For each load command, the position of the rejected input records in the medium and the cause of error
  - For each load command, the accumulated counts (inserted/modified lines and rejected lines)
- All executed unload commands
  - For each DATAEXTRACT command, the accumulated counts (number of extracted lines, and the number of lines extracted with errors)
- A message if the action was terminated

The log file is structured in such a way that parts of it can be easily used to create a [command file](#) [Page 20].

All comments begin in a new record with `// * or // <letter>` and are ignored by the SAP DB Loader.

Special comment lines:

<code>// M</code>	Line with messages These are informative system messages, such as the number of lines inserted and rejected, last completed transaction, and so on.
<code>// E</code>	Line with error messages This is used to introduce the two lines with the table and position values as well as the error number or error text for rejected lines.
<code>// R</code>	Line with REMARK comment These are comments that the user specified using the REMARK command.



## System Tables

During the installation of the database system, the following system tables are created for the SAP DB Loader:

- [TRANSFORMATIONMODEL](#) [Page 38]
- [TRANSFORMATIONMONITOR](#) [Page 39]
- [TRANSFORMATIONPACKAGES](#) [Page 40]
- [TRANSFORMATIONRESOURCES](#) [Page 40]
- [TRANSFORMATIONSTATISTICS](#) [Page 41]
- [TRANSFORMATIONSTEPS](#) [Page 41]

If you perform a load or unload process that spans multiple tables, the system tables are filled with the corresponding information.

The SAP DB Loader logs information about the runtime behavior for load and unload processes in the system tables.

You can also configure a transformation using the system tables.

The SAP DB Loader uses the entries in the system tables to be able to restart commands. You can use this ability to restart with the RESTART option ([restart \[Page 90\]](#)) of the command.



## TRANSFORMATIONMODEL

The SAP DB Loader [system table \[Page 37\]](#) TRANSFORMATIONMODEL contains all entries about [transformations \[Page 10\]](#) that span more than one table. After a successful transformation, these entries are unloaded to a [data stream \[Page 108\]](#) or deleted.

### TRANSFORMATIONMODEL

PACKAGEGUID	CHAR BYTE(24)	Unique ID of the transformation package
TASKID	FIXED(6,0)	ID of the transformation task
SERVERNODE	CHAR(32)	Server node on which the database instance to be transformed is installed
SERVERDB	CHAR(32)	Name of the database instance that is to be transformed
CODETYPE	CHAR(32)	Code type of the database instance (ASCII   UNICODE)
USERNAME	CHAR(32)	User name
CODEPAGE	CHAR(32)	ID of the ASCII code page that is to be used
STORAGEDEVICE	CHAR(32)	External storage medium that is to be used
TRANSFORMATION	CHAR(32)	Transformation type (EXTRACT   LOAD   UPDATE)
PART	CHAR(32)	Parts of the database catalog (ALL   USER   TABLE)
STEPID	FIXED(6,0)	ID of the transformation step
OWNER	CHAR(32)	Owner of the table
TABLENAME	CHAR(32)	Table name
TABLETYPE	CHAR(8)	Table type (TABLE   VIEW   SYNONYM)
CATALOGSTREAMNAME	CHAR(254)	ID of the CATALOG data stream
CATALOGSTREAMTYPE	CHAR(32)	Medium of the CATALOG data stream (FILE   PIPE)
CATALOGSTREAMFORMAT	CHAR(32)	Format of the CATALOG data stream (DDL   XML)
DATASTREAMNAME	CHAR(254)	ID of the DATA data stream
DATASTREAMTYPE	CHAR(32)	Medium of the DATA data stream (FILE   PIPE)
DATASTREAMFORMAT	CHAR(32)	Format of the DATA data stream (PAGES   RECORDS   COMPRESSED   FORMATTED)

DATASTREAMHEADERBLOCK	FIXED(6,0)	Start position of the header of the DATA data stream (in 2 GB blocks)
DATASTREAMHEADERPOS	FIXED(10,0)	Start position of the header of the DATA data stream (in bytes, less than 2 GB)
DATASTREAMDATABLOCK	FIXED(6,0)	Start position of the data of the DATA data stream (in 2 GB blocks)
DATASTREAMDATAPOS	FIXED(10,0)	Start position of the data of the DATA data stream (in bytes, less than 2 GB)
DATASTREAMTRAILBLOCK	FIXED(6,0)	Start position of the trailer of the DATA data stream (in 2 GB blocks)
DATASTREAMTRAILERPOS	FIXED(10,0)	Start position of the trailer of the DATA data stream (in bytes, less than 2 GB)
RESTART	BOOLEAN	Transformation step was executed (TRUE) or not (FALSE)
EXCLUDE	BOOLEAN	Transformation step was excluded (TRUE) or not (FALSE)
RESTORE	BOOLEAN	Transformation command was generated (TRUE) or not (FALSE)
PAGECOUNT	FIXED(10,0)	Number of pages that were transformed during a transformation step
ROWCOUNT	FIXED(10,0)	Number of rows that were transformed during a transformation step
STARTTIME	TIMESTAMP	Start of the transformation step
ENDTIME	TIMESTAMP	End of the transformation step
REQUESTTIME	FIXED(15,10)	Total execution time of a kernel request within a transformation step
RECEIVETIME	FIXED(15,10)	Total execution time of a kernel reply within a transformation step
IOTIME	FIXED(15,10)	Total I/O time within a transformation step
SEQ_NO	FIXED(6,0)	Sequence number with which split GRANT statements can be assigned to each other
CMD	CHAR ASCII (254)	Authorization statement
TRANSACTIONSIZE	FIXED(6,0)	Minimum number of rows that are to be transformed before the status of the transformation step is logged



## TRANSFORMATIONMONITOR

The SAP DB Loader [system table \[Page 37\]](#) TRANSFORMATIONMONITOR describes the resource usage of a [transformation \[Page 10\]](#).

### TRANSFORMATIONMONITOR

PACKAGEGUID	CHAR BYTE(24)	Unique ID of the transformation package
-------------	---------------	---

TABLECOUNT	FIXED(10,0)	Number of transformed table rows
ROWCOUNT	FLOAT(38)	Number of transformed rows
SYSTEMDURATION	FLOAT(38)	System performance
KERNELDURATION	FLOAT(38)	Kernel performance
LOADERDURATION	FLOAT(38)	Loader performance
DATADURATION	FLOAT(38)	Data Writer performance
LOGDURATION	FLOAT(38)	Log Writer performance
IODURATION	FLOAT(38)	I/O performance



## TRANSFORMATIONPACKAGES

The SAP DB Loader [system table \[Page 37\]](#) TRANSFORMATIONPACKAGES contains all steps of a [transformation \[Page 10\]](#) for a transformation package.

### TRANSFORMATIONPACKAGES

PACKAGEGUID	CHAR BYTE(24)	Unique ID of the transformation package
TASKID	FIXED(6,0)	ID of the transformation task
STEPID	FIXED(6,0)	ID of the transformation step
OWNER	CHAR(32)	Owner of the table
TABLERNAME	CHAR(32)	Table name
EXCLUDE	BOOLEAN	Transformation step was excluded (TRUE) or not (FALSE)
RESTART	BOOLEAN	Transformation step was executed (TRUE) or not (FALSE)



## TRANSFORMATIONRESOURCES

The SAP DB Loader [system table \[Page 37\]](#) TRANSFORMATIONRESOURCES describes the data sources and data rows involved in a [transformation \[Page 10\]](#).

### TRANSFORMATIONRESOURCES

Column name	Data Type	Explanation
PACKAGEGUID	CHAR BYTE (24)	Unique ID of the transformation package
SERVERNODE	CHAR(32)	Server node on which the database instance to be transformed is installed
SERVERDB	CHAR(32)	Name of the database instance that is to be transformed
USERNAME	CHAR(32)	User name

CODEPAGE	CHAR(32)	ID of the ASCII code page that is to be used
STORAGEDEVICE	CHAR(32)	External storage medium that is to be used
CATALOGSTREAMNAME	CHAR(254)	ID of the CATALOG data stream
CATALOGSTREAMTYPE	CHAR(32)	Medium of the CATALOG data stream (FILE   PIPE)
DATASTREAMNAME	CHAR(254)	ID of the DATA data stream
DATASTREAMTYPE	CHAR(32)	Medium of the DATA data stream (FILE   PIPE)



## TRANSFORMATIONSTATISTICS

The SAP DB Loader [system table \[Page 37\]](#) TRANSFORMATIONSTATISTICS describes the resource usage of a step of a [transformation \[Page 10\]](#).

### TRANSFORMATIONSTATISTICS

PACKAGEGUID	CHAR BYTE (24)	Unique ID of the transformation package
STEPID	FIXED(6.0)	Ascending number for each transformation package
OWNER	CHAR ASCII (32)	Owner of the table
TABLERNAME	CHAR ASCII (32)	Table name
BYTECOUNT	FIXED(11.0)	Size of the DATA data stream in bytes
ROWCOUNT	FIXED(10.0)	Number of table rows in the DATA data stream
SYSTEMTIME	FLOAT(38)	Transformation performance
KERNELTIME	FIXED(20.10)	Kernel performance
LOADERTIME	FLOAT(38)	Loader performance
DATATIME	FLOAT(38)	Pager performance
LOGTIME	FIXED(19.10)	Log Writer performance
IOTIME	FIXED(19.10)	I/O performance



## TRANSFORMATIONSTEPS

THE SAP DB Loader [system table \[Page 37\]](#) TRANSFORMATIONSTEPS describes a [transformation \[Page 10\]](#) at table level.

### TRANSFORMATIONSTEPS

PACKAGEGUID	CHAR BYTE(24)	Unique ID of the transformation package
STEPID	FIXED(6,0)	ID of the transformation step
TRANSFORMATION	FIXED(6,0)	Transformation type (EXTRACT   LOAD   UPDATE)
OWNER	CHAR(32)	Owner of the table
TABLENAME	CHAR(32)	Table name
CATALOGSTREAMNAME	CHAR(254)	ID of the CATALOG data stream
DATASTREAMNAME	CHAR(254)	ID of the DATA data stream
DATASTREAMHEADERBLOCK	FIXED(6,0)	Start position of the header of the DATA data stream (in 2 blocks)
DATASTREAMHEADERPOS	FIXED(10,0)	Start position of the header of the DATA data stream (in blocks less than 2 GB)
DATASTREAMDATABLOCK	FIXED(6,0)	Start position of the data of the DATA data stream (in 2 GB blocks)
DATASTREAMDATAPOS	FIXED(10,0)	Start position of the data of the DATA data stream (in bytes less than 2 GB)
RESTART	BOOLEAN	Transformation step was executed (TRUE) or not (FALSE)
START	FIXED(10,0)	Number of transformed rows



## Commands

Each SAP DB Loader command comprises one or more keywords, arguments, and options.

### Command Overview

[AUTOCOMMIT \[Page 43\]](#)  
[CATALOGEXTRACT \[Page 43\]](#)  
[CATALOGLOAD \[Page 44\]](#)  
[DATAEXTRACT \[Page 45\]](#)  
[DATALOAD \[Page 47\]](#)  
[DATAUPDATE \[Page 48\]](#)  
[DBEXTRACT \[Page 48\]](#)  
[DBLOAD \[Page 50\]](#)  
[FASTLOAD \[Page 51\]](#)  
[SET \[Page 53\]](#)  
[SQLMODE \[Page 53\]](#)  
[TABLEEXTRACT \[Page 54\]](#)  
[TABLELOAD \[Page 55\]](#)  
[USE SERVERDB \[Page 58\]](#)  
[USE USER \[Page 58\]](#)

### Syntax Elements

[Syntax Rules for Commands \[Page 58\]](#)  
[Keywords \[Page 103\]](#)

## AUTOCOMMIT Command

You use this [command \[Page 42\]](#) for setting up a [database session](#) to activate/deactivate AUTOCOMMIT mode for a database session.

### Syntax

```
<autocommit_statement> ::= = AUTOCOMMIT ON | AUTOCOMMIT OFF
```

### AUTOCOMMIT ON

The AUTOCOMMIT mode is activated (SAP DB Loader default value).

With commands for loading data, a [COMMIT](#) is used to complete the insertion of a certain number of data records. The user can determine this number using the [SET TRANSACTION SIZE \[Page 53\]](#) command. This mode is not relevant when data is unloaded, because the table contents in the database are not modified.

If you execute COMMIT and ROLLBACK statements when AUTOCOMMIT mode is activated, the SAP DB Loader ignores these statements.

Each [SQL statement \[Page 13\]](#) is terminated implicitly with a COMMIT by the SAP DB Loader.

### AUTOCOMMIT OFF

AUTOCOMMIT mode is deactivated.

All transactions must be terminated explicitly using a COMMIT.

You can group commands in units so that they can be completed or reset at once.

This mode does not apply to the commands [FASTLOAD \[Page 51\]](#), [TABLEEXTRACT \[Page 54\]](#), and [TABLELOAD \[Page 55\]](#), as these commands run outside the transaction concept of the database and are therefore always implicitly terminated with a COMMIT by the SAP DB Loader.



Large transactions (such as a large number of data records to be loaded) require a correspondingly large [log area](#).

## CATALOGEXTRACT Command

You can use this SAP DB Loader [command \[Page 42\]](#) to control the unloading of the entire [database catalog](#) or parts of it.

### Syntax

```
<catalogextract_statement> ::= CATALOGEXTRACT <part_spec [Page 89]>  
<ostream_spec [Page 89]>
```

You can also use the keywords EXTRACT CATALOG instead of the [keyword \[Page 104\]](#) CATALOGEXTRACT.

### Prerequisites

THE command CATALOGEXTRACT ALL can only be executed by the [SYSDBA](#) user.

## Use

### <part\_spec>

- CATALOGEXTRACT ALL  
The [data stream \[Page 108\]](#) <outstream\_spec> contains all commands for creating a database session and SQL statements for data definition that are required for the definition of the entire database catalog.
- CATALOGEXTRACT USER  
The data stream <outstream\_spec> contains all commands for creating a database session and SQL statements for data definition that are required for the definition of the parts of the database catalog that belong to the specified user.
- CATALOGEXTRACT TABLE  
The specified table is a base table. The user is the [owner](#) of this table.  
The data stream <outstream\_spec> contains all SQL statements for data definition that are required for the definition of the table the database catalog objects connected to it.

### Data Stream: <outstream\_spec>

During unloading, the data stream <outstream\_spec> is created for the database catalog definitions.

The system table [TRANSFORMATIONMODEL \[Page 38\]](#) and other system tables are filled.

## Result

The desired parts of the database catalog are unloaded to the data stream <outstream\_spec>.

You can use the corresponding [CATALOGLOAD command \[Page 44\]](#) to restore these database catalog objects.

## Errors

The unload process cannot be successfully completed. In this case, the SAP DB Loader logs the cause of the error in the [log file \[Page 37\]](#).

The values PACKAGEGUID and CLIENTNODE are both written to the log file.



## CATALOGLOAD Command

You can use this SAP DB Loader [command \[Page 42\]](#) to control the loading of the entire [database catalog](#) or parts of it.

## Syntax

```
<catalogload_statement> ::= CATALOGLOAD <part_spec [Page 89]>  
<instream_spec [Page 77]>
```

You can also use the keywords LOAD CATALOG instead of the [keyword \[Page 104\]](#) CATALOGLOAD.

## Prerequisite

A data stream was created with the [CATALOGEXTRACT command \[Page 43\]](#).

THE command CATALOGLOAD ALL can only be executed by the [SYSDBA](#) user.

## Use

### <part\_spec>

- CATALOGLOAD ALL  
The entire [database catalog](#) is restored from the [data stream \[Page 108\]](#) <instream\_spec> using the commands for creating a database session and the SQL statements for data definition.
- CATALOGLOAD USER  
The user that executes the CATALOGLOAD USER command must have the same user name as the [owner](#) of the database catalog objects in the data stream <instream\_spec>. The database catalog objects in the data stream are restored using the commands for creating a database session and the SQL statements for data definition.
- CATALOGLOAD TABLE  
The user that executes the CATALOGLOAD TABLE command must have the same user name as the owner of the database catalog objects in the data stream <instream\_spec>. The database catalog objects in the data stream are restored using the commands for creating a database session and the SQL statements for data definition.

### Data Stream: <instream\_spec>

The [data stream \[Page 108\]](#) <instream\_spec> is required when loading.

If you want to use the data stream generated with the CATALOGEXTRACT command, specify this name.

The system table [TRANSFORMATIONMODEL \[Page 38\]](#) and other system tables are filled.

## Result

The desired parts of the database catalog are loaded from the data stream <instream\_spec> to the database catalog.

## Errors

The load process cannot be successfully completed. In this case, the SAP DB Loader logs the cause of the error in the [log file \[Page 37\]](#).

The values PACKAGEGUID and CLIENTNODE are both written to the log file.



## DATAEXTRACT Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the unloading of [application data \[Page 108\]](#) from database tables.

## Syntax

```
<dataextract_statement> ::=
DATAEXTRACT <select_expression> [Page 92] [OUTFIELDS] [<output_column_list> [Page
88]] <extract_files_spec> [Page 73]
| DATAEXTRACT <restore_spec> [Page 90] TABLE <table_name> [Page 100]
[<order_clause> [Page 87]] <extract_files_spec> [Page 73]
```

You can also use the keywords EXTRACT DATA instead of the [keyword \[Page 104\]](#) DATAEXTRACT.

## Prerequisite

You must have access authorization for the tables from which you want to unload data.

You must be owner of the table to use the second variant of the DATAEXTRACT command.

## Use

The SAP DB Loader offers the following variants of the DATAEXTRACT command.

### Variant 1: DATAEXTRACT [OUTFIELDS]

You can do the following with this variant of the command:

- Define the columns of the table that is to be unloaded
- Link multiple tables that you want to unload
- Specify a sort sequence for the extracted data



```
DATAEXTRACT cno, name, zip, address from customer
OUTFILE 'customer.data'
```



```
DATAEXTRACT * from article
OUTFIELDS anr          01-08
           bez          09-39
           bestand     40-43 INTEGER
           min_best    44-45 INTEGER
           bestellt    46-49 INTEGER
           lieferdatum 50-57
           preis       58-65 DECIMAL (2)
           gewicht     66-69 REAL
OUTFILE 'article.data' FORMATTED
```

### Variant 2: DATAEXTRACT ... TABLE

You can only use this variant of the command to unload **complete single tables**. You unload all application data of a table in one [data stream \[Page 108\]](#) and the table definition in the database catalog in a second data stream. You can use these two data streams to recreate the unloaded table and load all application data.



```
DATAEXTRACT FOR DATALOAD TABLE article
OUTFILE 'article.command'
OUTFILE 'article.data'
```

Data stream for the definition from the database catalog: `article.command`

Data stream for application data: `article.data`

## Process Flow

The SAP DB Loader uses the information from the DATAEXTRACT command to generate an internal SELECT command and constructs a results table. The data in this results table is then unloaded into the target data stream, as instructed by the command.

