

UNLOAD format

```
UNLOAD
[ CHARACTERSET { cset_name | ccsid } ]
[ OUTFILE { sth_file_name } [ /group ] ]
[ FIELDS [ CHAR ] [ NULLFLAGS ] delimiter_spec ]
[ RECORDS TERMINATED [BY] {WHITESPACE|'char'|X'hexbyte'}]
[ USING ( field_spec [ , field_spec ]... ) ]
query ;
```

CHARACTERSET specifies the character set of result data

cset_name	WE8EBCDIC500 WE8PC850 WE8ISO8859P1
ccsid	500 850 819

OUTFILE unloads data to a StorHouse VRAM file

sth_file_name	StorHouse VRAM file name
/group	File access group name

FIELDS formats result data

CHAR	All result data fields are to be formatted as CHARACTER
NULLFLAGS	NULL flags are to be inserted at the beginning of each result row
delimiter_spec	[TERMINATED [BY] {WHITESPACE 'char' X'hexbyte'}] [[OPTIONALLY] ENCLOSED [BY] {'char' X'hexbyte'} [AND {'char' X'hexbyte'}]]

RECORDS appends a character to the end of records

TERMINATED	Required keyword
BY	Optional keyword
WHITESPACE	Keyword to terminate a record with a blank in the character set of the result data
'char'	Value of the record terminator, expressed in character format
X'hexbyte'	Value of the record terminator, expressed in hexadecimal format

USING describes each data field in result records

field_spec	position_spec constant_spec
position_spec	datatype_spec [POSITION (position * [+ num])] [IFNULL field_assignment]
datatype_spec	Name and length of the data field's data type
position	start_column [{ : - } end_column]
num	Unsigned integer optionally followed by K (x1024) or M (xKK)
field_assignment	[:field_name (position)] = { any_string BLANKS }
:field_name	Name of the data field in the result record
any_string	string X string
constant_spec	[:field_name] CONSTANT any_value
any_value	any_string num

query selects the StorHouse data to unload

```
SELECT [ALL | DISTINCT]
{* | expr [column_alias] [, expr [column_alias] ]...}
[FROM table_spec [, table_spec ]...]
[WHERE condition]
[GROUP BY column_name [,column_name]... [HAVING condition] ]
[ {UNION | UNION ALL} SELECT ...]
[ORDER BY {expr | position} [ASC | DESC] [, {expr | position} [ASC | DESC] ]... ]
[FOR {FETCH | READ} ONLY]
```

UNLOAD examples

The following examples show how to unload data from a table called INVENTORY. Here's the CREATE TABLE statement used to create the INVENTORY table:

```
CREATE TABLE INVENTORY
(Item CHAR(10),
Quantity INTEGER,
Status VARCHAR(10))
```

Here's sample data in the INVENTORY table (where – is NULL):

Item	Quantity	Status
plug	12	–
cable	–	out

Creating fixed-length fields and records:

```
UNLOAD
USING (CHAR(10), INTEGER EXTERNAL(10), CHAR(10))
SELECT * FROM INVENTORY;
```

Result data (b represents a blank):

```
plugbbbbbbbbbbbbbb12bbbbbbbbbb
cablebbbbbbbbbbbbbboutbbbbbb
```

Terminating data fields and records, adding keywords next to each data field:

```
UNLOAD
FIELDS TERMINATED BY ','
RECORDS TERMINATED BY ';'
USING (CONSTANT 'Item=', CHAR,
CONSTANT 'Quantity=', CHAR,
CONSTANT 'Status=', CHAR)
SELECT * FROM INVENTORY;
```

Result data (b represents a blank):

```
Item=plug,Quantity=12,Status=;
Item=cable,Quantity=b,Status=out;
```

UNLOAD

Quick Reference

StorHouse/RM Release 3.2

This *Quick Reference* contains the format and examples of the UNLOAD statement used to unload data from StorHouse® user tables. Refer to the *FileTek FTP Data Unloader Manual* for complete information about the UNLOAD statement.

Proprietary and Confidential

