

ACCEPT/REJECT

<pre> { ACCEPT } [IF] logical-condition { REJECT }</pre>
--

This chapter covers the following topics:

- Function
- Syntax Description
- Processing of Multiple ACCEPT/REJECT Statements
- Limit Notation
- Examples

For an explanation of the symbols used in the syntax diagram, see *Syntax Symbols*.

Related Statements: AT BREAK | AT START OF DATA | AT END OF DATA | BACKOUT TRANSACTION | BEFORE BREAK PROCESSING | DELETE | END TRANSACTION | FIND | HISTOGRAM | GET | GET SAME | GET TRANSACTION DATA | LIMIT | PASSW | PERFORM BREAK PROCESSING | READ | RETRY | STORE | UPDATE

Belongs to Function Group: *Database Access and Update*

Function

The statements ACCEPT and REJECT are used for accepting/rejecting a record based on user-specified logical criterion. The ACCEPT/REJECT statement may be used in conjunction with statements which read data records in a processing loop (FIND, READ, HISTOGRAM, CALL FILE, SORT or READ WORK FILE). The criterion is evaluated *after* the record has been selected/read.

Whenever an ACCEPT/REJECT statement is encountered for processing, it will internally refer to the innermost currently active processing loop initiated with one of the above mentioned statements.

When ACCEPT/REJECT statements are placed in a subroutine, in case of a record reject, the subroutine(s) entered in the processing loop will automatically be terminated and processing will continue with the next record of the innermost currently active processing loop.

Syntax Description

Syntax Element	Description
IF	<p>IF Clause: An IF clause may be used with an ACCEPT or REJECT statement to specify logical condition criteria in addition to that specified when the record was selected/read with a FIND, READ, or HISTOGRAM statement. The logical condition criteria are evaluated after the record has been read and after record processing has started.</p>
<i>logical-condition</i>	<p>Logical Condition Criterion: The basic criterion is a relational expression. Multiple relational expressions may be combined with logical operators (AND, OR) to form complex criteria.</p> <p>Arithmetic expressions may also be used to form a relational expression.</p> <p>The fields used to specify the logical criterion may be database fields or user-defined variables. For additional information on logical conditions, see <i>Logical Condition Criteria</i> in the <i>Programming Guide</i>.</p> <p>Note: When ACCEPT/REJECT is used with a HISTOGRAM statement, only the database field specified in the HISTOGRAM statement may be used as a logical criterion.</p>

Processing of Multiple ACCEPT/REJECT Statements

Normally, only one ACCEPT or REJECT statement is required in a single processing loop. If more than one ACCEPT/REJECT is specified *consecutively*, the following conditions apply:

- If consecutive ACCEPT and REJECT statements are contained in the same processing loop, they are processed in the specified order.
- If an ACCEPT condition is satisfied, the record will be accepted and consecutive ACCEPT/REJECT statements will be ignored.
- If a REJECT condition is satisfied, the record will be rejected and consecutive ACCEPT/REJECT statements will be ignored.
- If the processing continues to the last ACCEPT/REJECT statement, the last statement will determine whether the record is accepted or rejected.

If other statements are interleaved between multiple ACCEPT/REJECT statements, each ACCEPT/REJECT will be handled independently.

Limit Notation

If a LIMIT statement or other limit notation has been specified for a processing loop containing an ACCEPT or REJECT statement, each record processed is counted against the limit regardless of whether or not the record is accepted or rejected.

Examples

- Example 1 - ACCEPT
- Example 2 - ACCEPT / REJECT

Example 1 - ACCEPT

```

** Example 'ACREX1': ACCEPT
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 SEX
  2 MAR-STAT
END-DEFINE
*
LIMIT 50
READ EMPLOY-VIEW
  ACCEPT IF SEX='M' AND MAR-STAT = 'S'
  WRITE NOTITLE '=' NAME '=' SEX 5X '=' MAR-STAT
END-READ
END

```

Output of Program ACREX1:

```

NAME: MORENO           S E X: M      MARITAL STATUS: S
NAME: VAUZELLE        S E X: M      MARITAL STATUS: S
NAME: BAILLET         S E X: M      MARITAL STATUS: S
NAME: HEURTEBISE     S E X: M      MARITAL STATUS: S
NAME: LION            S E X: M      MARITAL STATUS: S
NAME: DEZELUS        S E X: M      MARITAL STATUS: S
NAME: BOYER          S E X: M      MARITAL STATUS: S
NAME: BROUSSE        S E X: M      MARITAL STATUS: S
NAME: DROMARD        S E X: M      MARITAL STATUS: S
NAME: DUC            S E X: M      MARITAL STATUS: S
NAME: BEGUERIE       S E X: M      MARITAL STATUS: S
NAME: FOREST         S E X: M      MARITAL STATUS: S
NAME: GEORGES        S E X: M      MARITAL STATUS: S

```

Example 2 - ACCEPT / REJECT

```

** Example 'ACREX2': ACCEPT/REJECT
*****
DEFINE DATA LOCAL
1 EMPLOY-VIEW VIEW OF EMPLOYEES
  2 NAME
  2 FIRST-NAME
  2 SALARY      (1)
*
1 #PROC-COUNT (N8) INIT <0>
END-DEFINE
*
EMP. FIND EMPLOY-VIEW WITH NAME = 'JACKSON'
  WRITE NOTITLE *COUNTER NAME FIRST-NAME 'SALARY:' SALARY(1)
  /*
  ACCEPT IF SALARY (1) LT 50000
  WRITE *COUNTER 'ACCEPTED FOR FURTHER PROCESSING'
  /*
  REJECT IF SALARY (1) GT 30000

```

```
WRITE *COUNTER 'NOT REJECTED'  
/*  
ADD 1 TO #PROC-COUNT  
END-FIND  
*  
SKIP 2  
WRITE NOTITLE 'TOTAL PERSONS FOUND ' *NUMBER (EMP.) /  
              'TOTAL PERSONS SELECTED' #PROC-COUNT  
END
```

Output of Program ACREX2:

1 JACKSON	CLAUDE	SALARY:	33000
1 ACCEPTED FOR FURTHER PROCESSING			
2 JACKSON	FORTUNA	SALARY:	36000
2 ACCEPTED FOR FURTHER PROCESSING			
3 JACKSON	CHARLIE	SALARY:	23000
3 ACCEPTED FOR FURTHER PROCESSING			
3 NOT REJECTED			

TOTAL PERSONS FOUND	3
TOTAL PERSONS SELECTED	1