All tables from which data is being unloaded are write-protected while the DATAEXTRACT command is being executed. This means that other users cannot make changes to this table while it is being unloaded.

## Result

The data is unloaded from the tables to be unloaded into the target data stream(s).

You can use a [DATALOAD \[Page 47\]](#) or [FASTLOAD \[Page 51\]](#) command to reload the table data back into the database instance.

## Errors

The unload process cannot be successfully completed. In this case, the SAP DB Loader logs the number of successfully unloaded data records in the [log file \[Page 37\]](#).

## DATALOAD Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the loading of [application data \[Page 108\]](#).

## Syntax

```
<dataload_statement> ::=
DATALOAD TABLE <table_spec [Page 101]> [<duplicates_clause [Page 72]>]
<load_column_spec mlt [Page 79]> <instream_spec [Page 77]> [<longfile_spec mlt [Page 81]>]
```

You can also use the keywords LOAD DATA instead of the [keyword \[Page 104\]](#) DATALOAD.



```
DATALOAD TABLE sqltravel00.customer
  cno          01-04
  name         06-16
  zip          17-22
  address      23-48
INFILE 'customer.data' FORMATTED
```



```
DATALOAD TABLE edemo.products
  productid    01-10 CHAR
  productname  11-51 CHAR
  unitinstock  52-55 INTEGER
  unitsonorder 56-59 INTEGER
  unitprice    60-67 DECIMAL (2)
INFILE 'products.data' FORMATTED
```

[Additional Examples \[Page 12\]](#)

## Use

Like the [FASTLOAD command \[Page 51\]](#), the DATALOAD command reads data from a [data stream \[Page 108\]](#) and writes it to the tables of a database instance.

DATALOAD commands are processed while the database is running.



A DATALOAD process last longer in total than a FASTLOAD process. The writing of the log entries, in particular, can lead to a loss of performance. If errors occur, you can undo DATALOAD entries, as the log entries can be evaluated.

## Prerequisites

The table exists in the database instance. Otherwise, it must be created before the DATALOAD command is executed.

## Process Flow

The SAP DB Loader generates an internal mass INSERT statement from the DATALOAD command and then executes it.

During the load operation, all of the tables modified by this DATALOAD command can also be read and changed by other users.

## Result

The data in the data stream has been loaded to the target table(s). All of the changes made to the target table(s) as a result have also been written to the log area of the database instance.

## Errors

If the load operation cannot be ended successfully, the SAP DB Loader logs the last row that was inserted successfully in the table, the number of rows inserted, and the number of rows rejected in the [log file \[Page 37\]](#).



## DATAUPDATE Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the updating of individual column values of tables.

## Syntax

```
<dataupdate_statement> ::=  
DATAUPDATE TABLE <table\_spec \[Page 101\]> <acc\_column\_spec\_mlt \[Page 62\]>  
<set\_column\_spec\_mlt \[Page 94\]> <instream\_spec \[Page 77\]> [longfile\_spec\_mlt \[Page 81\]>]
```

You can also use the keywords UPDATE DATA instead of the [keyword \[Page 104\]](#) DATAUPDATE.

## Prerequisites

The target table exists on the database instance.

## Process Flow

The SAP DB Loader generates an internal UPDATE statement from the DATAUPDATE command and then executes it.

## Result

The individual line values in the specified table columns are updated. All of the changes made to the target table(s) as a result have also been written to the [log area](#) of the database instance.

## Errors

If errors occur while a DATAUPDATE command is being executed, the SAP DB Loader terminates the process and displays an error message. A [log file \[Page 37\]](#) is written.



## DBEXTRACT Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the unloading of [application data \[Page 108\]](#) and the database catalog.

## Syntax

```
<dbextract_statement> ::=
DBEXTRACT [<configurationstream_spec [Page 70]>] <catalogostream_spec [Page 63]>
<dataostream_spec [Page 71]> [<packagestream_spec [Page 89]>] [<restart [Page
90]>]
```

You can also use the keywords EXTRACT DB instead of the [keyword \[Page 104\]](#) DATAEXTRACT.

## Prerequisite

THE DBEXTRACT command can only be executed by the [SYSDBA](#) user.

## Use

The DBEXTRACT command combines the commands [CATALOGEXTRACT \[Page 43\]](#) ALL and [TABLEEXTRACT \[Page 54\]](#) ALL.

### Data Streams: <catalogostream\_spec> and <dataostream\_spec>

At least two [data streams \[Page 108\]](#) must be created when unloading:

- Data stream for the database catalog definitions <catalogostream\_spec>
- Data stream for the application data <dataostream\_spec>

The system table [TRANSFORMATIONMODEL \[Page 38\]](#) and other system tables are filled.

### Exclude Tables from Unload: <configurationstream\_spec>

- Unless you specify the data stream <configurationstream\_spec>, the entire database is unloaded.
- By specifying a data stream <configurationstream\_spec>, you can exclude the application data of individual selected tables from the unload process. The tables must be specified in the data stream by the specification of their owner and the table name. The information about which tables were not unloaded is stored in the system tables. You can display these tables by displaying the system table [TRANSFORMATIONPACKAGES \[Page 40\]](#). The tables excluded from the unload of the application data have the value TRUE in the column EXCLUDE.

The database catalog entries for all tables are always unloaded, irrespective of possible restrictions of the data stream.

### Unloading System Tables: <packagestream\_spec>

- The system tables are not unloaded unless you specify <packagestream\_spec>.
- By specifying a data stream <packagestream\_spec>, all entries of the table TRANSFORMATIONMODEL are unloaded to this data stream.

If possible, you should not unload the system tables.

## Result

The application data and the database catalog are unloaded to the data streams <dataostream\_spec> and <catalogostream\_spec>.

If necessary, all entries of the table TRANSFORMATIONMODEL are unloaded in the data stream <packagestream\_spec>.

You can use the corresponding [DBLOAD command \[Page 50\]](#) to restore the database catalog and the application data.

## Errors

The unload process cannot be successfully completed. In this case, the SAP DB Loader logs the cause of the error in the [log file \[Page 37\]](#).

The values PACKAGEGUID and CLIENTNODE are both written to the log file. The PACKAGEGUID is required if you want to restart the unload process.

## DBLOAD Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the loading of [application data \[Page 108\]](#) and the database catalog.

## Syntax

```
<dbload_statement> ::=
DBLOAD [configurationstream\_spec \[Page 70\]] <cataloginstream_spec [Page 62]>
<datainstream_spec [Page 71]> [packagestream\_spec \[Page 89\]] [restart \[Page 90\]]
```

You can also use the keywords LOAD DB instead of the [keyword \[Page 104\]](#) LOAD DB.

## Prerequisite

THE DBLOAD command can only be executed by the [SYSDBA](#) user.

You have unloaded the database catalog and its content with the [DBEXTRACT command \[Page 48\]](#).

## Use

The DBLOAD command combines the commands [CATALOGLOAD \[Page 44\]](#) ALL and [TABLELOAD \[Page 55\]](#) ALL.

### Data Streams: <cataloginstream\_spec> and <datainstream\_spec>

At least two [data streams \[Page 108\]](#) are required when loading:

- Data stream for the database catalog definitions <cataloginstream\_spec>
- Data stream for the application data <datainstream\_spec>

If you want to use the data streams generated with the DBEXTRACT command, specify these names.

The system table [TRANSFORMATIONMODEL \[Page 38\]](#) and other system tables are filled.

### Exclude Tables from Load: <configurationstream\_spec>

- Unless you specify the data stream <configurationstream\_spec>, all of the data contained in the data streams <cataloginstream\_spec> and <datainstream\_spec> are loaded.
- By specifying a data stream <configurationstream\_spec>, you can exclude the application data of individual selected tables from the load process. The tables must be specified in the data stream by the specification of their owner and the table name. The information about which tables were not loaded is stored in the system tables. You can display these tables by displaying the system table [TRANSFORMATIONPACKAGES \[Page 40\]](#). The tables excluded from the loading of the application data have the value TRUE in the column EXCLUDE.

The database catalog entries for the tables contained in <cataloginstream\_spec> are always loaded, irrespective of possible restrictions of the data stream.

Differentiate between the following variants when making the required specifications in the data stream `<configurationstream_spec>`, depending on the [medium \[Page 122\]](#) from which you are loading the data streams:

- Variant 1: Loading from a file (**FILE**)  
The data stream must contain entries for the positions of table data that is to be excluded from the load process. You can find this information in the specifications for the data stream `<packagestream_spec>` of the corresponding DBEXTRACT command.
- Variant 2: Loading using a pipe (**PIPE**)  
The tables specified in the data stream are excluded from the transformation.

### Unloading System Tables: `<packagestream_spec>`

- The system tables are not unloaded unless you specify `<packagestream_spec>`.
- By specifying a data stream `<packagestream_spec>`, all entries of the table TRANSFORMATIONMODEL are unloaded to this data stream.

If possible, you should not unload the system tables.

### Result

The application data and the database catalog definitions are loaded using the data streams `<datastream_spec>` and `<cataloginstream_spec>`.

If necessary, all entries of the table TRANSFORMATIONMODEL are unloaded in the data stream `<packagestream_spec>`.

### Errors

The load process cannot be successfully completed. In this case, the SAP DB Loader logs the cause of the error in the [log file \[Page 37\]](#).

The values PACKAGEGUID and CLIENTNODE are both written to the log file. The PACKAGEGUID is required if you want to restart the unload process.



## FASTLOAD Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the loading of [application data \[Page 108\]](#) for the specified tables.

### Syntax

```
<fastload_statement> ::=
FASTLOAD [usage\_spec \[Page 102\]] TABLE table\_spec \[Page 101\]
[load\_column\_spec mlt \[Page 79\]] instream\_spec \[Page 77\]
```



```
FASTLOAD with 100 % USAGE
TABLE sqltravel100.customer
  cno          1
  name         2
  zip          3
  address      4
INFILE 'customer.data'
```

### Use

FASTLOAD commands are processed while the database is running.



When application data is loaded with FASTLOAD, it is entered directly in the [pages](#) on the database instance. This command loads data more quickly than the [DATALOAD command \[Page 47\]](#). However, because it does not write any log entries, you must back up the new data after you have loaded it. You can either back up the corresponding pages ([incremental data backup](#)) or all data ([complete data backup](#)). The table is write-protected until you have backed up the new application data.

## Prerequisites

- The target table exists on the database instance. Otherwise, the table can be created using a corresponding [SQL statement \[Page 13\]](#) before the FASTLOAD commands is executed.
- You have logged onto the SAP DB Loader with the user who is the owner of the target table.
- The data in the [data stream \[Page 108\]](#) to be imported is sorted in ascending order in the sequence of the primary key.
- The target table does not have an index.
- If the target table already contains application data, only those data records whose key values are greater than the largest key value in the target table can be inserted.
- The data that you want to load does not contain LONG columns.

## Process Flow

When a FASTLOAD command is started, the target table of the load operation in the database instance is locked so other users cannot write to it. The target table can still be read, however.

Unrestricted reading and writing is possible in all other tables.

Once a FASTLOAD command has been processed, other write operations triggered with FASTLOAD from the same user can be performed on this table. Once the load operation with FASTLOAD has been completed, all users only have read access for the table.

A backup of the database instance must be created before this table is released again for write operations from other users.

## Result

The data from the source table has been loaded to the target table.



The successful load of the application data is implicitly completed by the SAP DB Loader with a [COMMIT](#).

## Errors

If a FASTLOAD load operation is aborted, all of the lines transferred during the operation are deleted. Application data that existed in the table before the FASTLOAD load operation was started remains unchanged.

If you cancel the load operation because the SAP DB Loader refused to transfer certain lines, you can check the [log file \[Page 37\]](#) to find the data record that caused the operation to fail.

## SET Command

This SAP DB Loader [command \[Page 42\]](#) to create a [database session](#) allows you to define many properties of a database session.

### Syntax

```
<set_statement> ::=
  SET <bool_spec [Page 62]>
| SET CODETYPE <standard_code_spec [Page 97]>
| SET <code_page_spec [Page 63]>
| SET COMPRESSED '/<s>/<d>/'
| SET DATE <standard_date_mask [Page 97]>
| SET LANGUAGE 'ENG|DEU'
| SET MAXERRORCOUNT <valMAXERRORCOUNT>
| SET <null_spec [Page 85]>
| SET <number_spec [Page 86]>
| SET TIME <standard_time_mask [Page 98]>
| SET TIMESTAMP <standard_timestamp_mask [Page 99]>
| SET TRANSACTION SIZE <valTRANSACTION_SIZE>
```

COMPRESSED '/<s>/<d>/'	<p>Characters used in the <a href="#">COMPRESSED [Page 109]</a> format to separate or select data; current value can be changed for individual commands</p> <p>&lt;s&gt; Character used to separate data fields, default value: comma, value must be exactly one character long, and cannot be blank</p> <p>&lt;d&gt; Character used to select data, default value: double quotation marks, value can be exactly one character long or blank, if it is blank, no delimiter is defined</p>
LANGUAGE 'ENG DEU'	<p>Language in which the messages of the database instance and the SAP DB Loader appear</p> <p><b>ENG:</b> English, <b>DEU:</b> German</p>
MAXERRORCOUNT <valMAXERRORCOUNT>	<p>Definition of how many errors the SAP DB Loader accepts during the processing of an individual command before processing of the command is terminated</p> <p>This value applies during a complete session or until it is overwritten by another SET MAXERRORCOUNT command</p>
TRANSACTION SIZE <valTRANSACTION_SIZE>	<p>Defines the number of data records after which a <a href="#">COMMIT</a> is performed, when inserting data</p>

### Use

You can adjust predefined values of the control parameters of the SAP DB Loader to your requirements.



The new values are only valid for the current database session and must be restored if they are needed in subsequent sessions.

## SQLMODE Command

You can use this SAP DB Loader [command \[Page 42\]](#) to specify, during the creation of a [database session](#), the [SQL Mode](#) of the database instance.

## Syntax

<sql\_mode\_statement> ::= SQLMODE <sql\_mode [Page 96]>

## Use

The commands for loading and unloading executed by the SAP DB Loader and the [SQL statements \[Page 13\]](#) are interpreted in accordance with the defined SQL mode.

The defined SQL mode remains active until it is changed again with this command. If you enter an invalid SQL mode, the SAP DB Loader uses the internal SQL mode of the database instance.



## TABLEEXTRACT Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the unloading of database tables.

## Syntax

```
<tableextract_statement> ::=
TABLEEXTRACT <part_spec [Page 89]> [<configurationstream_spec [Page 70]>]
<dataoutstream_spec [Page 71]> [<packagestream_spec [Page 89]>] [<restart [Page 90]>]
```

You can also use the keywords EXTRACT DATA instead of the [keyword \[Page 106\]](#) DATAEXTRACT.

## Prerequisites

When you use TABLEEXTRACT TABLE, the source table must exist in the database instance.

THE command TABLEEXTRACT ALL can only be executed by the [SYSDBA](#) user.

## Use

- You can unload the [application data \[Page 108\]](#) (including LONG values) of the specified database tables in [data streams \[Page 108\]](#). The [format of the data stream \[Page 109\]](#) must be [PAGES \[Page 112\]](#) in each case.
- You can unload the application data application data (including LONG values) and the database catalog definitions of the specified database tables to data streams. The format of the data stream must be [RECORDS \[Page 112\]](#) in each case.

You cannot change the data streams.

Tables are write-protected while they are unloaded to the data stream.

### <part\_spec>

- TABLEEXTRACT ALL  
The data stream <dataoutstream\_spec> contains the application data of all tables.
- TABLEEXTRACT USER  
The data stream <dataoutstream\_spec> contains the application data of the tables that belong to the specified user.
- TABLEEXTRACT TABLE  
The specified table is a base table. The user is the [owner](#) of this table.  
The data stream <dataoutstream\_spec> contains the application data of the specified table.

## Data Stream: <dataostream\_spec>

The data streams <dataostream\_spec> are created during unloading.

When it processes the commands **TABLEEXTRACT ALL** and **TABLEEXTRACT USER**, the SAP DB Loader creates one or more data streams (depending on the size of the tables it is unloading), each of which can contain the application data from one or more tables. The SAP DB Loader specifies the maximum size of a single data stream as 1 GB.

The media names for the data streams are created using the name specified in the command, in that a four digit extension of the format 0001 is added to the name. The maximum number of [media \[Page 122\]](#) for each TABLEEXTRACT is therefore set at 9999.

The system table [TRANSFORMATIONMODEL \[Page 38\]](#) and other system tables are filled.

## Exclude Tables from Unload: <configurationstream\_spec>

- Unless you specify the data stream <configurationstream\_spec>, all specified tables are unloaded.
- By specifying a data stream <configurationstream\_spec>, you can exclude the application data of individual selected tables from the unload process. The tables must be specified in the data stream by the specification of their owner and the table name. The information about which tables were not unloaded is stored in the system tables. You can display these tables by displaying the system table [TRANSFORMATIONPACKAGES \[Page 40\]](#). The tables excluded from the unload of the application data have the value TRUE in the column EXCLUDE.

## Unload System Tables: <packagestream\_spec>

- The system tables are not unloaded unless you specify <packagestream\_spec>.
- By specifying a data stream <packagestream\_spec>, all entries of the table TRANSFORMATIONMODEL are unloaded to this data stream.

If possible, you should not unload the system tables.

## Result

The specified tables are unloaded to the data streams <dataostream\_spec>.

If necessary, all entries of the table TRANSFORMATIONMODEL are unloaded in the data stream <packagestream\_spec>.

You can reload the tables using the corresponding [TABLELOAD command \[Page 55\]](#).

## Errors

The unload process cannot be successfully completed. In this case, the SAP DB Loader logs the cause of the error in the [log file \[Page 37\]](#).

The values PACKAGEGUID and CLIENTNODE are both written to the log file. The PACKAGEGUID is required if you want to restart the unload process.



## TABLELOAD Command

With this SAP DB Loader [command \[Page 42\]](#), you can control the loading of database tables.

## Syntax

```
<tableload_statement> ::=
TABLELOAD <part_spec [Page 89]> [<configurationstream_spec [Page 70]>]
```

<[datastream\\_spec \[Page 71\]](#)> [[packagestream\\_spec \[Page 89\]](#)>] [[restart \[Page 90\]](#)>]

You can also use the keywords LOAD DATA instead of the [keyword \[Page 106\]](#) DATALOAD.

## Prerequisites

THE command TABLELOAD ALL can only be executed by the [SYSDBA](#) user.

You have unloaded the database tables with the [TABLEEXTRACT command \[Page 54\]](#).

You cannot change the format of the generated [data streams \[Page 108\]](#) ([PAGES \[Page 112\]](#) or [RECORDS \[Page 112\]](#)). If the data streams have the format RECORDS, you must create the required database table definitions in the database catalog using the corresponding data definition statements.

The target tables must be empty. If the tables are not empty, the SAP DB Loader attempts to delete the contents of the tables.

## Use

### <part\_spec>

- TABLELOAD ALL  
The data streams <datastream\_spec> contain the [application data \[Page 108\]](#) of all tables.
- TABLELOAD USER  
The data streams <datastream\_spec> contain the application data of the tables that belong to the current user.
- TABLELOAD TABLE  
The specified table is a base table. The user is the [owner](#) of this table.  
The data stream <datastream\_spec> contains the application data of the specified table.

## Data Streams <datastream\_spec>

You can recover database tables using the data streams created with the TABLEEXTRACT command. If you want to use the data streams generated with the TABLEEXTRACT command, specify this name as <datastream\_spec>.

The system table [TRANSFORMATIONMODEL \[Page 38\]](#) and other system tables are filled.

## Process Flow

The SAP DB Loader restores the table in the following order:

1. The database catalog information is adjusted (case 1) or newly created (case 2).
2. The application data is loaded (including LONGs).
3. Indexes are created if they exist in the source table.

No log entries are written during a load process with TABLELOAD.

### Case 1:

#### The table to be restored exists in the database instance

The SAP DB Loader can adjust the following differences between the database catalog information for the source and target tables:

- Table name and column name of source table and target table  
The table name and column names of the source table are copied.
- Differently defined defaults  
The defaults defined in the source table are copied. The defaults defined in the target table are deleted.

- Differently defined indexes  
The indexes defined in the source table are copied. The indexes defined in the target table are deleted.
- Differently defined constraints  
The constraints defined in the source table are copied. The constraints defined in the target table are deleted.

If there are differences between the source and target tables for the definitions of types and the length of the individual columns, the TABLELOAD process is terminated.

#### Case 2:

#### The table to be restored does not exist in the database instance

- If data streams have the format PAGES, the table is newly created using the existing database catalog information.
- If the data streams have the format RECORDS, you must create the required database table definitions in the database catalog using the corresponding data definition statements.

#### Exclude Tables from Load: <configurationstream\_spec>

- Unless you specify the data stream <configurationstream\_spec>, all data contained in the data streams <datastream\_spec> is loaded.
- By specifying a data stream <configurationstream\_spec>, you can exclude the application data of individual selected tables from the load process. The tables must be specified in the data stream by the specification of their owner and the table name. The information about which tables were not loaded is stored in the system tables. You can display these tables by displaying the system table [TRANSFORMATIONPACKAGES \[Page 40\]](#). The tables excluded from the loading of the application data have the value TRUE in the column EXCLUDE.

Differentiate between the following variants when making the required specifications in the data stream <configurationstream\_spec>, depending on the [medium \[Page 122\]](#) from which you are loading the data streams:

- Variant 1: Loading from a file (**FILE**)  
The data stream must contain entries for the positions of table data that is to be excluded from the load process. You can find this information in the specifications for the data stream <packagestream\_spec> of the corresponding TABLEEXTRACT command.
- Variant 2: Loading using a pipe (**PIPE**)  
The tables specified in the data stream are excluded from the load.

#### Unloading System Tables: <packagestream\_spec>

- The system tables are not unloaded unless you specify <packagestream\_spec>.
- By specifying a data stream <packagestream\_spec>, all entries of the table TRANSFORMATIONMODEL are unloaded to this data stream.

If possible, you should not unload the system tables.

## Result

The database tables are loaded using the data streams <datastream\_spec>.

If necessary, all entries of the table TRANSFORMATIONMODEL are unloaded in the data stream <packagestream\_spec>.

The table is write-protected during the restore process and after a successful restore. To remove the write-protection, back up the database instance after you have restored the table.

## Errors

The load process cannot be successfully completed. In this case, the SAP DB Loader logs the cause of the error in the [log file \[Page 37\]](#).

The values PACKAGEGUID and CLIENTNODE are both written to the log file. The PACKAGEGUID is required if you want to restart the unload process.



## USE SERVERDB Command

You can use this [command \[Page 42\]](#) for creating a [database session](#) to create a connection to another [database instance](#).

### Syntax

```
<use_serverdb_statement> ::= USE <database name statement \[Page 71\]>
```

### Prerequisites

A user with the same name and password must exist on the new database instance.

### Use

You switch to another database instance without having to leave the SAP DB Loader. It terminates the connection to the current database instance and sets up the new connection to the database instance.

All subsequent commands are referred to this database instance by the SAP DB Loader.



```
USE SERVERDB MK1 ON PCnew;
```



## USE USER Command

You can use this [command \[Page 42\]](#) for creating a [database session](#) to create a connection with the specified information to a [database instance](#).

### Syntax

```
<use_user_statement> ::= USE <user statement \[Page 103\]>
```

### Use

This command terminates the current connection of the SAP DB Loader to the database instance and creates a new connection with the specified parameters.

If you do not enter a database instance name under <user\_statement>, the new connection is created to the database instance that was used prior to the USE USER command.



## Syntax Rules for Commands

The syntax used for SAP DB Loader [commands \[Page 42\]](#) is the Backus Naur Form (BNF) with the following conventions:

<Rule name> ::= <Rule section>

The terms in angle brackets represent syntactical units. Each rule name must be resolved by a rule section.

A rule section is any combination of rules and endsymbols. Endsymbols are numbers, literals, or [keywords \[Page 103\]](#) and are not resolved further.

Rule sections set in angle brackets indicate an optional user input.

The keywords used in the commands are written in uppercase for the sake of clarity. They can be specified by the user in upper- or lowercase letters.

Blank characters are allowed in commands. They are not interpreted by the SAP DB Loader.

Endsymbols that start with `val` stand for values that must be specified by the user.

Remember that values for '`valLITERAL`' (as specified in the syntax rules) must be placed in single quotation marks.

If you use keywords as names of columns, tables or users, you must place them in double quotation marks.

- [Syntax Rules for Setting up a Session \[Page 59\]](#)
- [Syntax Rules for Table Descriptions \[Page 59\]](#)
- [Syntax Rules for Column Descriptions \[Page 60\]](#)
- [Syntax Rules for Describing the Data Stream \[Page 60\]](#)



## Syntax Rules for Setting up a Session

The [syntax rules \[Page 58\]](#) listed here are components of the commands for [setting up a database session \[Page 11\]](#). These use of these rules is explained in more detail.

You can go from the commands for setting up a database session directly to the descriptions of the syntax rules used in each case.

<a href="#">database_name_statement [Page 71]</a>	<a href="#">sql_mode [Page 96]</a>	<a href="#">user_statement [Page 103]</a>
---	------------------------------------	---



## Syntax Rules for Table Descriptions

The [syntax rules \[Page 58\]](#) listed here are components of the table description in the commands for loading and unloading [application data \[Page 108\]](#). The use of these syntax rules is explained in more detail.

You can go from the description of the commands for loading and unloading data directly to the description of the syntax rules used in each case.



If you want to use [keywords \[Page 103\]](#) as table or user names, you must place them in double quotation marks.

<a href="#">compare_operator [Page 69]</a> <a href="#">condition [Page 70]</a>	<a href="#">duplicates_clause [Page 72]</a>	<a href="#">field_format [Page 73]</a> <a href="#">field_pos [Page 74]</a>
<a href="#">if_condition [Page 76]</a>	<a href="#">restore_spec [Page 90]</a>	<a href="#">simple_condition [Page 95]</a>
<a href="#">table_name [Page 100]</a>	<a href="#">usage_spec [Page 102]</a>	

<a href="#">table_spec [Page 101]</a>		
---------------------------------------	--	--



## Syntax Rules for Column Descriptions

The [syntax rules \[Page 58\]](#) listed here are components of the column description in the commands for loading and unloading [application data \[Page 108\]](#). The use of these syntax rules is explained in more detail.

You can go from the description of the commands for loading and unloading data directly to the description of the syntax rules used in each case.



If you want to use [keywords \[Page 103\]](#) as column names, you must place them in double quotation marks.

<a href="#">acc_column_spec [Page 61]</a> <a href="#">acc_column_spec mlt [Page 62]</a>	<a href="#">column_assignment [Page 65]</a> <a href="#">column_descriptor [Page 66]</a> <a href="#">column_id [Page 67]</a> <a href="#">column_id_spec [Page 67]</a> <a href="#">column_names [Page 69]</a>	<a href="#">format_spec [Page 75]</a>
<a href="#">generate_spec [Page 76]</a>	<a href="#">key_column_spec [Page 78]</a>	<a href="#">lit_column_spec [Page 78]</a> <a href="#">load_column_spec [Page 79]</a> <a href="#">load_column_spec mlt [Page 79]</a>
<a href="#">null_assign [Page 83]</a> <a href="#">null_condition [Page 84]</a> <a href="#">numerical_functions [Page 86]</a>	<a href="#">order_clause [Page 87]</a> <a href="#">output_column [Page 87]</a> <a href="#">output_column_list [Page 88]</a>	<a href="#">round_or_trunc_spec [Page 91]</a>
<a href="#">scale_spec [Page 92]</a> <a href="#">select_expression [Page 92]</a> <a href="#">set_column_spec [Page 94]</a> <a href="#">set_column_spec mlt [Page 94]</a> <a href="#">sequence_number [Page 94]</a> <a href="#">simple_column_spec [Page 95]</a>		



## Syntax Rules for Describing the Data Stream

The [syntax rules \[Page 58\]](#) listed here are components of the description of the [data stream \[Page 108\]](#) in the load and unload commands. The use of these syntax rules is explained in more detail.

You can go from the commands for setting up a database session directly to the descriptions of the syntax rules used in each case.

<a href="#">bool_spec [Page 62]</a>	<a href="#">cataloginstream_spec [Page 62]</a> <a href="#">catalogostream_spec [Page 63]</a> <a href="#">code_page_spec [Page 63]</a> <a href="#">code_spec [Page 65]</a>	<a href="#">datainstream_spec [Page 71]</a> <a href="#">dataostream_spec [Page 71]</a> <a href="#">date_spec [Page 71]</a> <a href="#">delimiter_spec [Page 72]</a>
-------------------------------------	--	--

	<a href="#">configurationstream_spec [Page 70]</a>	
<a href="#">extract_files_spec [Page 73]</a>	<a href="#">instream_spec [Page 77]</a> <a href="#">int_spec [Page 77]</a>	<a href="#">longfile_code_spec [Page 80]</a> <a href="#">longfile_spec [Page 81]</a> <a href="#">longfile_spec_mlt [Page 81]</a>
<a href="#">mediumtype_spec [Page 82]</a>	<a href="#">noheader_spec [Page 82]</a> <a href="#">null_spec [Page 85]</a> <a href="#">number_spec [Page 86]</a>	<a href="#">outstream_spec [Page 89]</a>
<a href="#">packagestream_spec [Page 89]</a> <a href="#">part_spec [Page 89]</a>	<a href="#">restart [Page 90]</a>	<a href="#">separator_spec [Page 93]</a> <a href="#">standard_code_spec [Page 97]</a> <a href="#">standard_date_mask [Page 97]</a> <a href="#">standard_time_mask [Page 98]</a> <a href="#">standard_timestamp_mask [Page 99]</a> <a href="#">stream_extract [Page 99]</a> <a href="#">stream_format_spec [Page 99]</a>
<a href="#">time_spec [Page 101]</a> <a href="#">timestamp_spec [Page 102]</a>		



## acc\_column\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<acc_column_spec> ::= <key\_column\_spec \[Page 78\]> | <simple\_column\_spec \[Page 95\]>`

### Use

Use this syntax rule in a [DATAUPDATE command \[Page 48\]](#) to define whether the qualification columns are key columns or non-key columns of the target table.

If you use key columns as qualification columns ([key\\_column\\_spec \[Page 78\]](#)), each data record in the [data stream \[Page 108\]](#) updates that line in the target table that matches the specified key.



```
DATAUPDATE TABLE customer
KEY cno 01-04
SET city 05-16
INFILE 'customer.data' FORMATTED
```

If you use non-key columns as qualification columns ([simple\\_column\\_spec \[Page 95\]](#)), each data record in the source file updates each line in the target table that is identified with the value(s) of the qualification column(s).



```
DATAUPDATE TABLE customer
city 1
```

```
SET zip 2
INFILE 'customer.data'
```



## acc\_column\_spec\_mlt

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

```
<acc_column_spec_mlt> ::= <acc\_column\_spec \[Page 61\]>
<acc_column_spec_mlt>
```

### Use

Use this syntax rule in a [DATAUPDATE command \[Page 48\]](#) to define the qualification columns.

You can use both key columns and non-key columns as qualification columns.

The qualification columns must be listed before the target columns ([set\\_column\\_spec \[Page 94\]](#)) in a DATAUPDATE command.



## bool\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<bool_spec> ::= BOOLEAN <valVALUE_FOR_TRUE>/<valVALUE_FOR_FALSE>
```

valVALUE_FOR_TRUE	Defines the character string for values that are true
valVALUE_FOR_FALSE	Defines the character string for values that are false

The default value of the SAP DB Loader is **TRUE/FALSE**.

The character strings can contain a maximum of 10 characters.

### Use

Use this syntax rule to specify the character string that represents the BOOLEAN values in [data streams \[Page 108\]](#) to be unloaded from a database instance or to be loaded into a database instance.

You can also change the current value for individual commands.

#### See also:

[Data Format \[Page 109\]](#)



## cataloginstream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<cataloginstream_spec> ::= CATALOG <instream\_spec \[Page 77\]>
```



## catalogostream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<catalogostream_spec> ::= CATALOG <outstream\_spec \[Page 89\]>`



## code\_page\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<code_page_spec> ::= CODEPAGE [<valCODEPAGE_NAME>]`

<code>valCODEPAGE_NAME</code>	Name of code pages Default value: ISO-8859-1  If you specify an empty value, the code page is reset to the default value.
-------------------------------	--

### Use

If you convert CHAR data from ASCII to UCS2, you determine which code page is used (see also: [standard\\_code\\_spec \[Page 97\]](#)).

#### Case 1: The code page for the conversion is a user-defined code page and is not yet contained in the system table CODEPAGE.

To make user-defined code pages accessible to the SAP DB Loader for conversions, you first have to load these code pages into the system table [CODEPAGE \[Page 124\]](#). To do this, proceed as follows:

- In the directory `<dependent_path>/env`, create a file with the name `<valCODEPAGE_NAME>.txt`. This file has to be formatted like the files that are provided by the Unicode consortium at <ftp://ftp.unicode.org/Public/MAPPINGS/>.



File ISO-8859-8.txt

```
# Name:                ISO/IEC 8859-8:1999 to Unicode
# Unicode version:    3.0
# Table version:      1.1
# Table format:       Format A
# Date:               2000-Jan-03
# Authors:            Ken Whistler kenw@sybase.com
#
# Copyright (c) 1991-1999 Unicode, Inc. All Rights
reserved.
#
# This file is provided as-is by Unicode, Inc. (The Unicode
Consortium).
# No claims are made as to fitness for any particular
purpose. No
# warranties of any kind are expressed or implied. The
recipient
# agrees to determine applicability of information
```

```

provided. If this
# file has been provided on optical media by Unicode, Inc.,
the sole
# remedy for any claim will be exchange of defective media
within 90
# days of receipt.
#
# Unicode, Inc. hereby grants the right to freely use the
information
# supplied in this file in the creation of products
supporting the
# Unicode Standard, and to make copies of this file in any
form for
# internal or external distribution as long as this notice
remains
# attached.
...
# Format: Three tab-separated columns
#     Column #1 is the ISO/IEC 8859-8 code (in hex as 0xXX)
#     Column #2 is the Unicode (in hex as 0xFFFF)
#     Column #3 the Unicode name (follows a comment sign,
'#')
...

0x00 0x0000 # NULL
0x01 0x0001 # START OF HEADING
0x02 0x0002 # START OF TEXT
0x03 0x0003 # END OF TEXT
0x04 0x0004 # END OF TRANSMISSION
0x05 0x0005 # ENQUIRY
0x06 0x0006 # ACKNOWLEDGE
0x07 0x0007 # BELL
0x08 0x0008 # BACKSPACE
0x09 0x0009 # HORIZONTAL TABULATION
0x0A 0x000A # LINE FEED

```

5. Load the code page into the system table CODEPAGE.  
Use the [SET command \[Page 53\]](#) to do so. When processing the SET command **SET CODEPAGE <valCODEPAGE\_NAME>**, the SAP DB Loader first tries to find the specified code page in the system table CODEPAGE. If the table does not contain the code page, the SAP DB Loader tries to find the corresponding file <valCODEPAGE\_NAME>.txt. If this file exists, it is read, and the conversion table is entered in the system table CODEPAGE.

If the file is not found, and if the code page is not contained in the system table, the SAP DB Loader stops processing the SET command and displays an error message.

If the SET command is processed successfully, the user-defined code page is available in the system table CODEPAGE.

### **Case 2: The code page for the conversion is already contained in the system table CODEPAGE.**

## **Process Flow**

The SAP DB Loader reads the conversion table from the system table CODEPAGE. All subsequent conversions of the CHAR data from ASCII to UCS2 are performed by using this conversion table.

### **See also:**

[Data Format \[Page 109\]](#)

[Format of the Data Stream \[Page 109\]](#)

## code\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<code_spec> ::= <standard code spec \[Page 97\]> | CODESET <valCODESET_NAME>`

valCODESET_NAME	Code attribute
-----------------	----------------

### Use

You can define the default value for the interpretation of [data streams \[Page 109\]](#) that contain [plain text values \[Page 117\]](#).

**See also:**

[Data Format \[Page 109\]](#)

## column\_assignment

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<column_assignment> ::= <valCOLUMN_NAME> <'valLITERAL'>  
| <valCOLUMN_NAME> <generate\_spec \[Page 76\]>`

valCOLUMN_NAME	Column name Specify it according to the SQL conventions. It can also contain the table name as a prefix (such as, customer.cno).
'valLITERAL'	Constant, enclosed in single quotation marks

### Use

When you load application data, you can use this syntax rule to specify that a constant value is loaded into the specified column for each loaded data record, and which value this constant must have.

To do this, specify the constant in a position in the command. No data is then loaded from the [data stream \[Page 108\]](#) for the appropriate column.



If the data stream is empty, the constants specified in the command are not loaded.

### Loading Constants: <valCOLUMN\_NAME> <'valLITERAL'>

To load any constant, specify the constant in a position. Place the value in single quotation marks.

The SAP DB Loader handles the constant like a [plain text value \[Page 117\]](#) and converts it to the data type of the target column.

If you want to load the constant into a numeric column, it must have a valid numeric format.



```

DATALOAD TABLE article
ano          01-08
descr       09-39 CHAR
stock       40-43 INTEGER
min_ord     44-45 INTEGER
ordered     46-49 INTEGER
delivery_date 50-57 CHAR
price       58-65 DECIMAL(2)
weight      66-69 REAL
INFILE 'article.data' FORMATTED

```

### Loading Special Constants: <valCOLUMN\_NAME> <generate\_spec >

You can load the special values **STAMP**, **USER**, **USERGROUP**, **DATE**, **TIME**, **TIMESTAMP**, and **TRUE** and **FALSE**.



```

DATALOAD TABLE article
changed_by USER
changed_on DATE
changed_at TIME
ano 1
descr 2
stock 3 NULL IF POS 3 < '0'
INFILE 'customer.data'

```



## column\_descriptor

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

<column\_descriptor> ::= <valCOLUMN\_NAME> <field\_pos [Page 74]> <format\_spec [Page 75]>

valCOLUMN_NAME	Column name Specify it according to the SQL conventions. It can also contain the table name as a prefix (such as, customer.cno).
----------------	---

### Use

In a command to load application data, you can use this syntax rule to assign a data field of the data stream to a column of the target table. To do this, you specify the column name, the position, and the [external data type \[Page 114\]](#) of the data in the data stream.



```

DATALOAD TABLE article
ano          01-08 CHAR
descr       09-39 CHAR
stock       40-43 INTEGER
min_ord     44-45 INTEGER
ordered     46-49 INTEGER
del_date    50-57 CHAR
price       58-65 DECIMAL (2)
weight      66-69 REAL
INFILE 'article.data' FORMATTED

```



## column\_id

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

**<column\_id> ::= <valCOLUMN\_NAME> | <valCOLUMN\_ID>**

valCOLUMN_NAME	Column name Specify it according to the SQL conventions. It can also contain the table name as a prefix (such as, customer.cno).
valCOLUMN_ID	Column ID

### Use

In a command for unloading application data, use it to assign a column of the source table to a data field of the [data stream \[Page 108\]](#).



```

DATAEXTRACT cno, name, zip, place FROM customer
OUTFIELDS
  cno      01-04
  name     06-12
  zip      14-18
  place    20-31
OUTFILE 'customer.data' FORMATTED

```

The column number **valCOLUMN\_ID** indicates the position of the column in the SELECT statement.



```

DATAEXTRACT cno, name, zip, place FROM customer
OUTFIELDS
  1 01-04
  2 06-12
  3 14-18
  4 20-31
OUTFILE 'customer.data' FORMATTED

```



## column\_id\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

**<column\_id\_spec> ::= <column\_id [Page 67]> <field\_pos [Page 74]> <format\_spec [Page 75]> <null\_assign [Page 83]>**

### Use

Use it in unload commands to assign the data fields of the [data stream \[Page 108\]](#) to the columns of the source table. You specify the [external data types \[Page 114\]](#) and the conditions for unloading the application data.

### Note the following rules:

Decide whether you want your column descriptions to contain `field_pos` field descriptions.

However, you must specify position descriptions for **all or none** of the described columns. If you do not, the SAP DB Loader generates an error and terminates the command.

#### Format of the Data Stream: [COMPRESSED \[Page 109\]](#)

- Enter the position descriptions as relative positions only.

If you enter position descriptions with start and end positions for single columns or all columns, the SAP DB Loader generates an error and terminates the command.



```
DATAEXTRACT * from customer
OUTFIELDS
  cno      1
  name     2
  zip      3
  place    4-31  ERROR
OUTFILE 'customer.data' COMPRESSED
```

- Assign position 1 to the first column in your list. The position numbers of the following columns rise by 1 each time. This also means that each position number can only be assigned once.

If you do not keep to these rules, the SAP DB Loader generates an error and terminates the command.



```
DATAEXTRACT * from customer
OUTFIELDS
  cno      1
  name     2
  street   3
  zip      3  ERROR
  place    5  ERROR
OUTFILE 'customer.data' COMPRESSED
```

- You can assign a column to multiple **different** positions.



```
DATAEXTRACT * from customer
OUTFIELDS
  cno      1
  name     2
  zip      3
  place    4
  surname  5
OUTFILE 'customer.data' COMPRESSED
```

#### Format of the Data Stream: [FORMATTED \[Page 111\]](#)

- Specify all position descriptions with exact start and end positions.
- Specify only ascending, non-overlapping values for position descriptions. If the positions do not follow on from each other, the SAP DB Loader fills the gaps with **blank characters**. This also applies to binary data.



```
DATAEXTRACT * from customer
OUTFIELDS
```

```

cno          01-04
name         06-12
zip          14-18
place       16-31
OUTFILE 'customer.data' FORMATTED

```

- Define the positions for the data stream at least as long as the length of the values in the database.  
If you define a position for the data stream that is **longer** than the length of the value in the database, the following occurs:
  - Character strings are aligned left and space characters entered to make them the correct length
  - Numeric values are aligned right and space characters entered to make them the correct length
 If you define a position for the data stream that is **shorter** than the length of the values in the database, the Replication Manager generates an error and terminates the command.



## column\_names

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

```
<column_names> ::= <valCOLUMN_NAME> | <valCOLUMN_NAME>,<column_names>
```

valCOLUMN_NAME	Column name
----------------	-------------

### Use

Use this syntax rule in a [DATEXTRACT command \[Page 45\]](#) with `order_clause` (see variant 2 of the DATEXTRACT command) to specify the sort order of the columns that you want to unload.



```

DATEXTRACT FOR DATALOAD TABLE article ORDER BY delivery
date, price
OUTFILE 'article.command'
OUTFILE 'article.data'

```



## compare\_operator

[Syntax Rules for Table Descriptions \[Page 59\]](#)

### Syntax

```
<compare_operator> ::= < | > | = | <= | >= | !=
```

### Use

In the syntax rule [<simple condition> \[Page 95\]](#), use this syntax rule to specify the value with which a value in the data record is compared.



## condition

[Syntax Rules for Table Descriptions \[Page 59\]](#)

### Syntax

```
<condition> ::= <simple\_condition \[Page 95\]> | (<condition>) | <condition>
AND <condition>
| <condition> OR <condition> | NOT <condition>
```

### Use

The SAP DB Loader distinguishes between simple conditions and compound conditions. Simple conditions can be negated with **NOT**, combined with **AND** and **OR** to form compound conditions, or encapsulated as required.

Operators in parentheses are evaluated before those that are not in parentheses.

If no operators are in parentheses, the SAP DB Loader weights them as follows:

- **NOT** takes precedence over **AND** and **OR**
- **AND** takes precedence over **OR**
- If the weighting is identical, the operators are evaluated from left to right.

Only those records to which the simple or compound condition applies are loaded.

#### See also:

[Selecting Data Records \[Page 123\]](#)



You want to load those data records from the [data stream \[Page 108\]](#) `hotel.data` where the place is `BERLIN` and the price is less than `400.00` to the destination table `hotel`:

```
DATALOAD TABLE hotel
  IF POS 41-44 REAL < '400,00'
  AND
    POS 27-36 = 'BERLIN'
    hno      01-04 INTEGER
    name     09-18
    zip      20-25 DECIMAL
    place    27-36
    price    41-44 REAL
INFILE 'hotel.data' FORMATTED
```



## configurationstream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<configurationstream_spec> ::= CONFIGURATION <instream\_spec \[Page 77\]>
```

## database\_name\_statement

[Syntax Rules for Setting up a Session \[Page 59\]](#)

### Syntax

`<database_name_statement> ::= SERVERDB <valDBNAME> [ON <valDBNODE>]`

valDBNAME	Name of the <a href="#">database instance</a> , applies to the whole <a href="#">database session</a> with this database instance.  If the name of the database instance is identical to a <a href="#">keyword [Page 103]</a> or if the upper- and lowercase notation is relevant, the name must be placed in quotation marks.
valDBNODE	Name of the server on which the database instance that you want to open for a database session is installed.

### Use

Use it to open a database session for the database instance specified with **valDBNAME**.

If you also specify the name of the server **valDBNODE**, on which this database instance is installed, the database session for this database instance is opened on this server. Otherwise, the SAP DB Loader assumes that the specified database instance is on the server where the user is logged on to the SAP DB Loader.

## datainstream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<dataoutstream_spec> ::= DATA <instream\_spec \[Page 77\]>`

## dataoutstream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<dataoutstream_spec> ::= DATA <outstream\_spec \[Page 89\]>`

## date\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<date_spec> ::= DATE <standard\_date\_mask \[Page 97\]> | DATE  
'<valFREE_MASK>'`

valFREE_MASK	Output format  Specify which of the three possible components day, month, and year you want
--------------	---

	to appear in the date. You can use either the German abbreviations <b>T</b> , <b>M</b> , and <b>J</b> or the English abbreviations <b>D</b> , <b>M</b> , and <b>Y</b> . If you enter three characters for the month, that is <b>MMM</b> , the name of the month is displayed as the standard abbreviation, for example, Oct for October.
--	--

## Use

Use this syntax rule to specify the [data format \[Page 109\]](#) of the [plain text values \[Page 117\]](#) in which DATE columns are entered and output.

This format only applies to the load or unload command in which it is specified. Otherwise, the SAP DB Loader default is used.

Use the [SET command \[Page 53\]](#) to view the SAP DB Loader default.



## delimiter\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

```
<delimiter_spec> ::= DELIMITER '<valDELIMITER>'
```

<b>valDELIMITER</b>	DELIMITER display, No character, or exactly one character long Represent a blank DELIMITER by specifying of blank single quotation marks ('')
---------------------	---

## Use

Use it to specify which character should be used to select application data in [data streams \[Page 108\]](#) with the format [COMPRESSED \[Page 109\]](#). If you want to unselected data, enter a blank DELIMITER in the command.

Use the [SET command \[Page 53\]](#) to view the SAP DB Loader default.

### See also:

[Data Format \[Page 109\]](#)



## duplicates\_clause

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

```
<duplicates_clause> ::= IGNORE DUPLICATES | REJECT DUPLICATES |  
UPDATE DUPLICATES
```

<b>IGNORE DUPLICATES</b>	The new line is not inserted.
<b>REJECT DUPLICATES</b>	The new line is rejected with an error message.
<b>UPDATE DUPLICATES</b>	The new line overwrites the existing line.

## Use

Use this syntax rule in a [DATALOAD command \[Page 47\]](#) to specify how to proceed when loading data from a [data stream \[Page 108\]](#), and a line with the same key as the new line already exists in the target file.

If you do not make a specification, REJECT DUPLICATES is the default.



```
DATALOAD TABLE sqltravel00.customer
  UPDATE DUPLICATES
  customerid      01-04
  name            06-16
  zip             17-22
  address         23-48
INFILE 'customer.data' FORMATTED
```



## extract\_files\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

**<extract\_files\_spec>** ::= **<data\_outfile\_spec>** [**<longfile\_spec\_mlt>** [Page 81]]  
 | **<command\_outfile\_spec>** **<data\_outfile\_spec>** [**<longfile\_spec\_mlt>**]

<b>data_outfile_spec</b>	Definition of the data stream for <a href="#">application data [Page 108]</a>
<b>command_outfile_spec</b>	Definition of the data stream for load commands to be generated ( <a href="#">Command file [Page 20]</a> )
<b>longfile_spec_mlt</b>	Target data stream for data from table fields defined as LONG values

## Use

The first specification of this rule applies only to [DATAEXTRACT commands \[Page 45\]](#) that do not create command files (variant 1 of the DATEXTRACT command).



```
DATAEXTRACT * FROM article
OUTFILE 'article.data'
```

The second specification only applies for DATAEXTRACT commands that enable a complete recovery of the table (variant 2 of the DATAEXTRACT command).



```
DATAEXTRACT FOR DATALOAD TABLE article
OUTFILE 'article.command'
OUTFILE 'article.data'
```



## field\_format

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

`<field_format> ::= /* leer */ | CHAR | DECIMAL [<valFRACTION>]  
| ZONED [<valFRACTION>] | INTEGER | REAL`

valFRACTION	Number of decimal places for the <a href="#">external data types [Page 114]</a> DECIMAL and ZONED
-------------	---

## Use

Use this syntax rule to describe the external data type of the data fields in the [data streams \[Page 108\]](#).

You only need to specify the data type in a load or unload command or in a condition ([condition \[Page 70\]](#)), if the data field in question is to be read or output with a data type other than CHAR. Every internal database column format in the database can be read and output in CHAR format.

### See also:

[Converting Data Types \[Page 114\]](#)



## field\_pos

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

`<field_pos> ::= <valSTART_POS> | <valSTART_POS> - <valEND_POS>`

valSTART_POS	Start position of a data field
valEND_POS	End position of a data field

## Use

Use this syntax rule to describe:

- The position of the input field in the [data stream \[Page 108\]](#) when loading [application data \[Page 108\]](#)
- The position of an output field in the data stream when unloading application data
- The position of a comparison value for loading application data selectively ([simple condition \[Page 95\]](#)).

The position specification is dependent on the [format of the data stream \[Page 109\]](#).

## Format FORMATTED

If the data stream has the format [FORMATTED \[Page 111\]](#), enter only absolute start and end positions for the data fields in the data stream.



The data stream `customer.data` has the FORMATTED format. The data fields have a standard format and end with a line break.

<b>Position no.</b>	<b>1 2 3 4 5 6 7 8 9 . . . . .</b>
	0 1 m i l l e r - 1 0 2 7 7 N e w Y o r k - - -
	0 2 s m i t h - 6 0 6 7 8 C h i c a g o - - - - -
	0 3 k l e i n e r t 3 3 1 8 4 M i a m i - - - - -



## Syntax

`<format_spec> ::= <field_format [Page 73]> [HEX] [<numerical_functions [Page 86]>]`

## Use

In a command to load or unload application data, use this syntax rule to specify the data fields in the [data stream \[Page 108\]](#) ([external data type \[Page 114\]](#), position, and so on).

**HEX**: You can define [HEX values \[Page 117\]](#).



## generate\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

## Syntax

`<generate_spec> ::= USER | USERGROUP  
| STAMP | DATE | TIME | TIMESTAMP | <sequence_number> [Page 94]`

<b>USER</b>	Name of the current user is loaded
<b>USERGROUP</b>	Name of the group of the current user is loaded. If this user is not assigned to a user group, the name of the current user is loaded. The column into which you want to load one of these values must have the type CHAR (n) where n >= 32.
<b>DATE</b>	Current date The column into which you want to load one of these values must have the type DATE.
<b>TIME</b>	Current time The column into which you want to load one of these values must have the type TIME.
<b>TIMESTAMP</b>	Current timestamp The column into which you want to load one of these values must have the type TIMESTAMP.
<b>STAMP</b>	A unique value generated by SAP DB that can only be loaded into columns of the type CHAR (n) BYTE where n >= 8.

## Use

Use it to load the specified special constants into the database.



## if\_condition

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

`<if_condition> ::= IF <condition [Page 70]> | OTHERWISE`

## Use

Use this syntax rule to define the selection criterion that determines which records from the [data stream \[Page 108\]](#) are loaded to which target table.

You can selectively load data records that do not fulfill the selection criterion to a specific target table by using the `OTHERWISE` condition. This means that you can load all those records in the data stream, for which you cannot define a standard selection criterion, to a certain target table.

This syntax rule allows you to [select data records \[Page 123\]](#) when loading them from a data stream into a target table:



`OTHERWISE` can only be specified for the **last** table in a series of [DATALOAD commands \[Page 47\]](#).



## instream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<instream_spec> ::= INSTREAM [<mediumtype_spec [Page 82]>]
'<valSTREAM_NAME>' [<stream_format_spec [Page 99]>] [<stream_extract [Page 99]>] [<noheader_spec [Page 82]>]
```

valSTREAM_NAME	Name and path of the <a href="#">medium [Page 122]</a>
----------------	--

### Use

When loading data, use this syntax rule to describe the [data stream \[Page 108\]](#).

The rules `stream_extract` and `noheader_spec` are only evaluated for [FASTLOAD \[Page 51\]](#) and [DATALOAD commands \[Page 47\]](#). If you use these rules in other commands, they are ignored.

- You can use `stream_extract` to specify that only certain parts of a data stream are to be loaded.
- For data streams with [binary values \[Page 107\]](#), you can use `noheader_spec` to specify that the corresponding medium does not have a special header containing the sizes of a data record.



## int\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<int_spec> ::= INTEGER HILO | INTEGER LOHI
```

INTEGER HILO	The current <a href="#">data stream [Page 108]</a> stores integers so that the byte with the lowest valency is stored first; that is, is furthest right in the binary number (the big endian).
INTEGER LOHI	The current data stream stores integers so that the byte with the highest valency is stored first, that is, the furthest right in the binary number (little endian, byte swap).

## Use

Use it to define the representation of integers in data streams. You can only do this for data streams with the [FORMATTED BINARY \[Page 111\]](#) format ([binary values \[Page 107\]](#)). This rule is ignored when you load or unload in data streams that have the [COMPRESSED \[Page 109\]](#) format.

If the representation specified for a data stream does not match the current server, the values are adjusted before being loaded into the database or written to the data stream.

### See also:

[Data Format \[Page 109\]](#)



## key\_column\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

## Syntax

`<key_column_spec> ::= KEY <simple\_column\_spec \[Page 95\]>`

## Use

Use this rule to specify that a certain qualification column in the [DATAUPDATE command \[Page 48\]](#) is a key column of the target table.



## lit\_column\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

## Syntax

`<lit_column_spec> ::= '<valLITERAL>' <field\_pos \[Page 74\]>`

valLITERAL	Text constant
------------	---------------

## Use

Use this rule to define text constants for output in addition to the output values in the [DATAEXTRACT command \[Page 45\]](#).



```
DATAEXTRACT * from customer
OUTFIELDS
  'Customer number:' 01-15
  cno                 16-19
  'Name:'            20-25
  name                26-32
  zip                 34-38
  city                50-61
OUTFILE 'customer.data' FORMATTED
```

The text constant is specified in single quotation marks instead of the column name or column ID ([column\\_id \[Page 67\]](#)).

## load\_column\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

<load\_column\_spec> ::= <column\_descriptor> [\[Page 66\]](#) [<null\_condition> [\[Page 84\]](#)]  
| <column\_assignment> [\[Page 65\]](#)

### Use

Use this rule in a command for loading application data to assign the data fields in the [data stream \[Page 108\]](#) to the columns in the target table, and specify the [external data types \[Page 114\]](#) and conditions for loading the data.



```

DATALOAD TABLE article
  ano          01-08 CHAR
  descr       09-39 CHAR
  stock       40-43 INTEGER
  min_ord     44-45 INTEGER
  ordered     46-49 INTEGER
  del_date    50-57 CHAR
  price       58-65 DECIMAL (2)
  weight      66-69 REAL
INFILE 'article.data' FORMATTED

```

## load\_column\_spec\_mlt

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

<load\_column\_spec\_mlt> ::= <load\_column\_spec [\[Page 79\]](#)>  
<load\_column\_spec\_mlt>

### Use

Use this syntax rule in a command for loading application data to describe the data records you want to load from a [data stream \[Page 108\]](#) and their assignment to the columns of the target table.



Table definition:

```

CREATE TABLE sqltravel00.customer (cno FIXED(4,0), name
CHAR(10) NOT NULL, zip CHAR(5), address CHAR(25), PRIMARY
KEY (cno))

```

Load command:

```

DATALOAD TABLE customer
  cno          01-04
  name         06-16
  zip          17-22
  address      23-48
INFILE 'customer.data' FORMATTED

```

## Rules

Data must exist in the data stream for every column that you specify in the load command.

If you do not specify columns in the target table in the command, the entire column is populated with the default value defined for this column during the load operation. The NULL value is loaded if no specific default value is defined for the column.



```

DATALOAD TABLE customer
  cno      01-04
  name     06-16
INFILE 'customer.data' FORMATTED

```

zip and address are filled with the default value when the data is loaded

Key columns and mandatory columns (columns that are defined as NOT NULL without a default value) must be specified in the load command. Otherwise, the processing terminates with an SQL error.



```

DATALOAD TABLE customer
  name     06-16
  zip      17-22
  address  23-48
INFILE 'customer.data' FORMATTED

```

The command is terminated with an SQL error, as the column cno is missing.

If you do not specify any columns for the target table in the load command, the table is loaded as if all columns in the target table were specified in the command. If this is the case, data must exist in the data stream for all of the columns in the target table.



```

DATALOAD TABLE customer
INFILE 'customer.data' FORMATTED

```

Data must exist in the source file for all of the columns in the data stream.

## longfile\_code\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<longfile_code_spec> ::= <code_spec [Page 65]> | BINARY
```

### Use

With this specification, you define the default value for the interpretation of LONG [data streams \[Page 108\]](#).

Using this syntax rule, you can also transform LONG data between various code types when you load or unload the data.

If you load/unload ASCII LONG data into an ASCII database or UCS2 LONG data into a [UNICODE](#) database, the data does not have to be converted.

The following combinations are possible ([internal database data type \[Page 114\]](#) and [external data type \[Page 114\]](#)):

Internal database data type	Possible code attributes of LONG data stream during unloading
LONG ASCII	ASCII, UCS2, UTF8
LONG UNICODE	UCS2, UTF8
LONG BYTE	BINARY, ASCII (HEX), UCS2 (HEX), UTF8 (HEX)

Internal database data type	Possible code attributes of LONG data stream during loading
LONG ASCII	ASCII
LONG UNICODE	ASCII, UCS2, UTF8
LONG BYTE	BINARY, ASCII, UCS2 (HEX), UTF8 (HEX)

## longfile\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<longfile_spec> ::= LONGFILE <valCOLUMN_ID> <longfile_code_spec [Page 80]>
| LONGFILE <valCOLUMN_NAME> <longfile_code_spec>
| LONGFILE <valCOLUMN_ID> <valFILE_NAME> [<longfile_code_spec>]
| LONGFILE <valCOLUMN_NAME> <valFILE_NAME> [<longfile_code_spec>]
```

valCOLUMN_ID	Column ID
valCOLUMN_NAME	Column name
valFILE_NAME	Name and path of <a href="#">data stream [Page 108]</a>

### Use

If you use one of the first two syntax rules, `longfile_code_spec` is a required specification. In all other cases, this specification is optional.

- The first two syntax rules relate to loading/unloading in cases where each LONG value of a column is in a separate LONG data stream.
- The other rules relate to loading/unloading in cases where all LONG values of a column are in one LONG data stream.

#### See also:

[Loading LONG Values \[Page 118\]](#)

[Unloading LONG Values \[Page 115\]](#)

## longfile\_spec\_mlt

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<longfile_spec_mlt> ::= <longfile_spec [Page 81]> <longfile_spec_mlt>
```

## Use

You can use the [DATALOAD \[Page 47\]](#), [DATAEXTRACT \[Page 45\]](#), and [DATAUPDATE \[Page 48\]](#) commands to load LONG values to a target table, unload them from a [data stream \[Page 108\]](#), or modify the LONG values in a table. Use this rule to define the columns in a target table to which you want to load LONG values, or from which data source you want to unload them.

The data for LONG values is stored separately from the other data in LONG data stream. Specify these data streams with the syntax rule [longfile\\_spec \[Page 81\]](#).

If, when you are unloading LONG values, you specify more LONG data streams than there are LONG columns in the column list (see [output\\_column\\_list \[Page 88\]](#)), the SAP DB Loader ignores the surplus LONG data streams.

### See also:

[Loading LONG Values \[Page 118\]](#)

[Unloading LONG Values \[Page 115\]](#)



## mediumtype\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

**<mediumtype\_spec> ::= FILE | PIPE**

### Use

You can use this syntax rule for a [data stream \[Page 108\]](#) to assign a [medium \[Page 122\]](#).

- **FILE**: Medium for the data stream is the file system.
- **PIPE**: Medium for the data stream is a pipe.



## noheader\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

**<noheader\_spec> ::= NOHEADER <valRECORD\_LENGTH>**

<b>valRECORD_LENGTH</b>	Length of an individual data records in the <a href="#">data stream [Page 108]</a>
-------------------------	--

### Use

When loading [application data \[Page 108\]](#) with a [FASTLOAD \[Page 51\]](#) or [DATALOAD Command \[Page 47\]](#), you use this syntax rule to describe in [infile\\_spec](#) that a data stream with the [FORMATTED BINARY \[Page 111\]](#) format does not contain special headers. At the same time, you specify the length of an individual data record in the data stream.

This specification is optional.

If the data stream was generated with the [DATAEXTRACT command \[Page 45\]](#), it contains a special header with the length of an individual data record in the data stream.

If this header is missing, the SAP DB Loader can use `noheader_spec` to specify the length of an individual data record. If this rule is missing, the SAP DB Loader calculates the length of an individual data records using the information on the column positions in the load command.



## null\_assign

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

```
<null_assign> ::= [IF] NULL SET '<valLITERAL>'
```

valLITERAL	Null value representation
------------	---------------------------

### Explanation

Place the value for the null representation in single quotation marks. This is a [plain text value \[Page 117\]](#) that the SAP DB Loader represents as a character string or binary numeric value. This depends on the [external data type \[Page 114\]](#) of the column for which you want to generate the null value.

If you want to display the null value representation in one of the external data types for [binary values \[Page 107\]](#), it must have a valid numeral format. This means either a floating decimal number in mantissa/exponent representation or a fixed point number with the currently agreed or standard decimal setting.

The generated null value representation is written to the same place in the [data stream \[Page 108\]](#) as the actual column value.

- If the null value representation is shorter than the length of the value specified by the position, the value is filled with space characters. If the null value representation is longer, it is shortened to the specified length. The SAP DB Loader generates a warning.
- If columns in the target table permit NULL values and you have not specified a null value representation, the SAP DB Loader default is used for unloaded NULL values ([null\\_spec \[Page 85\]](#)). This has the external data type CHAR.
- If you define a null value representation for unloading NOT-NULL columns, it does not cause an error.

### Use

Use this syntax rule in a command for unloading application data to specify which value for the data field of the source table is written to the data stream if the value in the source table is a NULL value.



You want to unload the source table `article` from the database into the data stream `article.data`. Some of the columns in the table contain NULL values. Define a separate condition for each of these columns. If this condition is met, the corresponding null value representation is entered in the data stream.

```
DATAEXTRACT * FROM article
OUTFIELDS
  ano          01-08
  descr        09-39          IF NULL SET ' '
  stock        40-43 INTEGER  IF NULL SET '0'
  min_ord      44-45 INTEGER
  price        46-53 DECIMAL (2) IF NULL SET 'X'
```

```

weight      54-57 REAL
OUTFILE 'article.data' FORMATTED

```



You want to unload the source table `article` from the database instance into the data stream `article.data`. Some of the columns in the table contain NULL values.

Because there is no null value representation defined for the column in the command, but all values have the external data type CHAR, the SAP DB Loader default null value representation ' ? ' is used.

```

DATAEXTRACT * FROM article
OUTFIELDS
ano          01-08
descr       09-39
stock       40-51
min_ord     52-63
price       64-74
weight      75-85
OUTFILE 'article.data' FORMATTED

```



You want to unload the source table `article` from the database instance into the data stream `article.data`. Some of the columns in the table contain NULL values.

Because there is no null value representation defined for the column in the command, the SAP DB Loader attempts to use the null value representation defined as a default (data type CHAR). This causes an error if a numeric external data type has been defined for the columns. The command terminates with an error message.

```

DATAEXTRACT * FROM article
OUTFIELDS
ano          01-08
descr       09-39
stock       40-43 INTEGER
min_ord     44-45 INTEGER
price       46-53 DECIMAL (2)
weight      54-57 REAL
OUTFILE 'article.data' FORMATTED

```



## null\_condition

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

```
<null_condition> ::= NULL [IF] <condition [Page 70]> | DEFAULT NULL
```

### Explanation

When the condition is evaluated, the shorter of the comparison values (value in the [data stream \[Page 108\]](#) or the null value representation) is filled with blank characters.

A check is made before each line in the data stream is loaded to see whether the condition formulated for the columns applies. If it does, the NULL value is inserted in this column. If not, the value from the assigned field in the data stream is inserted.

## Use

Use this syntax rule in a command to load application data to specify the conditions under which the **NULL** value should be loaded into a column of the target table.

You can use the **DEFAULT-NULL** condition if the NULL value in the data stream is represented in the same way for **all** columns that you want to load from the target table. The character string that you need to specify after **NULL IF POS** is only specified once as a NULL specification ([null\\_spec \[Page 85\]](#)) in the **DEFAULT-NULL** condition.



You cannot load columns defined as key columns (KEY) or NOT NULL with the NULL value. If you do, the table load action terminates with an appropriate error message.

If, when you created the table in the database instance, you defined a default other than NULL for columns, you cannot load NULL values into the columns. If this is the case, the SAP DB Loader uses the default defined for the column instead of the NULL value.



You want to load the data from the data stream `article.data` into the database instance. You want to load the NULL value into some columns of the target table. Define a separate condition for each of these columns. The NULL value is entered if this condition is met.

```

DATALOAD TABLE article
  ano      01-08
  descr    09-39          NULL IF POS 09-11 = ' '
  stock    40-43 INTEGER  NULL IF POS 40-43 INTEGER < '0'
  min_ord  44-45 INTEGER
  price    46-53 DECIMAL (2) NULL IF POS 1 <> 'X'
                                     OR POS 46-53 DECIMAL < '0'
  weight   54-57 REAL
INFILE 'article.data' FORMATTED

```



You want to load the data from the data stream `article.data` into the database instance. You want to load the NULL value into some columns of the target table. The same ' ? ' representation of the NULL value applies to all columns in the command where **DEFAULT NULL** is specified.

```

DATALOAD TABLE article
  ano      01-08
  descr    09-39  DEFAULT NULL
  stock    40-43 INTEGER  DEFAULT NULL
  min_ord  44-45 INTEGER
  price    46-53 DECIMAL (2)  DEFAULT NULL
  weight   54-57 REAL
INFILE 'article.data' FORMATTED
NULL ' ? '

```



## null\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

```
<null_spec> ::= NULL '<valLITERAL>'
```

<b>valLITERAL</b>	Null value representation; maximum length of 20 characters The SAP DB Loader default value is: ? (a question mark and 19 blank characters).
-------------------	---

## Use

Use this syntax rule to specify the character string used to represent null values that were loaded from the database instance in [data streams \[Page 108\]](#).

You can also change the current value for individual commands.

### See also:

[Data Format \[Page 109\]](#)



## number\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

```
<number_spec> ::= DECIMAL '['<t>']/'<d>/'
```

<b>&lt;t&gt;</b>	Defines the character for structuring thousands The SAP DB Loader default value is no character
<b>&lt;d&gt;</b>	Defines the character used to separate integers from decimal places; The default value of the SAP DB Loader is a period

## Use

Use this syntax rule to specify which characters are to be used in decimal numbers to group thousands and separate integers from decimal places.

You can also change the current value for individual commands.

### See also:

[Data Format \[Page 109\]](#)



## numerical\_functions

[Syntax Rules for Column Descriptions \[Page 60\]](#)

## Syntax

```
<numerical_functions> ::= <scale\_spec \[Page 92\]>  
| <round\_or\_trunc\_spec \[Page 91\]>  
| <scale_spec> <round_or_trunc_spec>
```

## Use

Use this syntax rule to scale, round, or shorten numeric data values when you load data from the [data stream \[Page 108\]](#) to the target table or unload data from the source table to the data stream.



Note that you must always specify `scale_spec` before `round_or_trunc_spec`.



```

DATALOAD TABLE distance
...
  cm 7 SCALE 2
  cm 7 SCALE -3 ROUND 1
...
INFILE 'meter.data'

```



```

DATAEXTRACT * FROM distance
  cm 10-14 INTEGER SCALE 2
  m 14-17 INTEGER
  km 18-21 INTEGER SCALE -3 TRUNC 2

OUTFILE 'dimensions.bin' FORMATTED

```



## order\_clause

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<order_clause> ::= ORDER BY <column\_names \[Page 69\]>`

### Use

Use this syntax rule as a supplement to the DATAEXTRACT FOR DATALOAD command (variant 2 of the [DATAEXTRACT command \[Page 45\]](#)) to specify the sort order of the extracted data.



For a description of this SQL statement, see the *Reference Manual: SAP DB 7.4*, [ORDER clause \(order\\_clause\)](#) section.



## output\_column

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<output_column> ::= <column\_id\_spec \[Page 67\]> | <lit\_column\_spec \[Page 78\]>`

### Use

Use this rule in a command for unloading application data to assign the columns of the source table to the data fields in the [data stream \[Page 108\]](#), and to specify the [external data types \[Page 114\]](#) and conditions for unloading the data.



```
DATAEXTRACT * FROM article
OUTFIELDS
  ano          01-08 CHAR
  descr        09-39 CHAR
  stock        40-43 INTEGER
  min_ord      44-45 INTEGER
  ordered      46-49 INTEGER
  del_date     50-57 CHAR
  price        58-65 DECIMAL (2)
  weight       66-69 REAL
OUTFILE 'article.data' FORMATTED
```



## output\_column\_list

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

```
<output_column_list> ::= <output_column [Page 87]> | <output_column>
<output_column_list>
```

### Use

Use this syntax rule in a [DATAEXTRACT command \[Page 45\]](#) to describe the columns to be unloaded of a source table and the representation of the data records in the [data stream \[Page 108\]](#).

### Rules

- The columns can be in any order in the column description.
- The column list can only contain columns from the SELECT statement ([select expression \[Page 92\]](#)) or a subset of them.
- Values are output only for those columns in the column list. If the column list contains more columns than the SELECT statement, the SAP DB Loader generates an error and terminates the command.
- If you do not describe the columns of the source table in the command, you must not specify the OUTFIELDS key word. The data is output as [plain text values \[Page 117\]](#).
- If you have defined the format [COMPRESSED \[Page 109\]](#) for the data stream, the values of the columns to be extracted are output in the order in the SELECT statement, and separated with separators.
- If you have defined the format [FORMATTED \[Page 111\]](#) for the data stream, the values of the columns to be extracted are formatted and output in the order in the SELECT statement. The length of the individual output values depends on the defined sizes of the individual columns in the source table.
- If the SELECT statement does not contain any column names (**DATAEXTRACT \* FROM . . .**), the columns are formatted according to the format of the data stream. They are output in the order specified by the database for processing the command.



Definition of the source table:

```
create table customer (cno char (4), name char (6) NOT NULL,
zip integer, city char (11), PRIMARY KEY (cno))
```

Unload command:

```
DATAEXTRACT * from customer
OUTFIELDS
  cno      01-04
  name     06-12
  zip      14-18
  city     20-31
OUTFILE 'customer.data' FORMATTED;
```



## outstream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<outstream_spec> ::= OUTSTREAM <mediumtype_spec [Page 82]>
'<valSTREAM_NAME>' [<stream_format_spec [Page 99]>] [<stream_extract [Page
99]>] [APPEND]
```

valSTREAM_NAME	Name and path of the <a href="#">medium [Page 122]</a> , in a <a href="#">DATAEXTRACT command [Page 45]</a> , also the name of the <a href="#">command file [Page 20]</a> to be generated
APPEND	An existing medium is not overwritten. This means that data or generated load statements are added to the end of the medium.    This does <b>not</b> apply to <a href="#">TABLEEXTRACT commands [Page 54]</a> . A TABLEEXTRACT command overwrites an existing medium.

### Use

When unloading data, use this syntax rule to describe the [data stream \[Page 108\]](#).



## packagestream\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<packagestream_spec> ::= PACKAGE <outstream_spec [Page 89]>
```

### Use

[DBEXTRACT command \[Page 48\]](#)

[DBLOAD command \[Page 50\]](#)

[TABLEEXTRACT command \[Page 54\]](#)

[TABLELOAD command \[Page 55\]](#)



## part\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

`<part_spec> ::= TABLE <table_name [Page 100]> | USER | ALL`

<b>TABLE</b>	A single table is unloaded to a <a href="#">data stream [Page 108]</a> or loaded from a data stream. Prerequisite: The user must be the <a href="#">owner</a> of this table
<b>USER</b>	The tables of the user currently logged on to the SAP DB Loader are unloaded to one or more data streams or the tables unloaded with the corresponding commands are restored from these data streams. The user who restores the tables does not need to be the user who saved them.  If you are loading tables that were unloaded by the user <code>user1</code> , the tables can be reloaded by <code>user2</code> .
<b>ALL</b>	All tables of the database catalog are unloaded to one or more data streams or are loaded from these data streams. Prerequisite: The user must be the <a href="#">SYSDBA</a> user.

## Use

This syntax rule is used in the following commands: [TABLEEXTRACT \[Page 54\]](#), [TABLELOAD \[Page 55\]](#), [CATALOGEXTRACT \[Page 43\]](#), or [CATALOGLOAD command \[Page 44\]](#).

You specify whether an individual table, the tables of the user logged on, or all tables of the database catalog are loaded or unloaded.



## restart

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

`<restart> ::= RESTART <valPACKAGEGUID>`

<b>valPACKAGEGUID</b>	ID that is generated during the Loader runtime for the commands that span more than one table
-----------------------	---

## Use

The PACKAGEGUID and the host name of the client of the SAP DB Loader are entered in the table [TRANSFORMATIONPACKAGES \[Page 40\]](#) and logged in the [log file \[Page 37\]](#).

The user can determine the PACKAGEGUID from the table TRANSFORMATIONPACKAGES or from the log file to restart the command with the corresponding PACKAGEGUID.



## restore\_spec

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

`<restore_spec> ::= FOR DATALOAD | FOR FASTLOAD | FOR DATAUPDATE`

## Use

Use this syntax rule in a [DATAEXTRACT command \[Page 45\]](#) to specify which command is to be generated for loading data in the [command file \[Page 20\]](#).

If a [FASTLOAD command \[Page 51\]](#) is generated this command contains the value of 80% for the fill level of the database ([<usage\\_spec> \[Page 102\]](#)). This is the default value of the SAP DB Loader.



```
DATAEXTRACT FOR DATALOAD TABLE article
OUTFILE 'article.control'
OUTFILE 'article.data'
```



## round\_or\_trunc\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

**<round\_or\_trunc\_spec> ::= ROUND <valFRACTION> | TRUNC <valFRACTION>**

<b>valFRACTION</b>	<p>Number of decimal places</p> <p>The value must be between 0 and 18. This function does not have any effect if the number does not have any decimal places.</p>
--------------------	---

## Use

Use this syntax rule to specify the number of decimal places in a number.

### ROUND <valFRACTION>

The value is rounded off at the (**<valFRACTION>+1**)th decimal place. If this number is  $\geq 5$ , the value is rounded up. If it is  $< 5$ , the value is rounded down. The result is a number in which the (**<valFRACTION>+1**)th and all subsequent decimal places are equal to 0. The other digits in the number may have been changed if the value was rounded up.



```
DATALOAD TABLE distance
...
cm 7 SCALE 2
cm 7 SCALE -3 ROUND 1
...
INFILE 'meter.data'
```

### TRUNC <valFRACTION>

The (**<valFRACTION>+1**)th and all subsequent decimal places of the value are set to 0. The first **<valFRACTION>** decimal places remain unchanged.



```
DATAEXTRACT * FROM distance
cm 10-14 INTEGER SCALE 2
m 14-17 INTEGER
km 18-21 INTEGER SCALE -3 TRUNC 2

OUTFILE 'dimensions.bin' FORMATTED
```

## scale\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<scale_spec> ::= SCALE <valSCALE_FACTOR>`

<code>valSCALE_FACTOR</code>	Scaling factor, can be positive or negative The value to which the syntax rule refers is multiplied by the corresponding power of ten.
------------------------------	---

### Use

You can scale values using this syntax rule.

You can combine this syntax rule with [round or trunc spec \[Page 91\]](#).



```

DATALOAD TABLE distance
...
cm 7 SCALE 2
cm 7 SCALE -3
...
INFILE 'meter.data'

```



```

DATAEXTRACT * FROM
cm 10-14 INTEGER SCALE 2
m 14-17 INTEGER
km 18-21 INTEGER SCALE -3

OUTFILE 'dimensions.bin' FORMATTED

```

## select\_expression

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<select_expression>`

This database query is formulated in the same way as a [SELECT statement](#), however the keyword `DATAEXTRACT` replaces the keywords `EXTRACT DB`. All options of a `SELECT` statement are permitted:

- Select the result columns and determine their order in the result table
- Join multiple tables
- Use qualifications to select result lines
- Specify sort order
- Specify locks (WITH LOCK)

## Use

Use this syntax rule in a [DATAEXTRACT command \[Page 45\]](#) to specify which columns in a table to extract.



```
DATAEXTRACT * FROM article
OUTFIELDS
  ano      01-08
  descr    09-39
  stock    40-51
  min_ord  52-63
  price    64-74
  weight   75-85
OUTFILE 'article.data' FORMATTED
```



```
DATAEXTRACT ano,descr,stock,min_ord FROM article order by
ano
OUTFIELDS
  ano      1
  descr    2
  stock    3
  min_ord  4
OUTFILE 'article.data'
```



```
DATAEXTRACT ano,descr,stock,min_ord FROM article WITH LOCK
OUTFIELDS
  ano      1
  descr    2
  stock    3
  min_rod  4
OUTFILE 'article.data'
```



## separator\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<separator_spec> ::= SEPARATOR '<valSEPARATOR>'
```

valSEPARATOR	Separator for data fields; must be one character long
--------------	---

### Use

Use this syntax rule to specify which character is used to separate data fields in [data streams \[Page 108\]](#) that have the [COMPRESSED \[Page 109\]](#) format.

Use the [SET command \[Page 53\]](#) to view the SAP DB Loader default.

#### See also:

[Data Format \[Page 109\]](#)

## **set\_column\_spec**

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<set_column_spec> ::= SET <load\_column\_spec \[Page 79\]>`

### Use

Use this syntax rule in a [DATAUPDATE command \[Page 48\]](#) to define the columns of the target table that you want to update with this command.



```
DATAUPDATE TABLE customer
KEY   cno  01-04
SET  city 05-16
INFILE 'customer.data' FORMATTED
```

## **set\_column\_spec\_mlt**

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<set_column_spec_mlt> ::= <set\_column\_spec \[Page 94\]>  
<set_column_spec_mlt>`

### Use

Use this syntax rule in a [DATAUPDATE command \[Page 48\]](#) to define the columns of the target table that you want to update with this command.



```
DATAUPDATE TABLE customer
KEY   cno  01-04
SET  city 05-16
INFILE 'customer.data' FORMATTED
```

## **sequence\_number**

[Syntax Rules for Column Descriptions \[Page 60\]](#)

### Syntax

`<sequence_number> ::= SEQNO | SEQNO <valSTART>  
| SEQNO <valSTART> <valINCREMENT>`

<b>valSTART</b>	Start value and first value to be loaded
<b>valINCREMENT</b>	Increment between loaded values

The keyword **SEQNO** specifies that sequential numbers are generated. Both the start value and the increment can be negative.

If you only enter the keyword **SEQNO** in the command, the start value is 0 and the increment between values is 1.

If you only enter the start value in the command after **SEQNO**, the increment between the values is 1.

## Use

Use this syntax rule to load self-generated sequential numbers with a freely definable start value and increment.

The set of calculable numbers is cyclical. This means that the value that comes after the largest possible value is the smallest possible value. In this way, you can generate as many numbers as you want, including repeated sequential numbers.

A column of the type **FIXED(10)** is enough for storing the sequential numbers.



```

DATALOAD TABLE article
  ano          SEQNO 10000 5
  descr        2
  stock        3 NULL IF POS 3 < '0'
INFILE 'article.data' FORMATTED

```



## simple\_column\_spec

[Syntax Rules for Column Descriptions \[Page 60\]](#)

## Syntax

**<simple\_column\_spec>** ::= **<column\_descriptor [Page 66]>** | **<column\_assignment [Page 65]>**

## Use

Use this syntax rule in a [DATAUPDATE command \[Page 48\]](#) to assign the data fields of the qualification columns in the data stream to the columns of the target table and to specify the [external data types \[Page 114\]](#).



```

DATAUPDATE TABLE article
  ano    01-08      CHAR
  descr  09-39      CHAR
SET min_ord 40-41  INTEGER
SET del_date 42-49 CHAR
INFILE 'article.data' FORMATTED

```



## simple\_condition

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

**<simple\_condition>** ::= **POS <field\_pos [Page 74]> <field\_format [Page 73]> [HEX]**  
**<compare\_operator [Page 69]> '<valLiteral>'**

<b>valLITERAL</b>	Constant
-------------------	----------

As with the other fields in a data record, you use the position of a value that you want to compare to describe it. You only need to specify the [external data type \[Page 114\]](#) of this value if the data type is not CHAR.

You specify a constant as a [plain text value \[Page 117\]](#) and place it in quotation marks. The constant is converted to the data type of the value that you want to compare in the data record.

If the constant you want to use as a comparison value is a number, it must have a valid number format, that is, it must be a floating point number in mantissa/exponent notation or a fixed-point number with the currently defined decimal representation or the default decimal representation in the SAP DB Loader.

## Use

Use this syntax rule to define the selection criterion that determines which records from the [data stream \[Page 108\]](#) are loaded to which target table. The data records that are you want to load are selected by comparing them with a constant.



You want to load only those data records from the source data stream `hotel.data` where the price is less than `400.00` to the target table `hotel`:

```

DATALOAD TABLE hotel
  IF POS 41-44 REAL < '400,00'
    hno      01-04 INTEGER
    name     09-18
    zip      20-25 DECIMAL
    city     27-36
    price    41-44 REAL
  INFILE 'hotel.data' FORMATTED

```



## sql\_mode

[Syntax Rules for Setting up a Session \[Page 59\]](#)

## Syntax

`<sql_mode> ::= INTERNAL | ANSI | DB2 | ORACLE`

<b>INTERNAL</b>	Internal database system <a href="#">SQL Mode</a> (system default)
<b>ANSI</b>	SQL dialect in accordance with the ANSI standard (ANSI X3.135-1992, Entry SQL)
<b>DB2</b>	SQL dialect in accordance with the definition for DB2 Version 4
<b>ORACLE</b>	SQL dialect in accordance with the ORACLE7 definition

## Use

The active SQL mode determines which SQL dialect is understood by the database system. It is transferred between the SAP DB Loader and the database instance when the connection is set up. The [SQL statements \[Page 13\]](#) and load [commands \[Page 42\]](#) are interpreted in accordance with the specified SQL mode, if the [data stream \[Page 108\]](#) does not have the format [PAGES \[Page 112\]](#).

- If you do not specify an SQL mode, the SQL mode INTERNAL is used.

- If you enter an invalid SQL mode, the database system automatically assumes that you want to use the INTERNAL SQL mode.
- The SQL mode can change within a [command file \[Page 20\]](#) or a script.



## standard\_code\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<standard_code_spec> ::= ASCII | UCS2 | UTF8
```

### Use

In the [code\\_spec \[Page 65\]](#), you define the default value for the interpretation of [data streams \[Page 108\]](#) that contain [plain text values \[Page 117\]](#). The default value of the SAP DB Loader is ASCII.

When you load or unload data, you can transform it between various code types.

If you load/unload ASCII data into an ASCII database or UCS2 data into a [Unicode](#) database, the data does not have to be converted.

The following combinations are possible ([internal database data type \[Page 114\]](#) and [external data type \[Page 114\]](#)):

Internal database data type	Possible code attributes for data streams with plain text values when unloaded
(VAR)CHAR ASCII	ASCII, UCS2, UTF8
(VAR)CHAR UNICODE	UCS2, UTF8
(VAR)CHAR BYTE	BINARY, ASCII (HEX), UCS2 (HEX), UTF8 (HEX)
Numerical data types	ASCII, UCS2, UTF8

Internal database data type	Possible code attributes for data streams with plain text values when loaded
(VAR)CHAR ASCII	ASCII
(VAR)CHAR UNICODE	ASCII, UCS2, UTF8
(VAR)CHAR BYTE	BINARY, ASCII, UCS2 (HEX), UTF8 (HEX)
Numerical data types	ASCII, UCS2, UTF8

If you want to use a particular code page for the ASCII to UCS2 conversion, you can specify the page with the [SET command \[Page 53\]](#) `SET <code_page_spec [Page 63]>`.

When you load/unload LONG data, you determine the default values by using the syntax rule [longfile\\_code\\_spec \[Page 80\]](#).



## standard\_date\_mask

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

`<standard_date_mask> ::= ISO | USA | EUR | JIS | INTERNAL`

ISO	YYYY-MM-DD
USA	MM/DD/YYYY
EUR	DD.MM.YYYY
JIS	YYYY-MM-DD
INTERNAL	YYYYMMDD

Y stands for year, M for month, and D for day.

## Use

Use it to specify the format for [plain text values \[Page 117\]](#) in which DATE columns are entered and output.

This format applies to both load and unload [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#).

The default value of the SAP DB Loader is INTERNAL.



## standard\_time\_mask

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

## Syntax

`<standard_time_mask> ::= ISO | USA | EUR | JIS | INTERNAL`

ISO	HH.MM.SS
USA	HH:MM AM (PM)
EUR	HH.MM.SS
JIS	HH:MM:SS
INTERNAL	HHHHMMSS

H stands for hours, M for minutes, and S for seconds.



You can use the usual standard formats, or define your own format. To do this, use **H** for hours, **M** for minutes, and **S** for seconds. Use two figures for minutes and seconds. You can choose any number of figures for **H**.

## Use

Use this syntax rule to specify the data format of the [plain text values \[Page 117\]](#) in which TIME columns are entered and output.

This format applies to both load and unload [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#).

You can also change the current value for individual commands.

The default value of the SAP DB Loader is INTERNAL.

## standard\_timestamp\_mask

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<standard_timestamp_mask> ::= ISO | USA | EUR | JIS | INTERNAL
```

ISO	YYYY-MM-DD-HH.MM.SS.NNNNNN
USA	ISO
EUR	ISO
JIS	ISO
INTERNAL	YYYYMMDDHHMMSSNNNNNN

H stands for hours, M for minutes, S for seconds, and N for milliseconds or microseconds.

### Use

Use this syntax rule to specify the data format of the [plain text values \[Page 117\]](#) in which TIMESTAMP columns are entered and output.

This format applies to both load and unload [commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#).

The default value of the SAP DB Loader is INTERNAL.

## stream\_extract

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<stream_extract> ::= START <valSTART_POS>
| START <valSTART_POS> <valEND_POS>
| START <valSTART_BLOCK> : <valSTART_POS> <valEND_BLOCK> :
<valEND_POS>
```

valSTART_POS	Number of the first data record to be loaded
valEND_POS	Number of the last data record to be loaded
valSTART_BLOCK	Block number of the first data record to be loaded
valEND_BLOCK	Block number of the last data record to be loaded

## stream\_format\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<stream_format_spec> ::=
  <code_spec [Page 65]> <stream_format_spec>
| <number_spec [Page 86]> <stream_format_spec>
| <date_spec [Page 71]> <stream_format_spec>
```

```

| <time_spec [Page 101]> <stream_format_spec>
| <timestamp_spec [Page 102]> <stream_format_spec>
| <null_spec [Page 85]> <stream_format_spec>
| <bool_spec [Page 62]> <stream_format_spec>
| <int_spec [Page 77]> <stream_format_spec>
| <separator_spec [Page 93]> <stream_format_spec>
| <delimiter_spec [Page 72]> <stream_format_spec>
| COMPRESSED [Page 109] <stream_format_spec>
| FORMATTED [Page 111] <stream_format_spec>
| FORMATTED BINARY [Page 111] <stream_format_spec>
| PAGES [Page 112] <stream_format_spec>
| RECORDS [Page 112] <stream_format_spec>

```

## Use

Use this syntax rule within a load and unload command to define the [data formats \[Page 109\]](#) and the [format of the data stream \[Page 109\]](#).



## table\_name

[Syntax Rules for Table Descriptions \[Page 59\]](#)

## Syntax

```
<table_name> ::= <valTABLE_NAME> | <valTABLE_OWNER>.<valTABLE_NAME>
```

valTABLE_NAME	Name of the table Specify the table name in accordance with the SQL conventions. You can also define a user name as a prefix.
valTABLE_OWNER	Owner of the table

## Use

Use this syntax rule in a [command \[Page 42\]](#) to [load](#) or [unload data](#), or to specify a table name.



```

FASTLOAD TABLE hotel
  IF (POS 41-44 REAL < '400.00')
  AND
    (POS 41-44 REAL >= '400.00')
    hno      01-04 INTEGER
    name     09-18
    zip      20-25 DECIMAL
    city     27-36
    price    41-44 REAL
INFILE 'hotel.data' FORMATTED;

DATALOAD TABLE berolina.hotel
  IF (POS 41-44 REAL < '400.00')
  AND
    (POS 41-44 REAL >= '400.00')
    hno      01-04 INTEGER
    name     09-18
    zip      20-25 DECIMAL
    city     27-36

```

```
price 41-44 REAL
INFILE 'hotel.data' FORMATTED;
```

## table\_spec

[Syntax Rules for Table Descriptions \[Page 59\]](#)

### Syntax

`<table_spec> ::= <table_name [Page 100]> <if_condition [Page 76]>`

### Use

Use this syntax rule in a [DATALOAD command \[Page 47\]](#) to specify the name(s) and contents of the target table(s).

It also applies to loading a table in the [FASTLOAD command \[Page 51\]](#).



You want to load from the [data stream \[Page 108\]](#) `addresses.data` the data records that begin with the letter 'k' into the target table with the name `customer`.

```
DATALOAD TABLE sqltravel00.customer      IF POS 1 = 'k'
customerid 2
name       3
zip        5
address    6
```

## time\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

`<time_spec> ::= TIME <standard_time_mask [Page 98]> | TIME  
'<valFREE_MASK>'`

<code>valFREE_MASK</code>	Self-defined output format Use <b>H</b> for hours, <b>M</b> for minutes, and <b>S</b> for seconds. Use two digits for minutes and seconds. You can define any number of digits for <b>H</b> .
---------------------------	--

### Use

Use this syntax rule to specify the [data format \[Page 109\]](#) for [plain text values \[Page 117\]](#) in which TIME columns are entered and output.

This data format applies only to the load or unload command in which it is specified. Otherwise, the SAP DB Loader default is used.

Use the [SET command \[Page 53\]](#) to view the SAP DB Loader default.

## timestamp\_spec

[Syntax Rules for Describing the Data Stream \[Page 60\]](#)

### Syntax

```
<timestamp_spec> ::= TIMESTAMP <standard_timestamp_mask [Page 99]>
| TIMESTAMP '<valFREE_MASK>'
```

<b>valFREE_MASK</b>	Self-defined output format  Use <b>H</b> for hours, <b>M</b> for minutes, and <b>S</b> for seconds. Use two digits for minutes and seconds. You can define any number of digits for <b>H</b> .
---------------------	--

### Use

Use this syntax rule to specify the [data format \[Page 109\]](#) of the [plain text values \[Page 117\]](#) in which **TIMESTAMP** columns are entered and output.

This data format applies only to the load or unload command in which it is specified. Otherwise, the SAP DB Loader default is used.

Use the [SET command \[Page 53\]](#) to view the SAP DB Loader default.

## usage\_spec

[Syntax Rules for Table Descriptions \[Page 59\]](#)

### Syntax

```
<usage_spec> ::= WITH <valUSAGE> % USAGE | WITH <valUSAGE> ROWS USAGE
```

<b>valUSAGE</b>	Positive integer  For a percentage, this number must be between 50 and 100.  If you specify the number of rows, the length of a single table record plays a role.
-----------------	---

### Use

Use this syntax rule in a [FASTLOAD command \[Page 51\]](#) to specify the extent to which you want to fill a table page with data records.

To do this, you can specify a percentage between 50 and 100 or define the number of rows (data records) that you want to load in each table page. If you specify a number of rows that exceeds the actual number possible, the SAP DB Loader displays an error message during the load operation, indicating the maximum number of rows permitted in each table page.

The default value of the SAP DB Loader is 80%.

- If the table is not modified at all, or only slightly, it is a good idea to utilize the occupied memory by more than 80%.
- If considerable dynamic growth is anticipated for the table, it is a good idea to utilize the occupied memory by less than 80%.



Specifying the desired memory utilization does not guarantee that it will actually be achieved. Use the Database Manager program to check the current memory utilization for the relevant table.



You want to specify a memory utilization of 100% for a FASTLOAD of data from the [data stream \[Page 108\]](#) `customer.data` to the `customer` table.

```
FASTLOAD with 100 % USAGE
TABLE customer
  cno          1
  name        2
  zip         3
  city        4
INFILE 'customer.data'
```



You want to enter 25 data records in each table page on the database instance.

```
FASTLOAD with 25 ROWS USAGE
TABLE customer
  cno          1
  name        2
  zip         3
  city        4
INFILE 'customer.data'
```



## user\_statement

[Syntax Rules for Setting up a Session \[Page 59\]](#)

### Syntax

```
<user_statement> ::= USER <valUSERNAME> <valPASSWORD>
[<database_name_statement> [Page 71]]
```

valUSERNAME	User name
valPASSWORD	User password

The database system automatically converts the user name and password to uppercase letters. If you want to suppress the conversion, you must place the names in quotation marks.

### Use

The user data transferred is used to set up a [database session](#) with the database instance.



## Keywords

Keywords are components of [commands \[Page 42\]](#). They define the function of the command, which is modified by arguments and options.



```
DATALOAD TABLE article
  ano          01-08 CHAR
  descr        09-39 CHAR
  stock        40-43 INTEGER
  min_order    44-45 INTEGER
```

```

ordered      46-49 INTEGER
del_date    50-57 CHAR
price       58-65 DECIMAL (2)
weight      66-69 REAL
INFILE 'customer.data' FORMATTED

```

The SAP DB Loader uses the keywords listed below. In some cases, you can also use the specified alternatives to the keywords.

[Keywords A - C \[Page 104\]](#)

[Keywords D - E \[Page 104\]](#)

[Keywords F - K \[Page 105\]](#)

[Keywords L - P \[Page 105\]](#)

[Keywords R - S \[Page 106\]](#)

[Keywords T - Z \[Page 106\]](#)



## Keywords A - C

[Keywords \[Page 103\]](#) of the SAP DB Loader

Keyword	Alternative Keywords
ALL AND ANSI APPEND ASCII AUTOCOMMIT BINARY BOOLEAN BY CATALOG CATALOGEXTRACT CATALOGLOAD CHAR CODESET CODETYPE COMPRESS CONFIGURATION COUNT CURRENT	EXTRACT CATALOG LOAD CATALOG



## Keywords D - E

[Keywords \[Page 103\]](#) of the SAP DB Loader

Keyword	Alternative Keywords
DATA DATAEXTRACT DATALOAD DATAUPDATE DATE	EXTRACT DATA LOAD DATA UPDATE DATA

DB2 DBEXTRACT DBLOAD DEC DECIMAL DEFAULT DELIMITER DUPLICATES EBCDIC EUR EXTRACT CATALOG EXTRACT DATA EXTRACT DB EXTRACT TABLE	EXTRACT DB LOAD DB  CATALOGEXTRACT DATAEXTRACT DBEXTRACT TABLEEXTRACT
---	---



## Keywords F - K

[Keywords \[Page 103\]](#) of the SAP DB Loader

FASTLOAD FILE FOR FORMATTED HEX HILO IF IGNORE INFILE INSTALLATION INSTREAM INTERNAL INTEGER ISO JIS KEY
---



## Keywords L - P

Keywords of the SAP DB Loader

Keywords	Alternative Keyword
LANGUAGE LOAD CATALOG LOAD DATA LOAD DB LOAD TABLE LOHI LONGFILE MESSAGE NOT NULL NUMBER	CATALOGLOAD DATALOAD DBLOAD TABLELOAD

OFF ON OR ORACLE ORDER OTHERWISE OUTFILE OUTFIELDS OUTSTREAM PACKAGE PAGES PIPE POS	
---	--



## Keywords R - S

[Keywords \[Page 103\]](#) of the SAP DB Loader

REAL RECORDS REJECT RELEASE RESTART ROUND ROWS SCALE SEPARATOR SEQNO SERVERDB SET SQLMODE STAMP START SQLID SYSDATE
---



## Keywords T - Z

[Keywords \[Page 103\]](#) of the SAP DB Loader

Keywords	Alternative Keywords
TABLE	
TABLEEXTRACT	EXTRACT TABLE
TABLELOAD	LOAD TABLE
TABLEUNLOAD	
TABLEUPDATE	UPDATE TABLE
TAPE	
TERMCHARSET	
TIME	
TIMESTAMP	
TRUNC	
UID	

UPDATE UPDATE DATA UPDATE TABLE USA USAGE USE USER USERKEY USERGROUP VERSION WITH ZONED	DATAUPDATE TABLEUPDATE
--	---------------------------



## Terms and Procedures

[Binary Values \[Page 107\]](#)

[Data \[Page 108\]](#)

[Data stream \[Page 108\]](#)

[Data Types \[Page 113\]](#)

[Unloading LONG Values \[Page 115\]](#)

[HEX Values \[Page 117\]](#)

[Plain Text Values \[Page 117\]](#)

[Loading Binary Values \[Page 118\]](#)

[Loading LONG Values \[Page 118\]](#)

[Medium \[Page 122\]](#)

[Metadata \[Page 122\]](#)

[Selecting Data Records \[Page 123\]](#)



## Binary Values

The following applies for the SAP DB Loader:

All numeric [external data types \[Page 114\]](#) (INTEGER, REAL, DECIMAL, ZONED) are interpreted as binary values. These can be converted to any of the [internal database data types \[Page 114\]](#) (FIXED, SMALLINT, INTEGER, FLOAT).

- Data of the data type INTEGER is always interpreted as signed values by the SAP DB Loader. You can adjust its display in a [data stream \[Page 108\]](#) with [int\\_spec \[Page 77\]](#) (Specified in commands to load and unload as a processing specification for the data stream as the syntax rule [stream\\_format\\_spec \[Page 99\]](#)).
- Data of the data type REAL is coded in host-specific format and cannot be transported between different operating systems and computer platforms without being converted again.

**See also:**

[Converting the Data Types \[Page 114\]](#)



## Data

The SAP DB Loader provides functions to load, unload, and format data between different data sources and data targets in different granularities ([Commands for Loading and Unloading \[Page 12\]](#)).

The SAP DB Loader differentiates the following data:

- [Application Data \[Page 108\]](#)
- [Metadata \[Page 122\]](#)



## Application Data

Application data is the [data \[Page 108\]](#) that is stored in the database tables.

The SAP DB Loader writes the application data to a [data stream \[Page 108\]](#) when unloading from the database system and to the database system when loading from a data stream.



## Metadata

The SAP DB database system manages its metadata in the [database catalog](#).

For the SAP DB Loader, metadata is data that mirrors the database catalog information and also describes the data, the [application data \[Page 108\]](#), database catalog information and load and unload processes ([Transformations \[Page 10\]](#)) in accordance with the requirements of the SAP DB Loader.

- [Metadata of the Application Data \[Page 122\]](#)
- [Metadata of the Database Catalog \[Page 122\]](#)
- [Metadata of the Transformations \[Page 123\]](#)



## Data Stream

The SAP DB Loader transforms metadata of the database catalog, application data, and pages of a SAP DB database instance to metadata and application data (data definitions, data records, and data fields), and pages and conversely in a format that is readable for the SAP DB Loader. The data is unloaded into data streams, or loaded from these streams.

Every data stream is identified using the following data:

- [Metadata \[Page 122\]](#) and/or [Application Data \[Page 108\]](#)
- [Data Format \[Page 109\]](#) and [Format of the Data Stream \[Page 109\]](#)
- [Direction \[Page 113\]](#)
- [Medium \[Page 122\]](#)



## Data Format

With most commands for loading and unloading, you can specify the [external data type \[Page 114\]](#) and the data format that the relevant field values in the [data stream \[Page 108\]](#) have or should have.

You can also define which separators and delimiters are to be used.

### Procedure

You can specify the data format in load and unload commands as a processing specification for the data stream (syntax rule [stream\\_format\\_spec \[Page 99\]](#)).

[code\\_spec \[Page 65\]](#)

[number\\_spec \[Page 86\]](#)

[date\\_spec \[Page 71\]](#)|[time\\_spec \[Page 101\]](#)|[timestamp\\_spec \[Page 102\]](#)

[null\\_spec \[Page 85\]](#)

[bool\\_spec \[Page 62\]](#)

[int\\_spec \[Page 77\]](#)

[separator\\_spec \[Page 93\]](#)|[delimiter\\_spec \[Page 72\]](#)

The specified data format applies to the columns that have the corresponding [internal database data type \[Page 114\]](#).



## Format of the Data Stream

Data streams can have the following formats:

- [COMPRESSED \[Page 109\]](#)
- [FORMATTED \[Page 111\]](#)
- [FORMATTED BINARY \[Page 111\]](#)
- [PAGES \[Page 112\]](#)
- [RECORDS \[Page 112\]](#)

You can specify the format in commands to load an unload as a processing specification for the [data stream \[Page 108\]](#) (syntax rule [stream\\_format\\_spec \[Page 99\]](#)).

The SAP DB Loader requires that data streams with [clear text values \[Page 117\]](#) have the format COMPRESSED. If this is not the case, you must explicitly specify the FORMATTED format or, with [binary \[Page 107\]](#) coded data streams, FORMATTED BINARY format in the load or unload command.



## COMPRESSED

COMPRESSED (CSV, Comma Separated Values) is a possible [format of a data stream \[Page 109\]](#).

If a [command \[Page 42\]](#) to load to unload does not explicitly specify the format [FORMATTED \[Page 111\]](#) or [FORMATTED BINARY \[Page 111\]](#), the system assumes that the data in the [source data stream \[Page 108\]](#) is in the COMPRESSED format (SAP DB Loader default value). This means that it does not have to be specified explicitly in a load or unload command.

The COMPRESSED format can only be used for source data streams with [plain text values \[Page 117\]](#). Data streams with [binary values \[Page 107\]](#) must be loaded with the [FORMATTED BINARY \[Page 111\]](#) format.

If you do not specify a format in a load or unload command, however the data is not in plain text, the SAP DB Loader creates an error message.



`customer.data`

Position number	1	2	3	4	....
	001,	miller,	10277,	New York	
	002,	smith,	60678,	Chicago	
	003,	kleinert,	33184,	Miami	

The data fields do not have a uniform format but are separated by commas:

FASTLOAD command:

```
FASTLOAD TABLE customer
  cno      1
  name    2
  zip     3
  location 4
INFILE 'customer.data'
```

## Data Line

In COMPRESSED format, a data line must at least be long enough to represent the data. The length of the individual data fields and the total length of the data lines can vary. Each data line ends with a line break.

When data is loaded, a data line in a source data stream of this format corresponds to exactly one data record in the target table.

When data is unloaded, a data line in a source table of this format corresponds to exactly one data record in the target data stream.

## Data Fields

Data fields in the data streams must have the [external data type \[Page 114\]](#) CHAR.

The individual data fields are separated by a freely selectable character (SAP DB Loader default value: comma). They can also be encapsulated in freely selectable characters (SAP DB Loader default value: double quotation marks).

If delimiters follow each other directly in a data line, then the value is the empty character string. If the relative position is larger than the number of values in the line, the value is also the empty character string.

## Position Specification

The assignment of a data field of the data stream to a column in the table is determined by the position specification [<field\\_pos> \[Page 74\]](#) in a command for loading or unloading data. This specifies a relative position within the data record. It therefore specifies which value in a data record is to be loaded into or unloaded from a specific column.

When you load data from a source table, the data fields in the target table in the database must be sorted in ascending order. You can also assign the same input values to different columns or even omit positions. If consecutive delimiters are detected or if the relative position is greater than the number of values in the line, the value is the empty character string.

When you unload data from the database to a target data stream, assign the position 1 to the first column in your list. The position numbers of the following columns rise by 1 each time. This also means that each position number can only be assigned once.



The FORMATTED BINARY format is largely the same as the [FORMATTED \[Page 111\]](#) format. The only difference is that the length of a data record is only determined by the total length of the individual data fields.



**customer.data**

Position no.	1	2	3	4	5	6	7	8	9	.....
	0	1	m	i	l	l	e	r		1
										0
										2
										s
										m
										i
										t
										0
										3
										k
										l
										e
										i
										n
										e
										r
										t
										3
										3
										1
										8
										4
										M
										i
										a
										m
										i

The data fields have a uniform format.

FASTLOAD command:

```
FASTLOAD TABLE customer
  cno          01-02
  name        03-10
  zip         11-15
  place      16-26
INFILE 'customer.data' FORMATTED BINARY
```

## Data Fields

The data fields can contain special characters and all have the same fixed length. This also applies to the individual data records.

The assignment between data fields in the [data stream \[Page 108\]](#) and table columns is made in accordance with the [<field\\_pos> \[Page 74\]](#) position specification in the load or unload command.

Unlike the FORMATTED format, a data record does **not** end with an additional line break.

## Position Specifications

Data fields are described by their byte start and end positions. The first possible byte start position of a data record is 1. Specifying an end position is optional. If you do not specify an end position, the corresponding data field has a length of 1.

The data fields of the data stream can be assigned in any order to the table columns. The order does not affect the processing speed.

The positions do not need to follow each other directly when you load or unload data. When it unloads data, the SAP DB Loader fills any gaps with blank characters.



## PAGES

PAGES is a possible [format of a data stream \[Page 109\]](#). The [application data \[Page 108\]](#) is stored in [pages](#).



## RECORDS

RECORDS is a possible [format of the data stream \[Page 109\]](#).

This is a proprietary SAP DB format.



## Direction

Direction of the [data stream \[Page 108\]](#)

INSTREAM: From a SAP DB Loader data stream to a SAP DB database instance

OUTSTREAM: From a SAP DB database instance to a SAP DB Loader data stream



## Data Types

With most commands for loading and unloading, you can specify the data type which the relevant field values in the [data stream \[Page 108\]](#) have or should have. This specification tells the SAP DB Loader how to interpret the data in the data stream.

- When loading [application data \[Page 108\]](#) in a table, the specified [external data types \[Page 114\]](#) are converted to [internal database types \[Page 114\]](#) and the corresponding column values of a table are loaded.
- When unloading application data from a table, the internal database types are converted to the specified external data types and the corresponding column values are unloaded.

### See also:

[Converting the Data Types \[Page 114\]](#)



You cannot specify an external data type with the [TABLELOAD \[Page 55\]](#) and [TABLEEXTRACT \[Page 54\]](#) commands.

The external data type and the corresponding internal database data type of the column value in the target table do not have to be identical.



The database table `article` requires the internal database data type `FIXED` for the value in the `ordered` column.

The `FASTLOAD` command defines the external data type `INTEGER` for the column `ordered`.

```
FASTLOAD TABLE article
ano          01-08
descr       09-39 CHAR
stock       40-43 INTEGER
min_ord     44-45 INTEGER
ordered     46-49 INTEGER
del_date    50-57 CHAR
price       58-65 DECIMAL(2)
weight      66-69 REAL
INFILE 'article.data' FORMATTED
```

The data in the column `ordered` is converted to the internal database data type `FIXED` when it is loaded.



## Internal Database Data Types

Internal database data types are the [data types \[Page 113\]](#) defined for the columns in the database table.

For an explanation of all internal database data types, see *Reference Manual: SAP DB 7.4, Data Type (data\_type)* section.



## External Data Types

External data types are the [data types \[Page 113\]](#) that the SAP DB Loader can convert to [internal database data types \[Page 114\]](#) and to which the SAP DB Loader can convert the internal database data types when unloading application data.

You can specify an external data type as a field specification in the command to load or unload for each field value to be loaded or unloaded ([format spec \[Page 75\]](#)).

You cannot specify an external data type in the commands [TABLEEXTRACT \[Page 54\]](#) and [TABLELOAD \[Page 55\]](#).

If no external data type is specified for a field value, the SAP DB Loader selects the CHAR data type as the default for this field value. Therefore, if the field value does not have the data type CHAR or the data type CHAR is not desired, you must specify the data type.

The SAP DB Loader can process the following external data types:

CHAR ( <a href="#">plain text value [Page 117]</a> )	ASCII, UCS2, or UTF8-coded, depending on the server, max. 254 Bytes long
INTEGER ( <a href="#">Binary value [Page 107]</a> )	Binary coded (server-specific), 1, 2, or 4 bytes long, sign in bit 0, negative values in two's complement notation
REAL (Binary value)	Floating point notation with mantissa and exponent, 4 or 8 bytes long, in server-specific notation
DECIMAL (Binary value)	Packed decimal: one digit per half byte, sign in extreme right half byte, 1 to 10 bytes long, maximum of 18 digits, the position of the decimal point is derived from the table column type
DECIMAL(n) (Binary value)	n specifies the number of digits on the right of the decimal point
ZONED (Binary value)	Zoned decimal: /370 zoned data format is permitted, the position of the decimal point is derived from the table column type
ZONED(n) (Binary value)	n specifies the number of digits on the right of the decimal point



## Converting the Data Types

The SAP DB Loader can convert between numeric [data types \[Page 113\]](#), if necessary.

You can use the [external data type \[Page 114\]](#) CHAR to read or output any [internal database data type \[Page 114\]](#).

- When loading data into a table, the specified external data type is converted to internal database data types and the corresponding column values of a table are loaded.

- When unloading data from a table, the internal database types are converted to the specified external data types and the corresponding column values are unloaded.

	External Data Type	Internal Database Data Type
<b>Loading data to a database instance</b>	INTEGER, DECIMAL, ZONED, REAL	<a href="#">FIXED</a> , <a href="#">FLOAT</a> , <a href="#">SMALLINT</a> , <a href="#">INT[EGER]</a>
<b>Unloading data from a database instance</b>	FIXED, SMALLINT, INTEGER	INTEGER, <a href="#">DECIMAL</a> , <a href="#">ZONED</a> , <a href="#">REAL</a>
	FLOAT	REAL or <a href="#">CHAR[ACTER]</a>



```

DATALOAD TABLE article
  ano          01-08
  descr        09-39
  stock        40-43  INTEGER
  min_ord      44-45  INTEGER
  ordered      46-49  INTEGER
  del_date     50-57
  price        58-65  DECIMAL (2)
  weight       66-69  REAL
INFILE 'customer.data' FORMATTED

```



## Unloading LONG Values

When unloading application data, there are special features, if values with the [internal database data type \[Page 114\]](#) LONG are unloaded.

In the syntax rule [longfile spec \[Page 81\]](#) in [longfile spec mlt \[Page 81\]](#), specify the name and path of the LONG [data stream \[Page 108\]](#) in which the LONG values are to be inserted.

You can use both variants of the syntax rule to unload the data.

- Always use a column ID for the LONG column that you want to unload if a column ID is specified in the column list for this column (see [output column list \[Page 88\]](#)).
- Use the column name if the column name is specified in the column list.



```

DATAEXTRACT * FROM hotel
OUTFILE 'hotel.data'
LONGFILE info 'info.data'

DATAEXTRACT cno, name, zip, address, info FROM hotel
OUTFIELDS
  cno      1
  name     2
  zip      3
  address  4
  info     5
OUTFILE 'hotel.data'
LONGFILE info 'info.data'

```

When you unload LONG values, you must differentiate between the following cases:

- [Each LONG Value to Be Unloaded in a Separate LONG Data Stream \[Page 116\]](#)
- [All LONG Values to Be Unloaded in One LONG Data Stream \[Page 116\]](#)

## Each LONG Value to Be Unloaded in a Separate LONG Data Stream

Each LONG value to be unloaded from a column is written to a separate [data stream \[Page 108\]](#).

In the command for unloading data, you specify the name of the LONG data stream with a number of placeholders for sequential numbering of the generated LONG data streams for each LONG column that you want to unload.

If the LONG column you want to unload has no value in a data record (the value is an empty character string), an empty LONG data stream is generated for this LONG value.

The generated unique media name enables the individual LONG data streams to be assigned to the corresponding data record in the target table.



Use a sufficient number of numeric characters at the end of the media name as a placeholder. If the upper limit of the numbering is reached while the data is being unloaded, but there are still values left to be unloaded, the SAP DB Loader generates an error message and terminates the command.



```
DATAEXTRACT * FROM hotel
  cno      1
  name    2
  info    3
OUTFILE 'hotel.data' FORMATTED
LONGFILE info 'info.data.###'
```

The SAP DB Loader generates the data stream with the name `info.data.001` for the first LONG value to be unloaded, the data stream `info.data.002` for the second, and so on.

Content of the target data stream:

```
10,Excelsior, 'info.data.001'
30,Flora, 'info.data.002'
60,Bellevue, 'info.data.003'
```

### See also:

[Unloading LONG Values \[Page 115\]](#)

[All LONG Values to Be Unloaded in One LONG Data Stream \[Page 116\]](#)

## All LONG Values to Be Unloaded in One LONG Data Stream

All LONG values to be unloaded from a column are written in one [data stream \[Page 108\]](#).

In the command for unloading data you enter the name of a data stream for each LONG column, into which the each of the LONG values of this LONG column to be unloaded are entered.

If the LONG column you want to unload has no value in a data record (the value is an empty character string), the position specification for this LONG value is generated as follows: the start position is the end position of the preceding LONG value in the column plus 1; the end position is the end position of the preceding LONG value in the column. This means that the start position is always one value larger than the end position.

The start and end position of the generated LONG value in the data stream enable the LONG values to be assigned to data records.



```
DATAEXTRACT * FROM hotel
  cno      1
  name    2
  info    3
OUTFILE 'hotel.data'
LONGFILE info 'info.data'
```

Content of the target data stream:

```
10,Excelsior,1-880
30,Flora,881-1046
60,Bellevue,1047-1360
```

**See also:**

[Unloading LONG Values \[Page 115\]](#)

[Each LONG Value to Be Unloaded in a Separate LONG Data Stream \[Page 116\]](#)



## HEX Values

You can use the additional specification HEX to describe an [external data type \[Page 114\]](#) as a hexadecimal value.

This enables you to load or unload all data types in hexadecimal form. Every byte of data in the result is then represented by two hexadecimal numbers. Every value in the [data stream \[Page 108\]](#) then occupies twice as much space as the value of the same data type without the specification as a HEX value.



```
FASTLOAD TABLE customer
  cno      01-03 CHAR HEX
  name    04-12 CHAR HEX
  zip     13-17 INTEGER HEX
  place   18-29 CHAR HEX
INFILE 'customer.data' FORMATTED

DATALOAD TABLE article IF POS 40-47 INTEGER HEX > '0'
  ano      01-08
  descr    09-39 NULL IF POS 09-11 = ' '
  stock    40-47 INTEGER HEX
  min_ord  47-48 INTEGER
  price    48-55 DECIMAL (2)
  weight   56-59 REAL
INFILE 'article.data' FORMATTED
```



## Plain Text Values

Field values with the [external data type \[Page 114\]](#) CHAR are plain text values; that is, these values consist of readable characters.

- Plain text values can conform to different specifications: ASCII, UCS2 or UTF8 (data format [code spec \[Page 65\]](#)).

- You can specify the data formats for plain text values that represent the date, time, or a timestamp ([date\\_spec \[Page 71\]](#), [time\\_spec \[Page 101\]](#), [timestamp\\_spec \[Page 102\]](#)).

You can specify these [data formats \[Page 109\]](#) in load and unload commands as a processing specification for the data stream (syntax rule [stream\\_format\\_spec \[Page 99\]](#)).

Plain text values can be used to input data into columns of any [internal data type used by the database \[Page 114\]](#) and can be converted to any data type. If you want to load plain text values into numerical columns, the values must be able to be interpreted as numbers.



## Loading Binary Values

All numeric [external data types \[Page 114\]](#) (INTEGER, REAL, DECIMAL, ZONED) are interpreted as [binary values \[Page 107\]](#).

These can be converted to any of the [internal database data types \[Page 114\]](#) (FIXED, SMALLINT, INTEGER, FLOAT) ([Converting Data Types \[Page 114\]](#)).

- Data of the data type INTEGER is always interpreted as signed values by the SAP DB Loader. Use the syntax rule [<int\\_spec> \[Page 77\]](#) to specify how these values are represented in the [source data stream \[Page 108\]](#).
- Data of the data type REAL is coded in host-specific format and cannot be transported between different operating systems and computer platforms without being converted again.

You can use the [FASTLOAD command \[Page 51\]](#) and the [DATALOAD command \[Page 47\]](#) to load external binary values into database tables.



```
DATALOAD TABLE hotel
  hno      01-04  INTEGER
  name     09-18
  zip      20-25  DECIMAL
  city     27-36
  price    41-44  REAL
INFILE 'hotel.data' FORMATTED
```

## Prerequisites

You must specify the source data stream in the load command with either the [FORMATTED \[Page 111\]](#) format or the [FORMATTED BINARY \[Page 111\]](#) format. The data stream must be coded in binary.



## Loading LONG Values

In a command for loading application data, use the syntax rule [longfile\\_spec \[Page 81\]](#) in [longfile\\_spec\\_mlt \[Page 81\]](#) to specify for each column with the [internal database data type \[Page 114\]](#) LONG the name and path of the [data stream \[Page 108\]](#) from which you want to load the LONG values.

When you load LONG values, you can only use the `LONGFILE <valCOLUMN_NAME>` '`<valFILE_NAME>`' variant of the syntax rule, since you can only use column names in the load command, and no column IDs (see: [column\\_descriptor \[Page 66\]](#)).

When you load LONG values, you must differentiate between the following cases:

- [Each LONG Value to Be Inserted in a Separate LONG Data Stream \[Page 119\]](#)

- [All LONG Values to Be Inserted in One LONG Data Stream \[Page 120\]](#)

## Each LONG Value to Be Inserted in a Separate LONG Data Stream

Each LONG value to be inserted into a column is in a separate [data stream \[Page 108\]](#).

In the data stream, you specify the names of the data streams that contain the LONG values instead of the column values (if necessary, also entering position specifications). The entire data streams or parts of them are loaded as LONG values.



Complete data stream is loaded as LONG values

```
DATALOAD TABLE sqltravel10.hotel
  hno      1
  name     2
  info    3
INFILE 'hotel.data'
```

Content of the data stream:

```
10,Excelsior, 'EXCELSIOR.LNG'
30,Flora, 'FLORA.LNG'
60,Bellevue, 'BELLEVUE.LNG'
```



Parts of the data stream are loaded as LONG values.

```
DATALOAD TABLE sqltravel10.hotel
  hno      1
  name     2
  info    3
INFILE 'hotel.data'
```

Content of the data stream:

```
10,Excelsior, 'EXCELSIOR.LNG' 1-880
30,Flora, 'FLORA.LNG' 8-1046
60,Bellevue, 'BELLEVUE.LNG' 100-260
```

## Specifying a Code Attribute

You can optionally specify a code attribute in the load command for all data streams that contain LONG values that are to be loaded. You can only enter a single code attribute for all data streams. You cannot specify a code attribute for each individual specified data stream.

[Code Attribute for a Data Stream with LONG Values \[Page 121\]](#)



Specifying the Code Attribute ASCII

```
DATALOAD TABLE sqltravel10.hotel
  hno      1
  name     2
  info    3
INFILE 'hotel.data'
LONGFILE info ASCII
```

Content of the data stream:

```
10,Excelsior,'EXCELSIOR.LNG'
30,Flora,'FLORA.LNG'
60,Bellevue,'BELLEVUE.LNG'
```



No media name can be used after the keyword LONGFILE.

#### See also:

[Loading LONG Values \[Page 118\]](#)

[All LONG Values to Be Inserted in One LONG Data Stream \[Page 120\]](#)



## All LONG Values to Be Inserted in One LONG Data Stream

All LONG values to be inserted into a column are in one [data stream \[Page 108\]](#).

### Variant 1

In the data stream, you specify the name of the data stream containing the LONG values with the start and end position of the LONG value to be loaded, instead of a column value.



```
DATALOAD TABLE sqltravell10.hotel
  hno      1
  name     2
  info     3
INFILE 'hotel.data'
```

Content of the data stream:

```
10,Excelsior,'HOTEL.LNG' 1-880
30,Flora,'HOTEL.LNG' 881-1046
60,Bellevue,'HOTEL.LNG' 1047-1360
```

### Variant 2

You specify the name of the data stream that contains the LONG values after the keyword **LONGFILE**. In the data stream itself, specify only the position specifications without a preceding data stream name.



```
DATALOAD TABLE sqltravell10.hotel
  hno      1
  name     2
  info     3
INFILE 'hotel.data'
LONGFILE info 'HOTEL.LNG'
```

Content of the data stream:

```
10,Excelsior,1-880
30,Flora,881-1046
60,Bellevue,1047-1360
```

### Specifying a Code Attribute for the Media Specified with LONGFILE

In the load command, you can optionally specify a code attribute for data streams specified with LONGFILE.

[Code Attribute for a Data Stream LONG Values \[Page 121\]](#)


### Specifying the Code Attribute ASCII

```

DATALOAD TABLE sqltravell10.hotel
  hno      1
  name     2
  info    3
INFILE 'hotel.data'
LONGFILE info 'HOTEL.LNG' ASCII

```

### Content of the data stream:

```

10,Excelsior,1-880
30,Flora,881-1046
60,Bellevue,1047-1360

```

### See also:

[Loading LONG Values \[Page 118\]](#)
[Each LONG Value to Be Inserted in a Separate LONG Data Stream \[Page 119\]](#)


## Code Attribute for a Data Stream with LONG Values

If you do not specify a code attribute in the command, the SAP DB Loader uses a default value for the code attribute of the data stream. This is derived from the [internal database data type \[Page 114\]](#) of the column to be loaded. The following conversion table is used:

Internal Database Data Type	External Code Attribute
LONG ASCII	ASCII
LONG BYTE	BINARY
LONG UNICODE	UCS2

If the external code attribute and the internal database data type are different, the SAP DB Loader converts the data. If the types are incompatible, the SAP DB Loader generates an error message and stops processing the command.

### Compatibility Table

External Code Attribute	Internal Database Data Type	Compatible yes/no
ASCII	LONG ASCII	yes
	LONG BYTE	yes
	LONG UNICODE	yes
UCS2	LONG ASCII	no
	LONG BYTE	yes
	LONG UNICODE	yes
BINARY	LONG ASCII	no
	LONG BYTE	no
	LONG UNICODE	yes

## Use

[Each LONG Value to Be Inserted in a Separate LONG Data Stream \[Page 119\]](#)
[All LONG Values to Be Inserted in One LONG Data Stream \[Page 120\]](#)



## Medium

The SAP DB Loader uses a medium to physically store the [data stream \[Page 108\]](#) outside the database. If you load or unload [data \[Page 108\]](#) with the SAP DB Loader, the data source and data target are SAP DB databases and media.

The SAP DB Loader differentiates between the media types specified in [mediumtype\\_spec \[Page 82\]](#).



## Metadata

The SAP DB database system manages its metadata in the [database catalog](#).

For the SAP DB Loader, metadata is data that mirrors the database catalog information and also describes the data, the [application data \[Page 108\]](#), database catalog information and load and unload processes ([Transformations \[Page 10\]](#)) in accordance with the requirements of the SAP DB Loader.

- [Metadata of the Application Data \[Page 122\]](#)
- [Metadata of the Database Catalog \[Page 122\]](#)
- [Metadata of the Transformations \[Page 123\]](#)



## Metadata of the Application Data

The [metadata \[Page 122\]](#) of the [application data \[Page 108\]](#) (Metadata of the type DATA) are defined in the [data streams \[Page 108\]](#) using [SAP DB Loader commands \[Page 42\]](#) and [SQL statements \[Page 13\]](#).

The SAP DB Loader stores this information in the [system tables \[Page 37\]](#), if the [transformation \[Page 10\]](#) spans multiple tables.

The corresponding data is stored in the SAP DB database instance in the form of pages or rows and in a data stream in the non-readable formats [PAGES \[Page 112\]](#) and [RECORDS \[Page 112\]](#) or in the readable formats [FORMATTED \[Page 111\]](#), [COMPRESSED \[Page 109\]](#) and LONG.

If the data are transformed to the format of rows, columns of the [external data type \[Page 114\]](#) CHAR with the code attribute ASCII can be transformed to values of data fields with the code attribute ASCII, UCS-2 or UTF-8.



## Metadata of the Database Catalog

[Metadata \[Page 122\]](#) of the [database catalog](#) (Metadata of the type CATALOG) describe the structure of the data in the database catalog.

The SAP DB Loader stores this information in its [system tables \[Page 37\]](#), if the [transformation \[Page 10\]](#) spans multiple tables.

The corresponding data is stored in the SAP DB database instance in the form of rows in the system tables and in a [data stream \[Page 108\]](#) in the form of SQL statements for data definition.



## Metadata of the Transformations

The [metadata \[Page 122\]](#) of the [transformations \[Page 10\]](#) (Metadata of types CONFIGURATION and PACKAGE) are declared in the form of [commands \[Page 42\]](#) or in the form of SAP DB Loader-specific [system tables \[Page 37\]](#).

This metadata describes the configuration and the runtime behavior of the transformations that span multiple tables.

The corresponding data, such as user and table names, is stored, sorted by referential constraint conditions.

In the SAP DB database instance, the data is stored in the form of rows in the system tables and in a [data stream \[Page 108\]](#) in the readable format [COMPRESSED \[Page 109\]](#).



## Selecting Data Records

When you load [application data \[Page 108\]](#), you can select the data records according to their contents.

The conditions defined here can be negated with **NOT**, linked with **AND** and **OR**, or encapsulated as required. The SAP DB Loader evaluates the operators accordingly.

### See also:

[condition \[Page 70\]](#)

[if\\_condition \[Page 76\]](#)



You want to load only those data records from the [source data stream \[Page 108\]](#) `hotel.data` into the target table `hotel` that satisfy the following conditions:

The price in the data record is less than 400.00 (**IF**).

The name of the hotel is not `City` (**NOT**).

The hotel is located in `BERLIN` (**AND**).

The zip code is 13125 or 13126 (**OR**):

```
DATALOAD TABLE hotel
  IF POS 41-44 REAL < '400.00'
  AND
    POS 27-36 = 'BERLIN'
  AND
    (POS 20-25 = '13125' OR POS 20-25 = '13126')
  AND NOT
    POS 09-18 = 'City'
  hno      01-04 INTEGER
  name     09-18
  zip      20-25 DECIMAL
  city     27-36
  price    41-44 REAL
INFILE 'hotel.data' FORMATTED
```



You want to load only those data records from the source file `hotel.data` into the target table `hotel` where the price is not more than 400.00:

```

DATALOAD TABLE hotel
  IF NOT POS 41-44 REAL > '400.00'
    hno      01-04 INTEGER
    name     09-18
    zip      20-25 DECIMAL
    city     27-36
    price    41-44 REAL
  INFILE 'hotel.data' FORMATTED

```



## System Table CODEPAGE

The system table CODEPAGE contains the conversion tables that are used by the SAP DB Loader for the conversion.

### Table Structure

```

CREATE TABLE SYSDBA.CODEPAGE
  (CODEPAGEID CHAR (32) NOT NULL,
   CODEPAGETABLE CHAR (512) BYTE NOT NULL,
   PRIMARY KEY (CODEPAGEID))

```

Column	Explanation
CODEPAGEID	Name of code pages (ISO-8859-2, for example)
CODEPAGETABLE	Conversion table from ASCII to UCS2 for each code page

### Content

The system table CODEPAGE is preconfigured. The CODEPAGE column contains 27 entries when it is delivered:

ISO-8859-2, ISO-8859-3, ISO-8859-4, ISO-8859-5, ISO-8859-6, ISO-8859-7, ISO-8859-8, ISO-8859-9, ISO-8859-10, ISO-8859-11, ISO-8859-13, ISO-8859-14, ISO-8859-15, ISO-8859-16, WINDOWS-874, WINDOWS-932, WINDOWS-936, WINDOWS-949, WINDOWS-950, WINDOWS-1250, WINDOWS-1251, WINDOWS-1252, WINDOWS-1253, WINDOWS-1254, WINDOWS-1255, WINDOWS-1256, WINDOWS-1257.

The default value of the SAP DB Loader (code page ISO-8859-1) is **not** in the table.

### Use

See [code\\_page\\_spec \[Page 63\]](#)