

DECIDE FOR

```
DECIDE FOR { FIRST } CONDITION
           { EVERY }
           { WHEN logical-condition statement ... } ...
           [ WHEN ANY statement ... ]
           [ WHEN ALL statement ... ]
           WHEN NONE statement ...
END-DECIDE
```

This chapter covers the following topics:

- Function
- Syntax Description
- Examples

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

Related Statements: DECIDE ON | IF | IF SELECTION | ON ERROR

Belongs to Function Group: *Processing of Logical Conditions*

Function

The DECIDE FOR statement is used to decide for one or more actions depending on multiple conditions (cases).

Note:

If *no* action is to be performed under a certain condition, you must specify the statement IGNORE in the corresponding clause of the DECIDE FOR statement.

Syntax Description

| | |
|--|---|
| FIRST CONDITION | Only the first true condition is to be processed. See also <i>Example 1</i> . |
| EVERY CONDITION | Every true condition is to be processed. See also <i>Example 2</i> . |
| WHEN <i>logical-condition statement</i> | With this clause, you specify the logical condition(s) to be processed. See the section <i>Logical Condition Criteria</i> in the <i>Programming Guide</i> . |
| WHEN ANY <i>statement</i> | With WHEN ANY, you can specify the statement(s) to be executed when any of the logical conditions are true. |
| WHEN ALL <i>statement</i> | With WHEN ALL, you can specify the statement (s) to be executed when all logical conditions are true. This clause is applicable only if EVERY has been specified. |
| WHEN NONE <i>statement</i> | With WHEN NONE, you specify the statement(s) to be executed when none of the logical conditions are true. |
| END-DECIDE | The Natural reserved word END-DECIDE must be used to end the DECIDE FOR statement. |

Examples

- Example 1 - DECIDE FOR with FIRST Option
- Example 2 - DECIDE FOR with EVERY Option

Example 1 - DECIDE FOR with FIRST Option

```

** Example 'DECEX1': DECIDE FOR (with FIRST option)
*****
DEFINE DATA LOCAL
1 #FUNCTION (A1)
1 #PARM (A1)
END-DEFINE
*
INPUT #FUNCTION #PARM
*
DECIDE FOR FIRST CONDITION
  WHEN #FUNCTION = 'A' AND #PARM = 'X'
    WRITE 'Function A with parameter X selected.'
  WHEN #FUNCTION = 'B' AND #PARM = 'X'
    WRITE 'Function B with parameter X selected.'
  WHEN #FUNCTION = 'C' THRU 'D'
    WRITE 'Function C or D selected.'
  WHEN NONE
    REINPUT 'Please enter a valid function.'
    MARK *#FUNCTION
END-DECIDE
*
END

```

Output of Program DECEX1:

```
#FUNCTION A #PARM Y
```

After pressing ENTER:

```
PLEASE ENTER A VALID FUNCTION
#FUNCTION A #PARM Y
```

Example 2 - DECIDE FOR with EVERY Option

```
** Example 'DECEX2': DECIDE FOR (with EVERY option)
*****
DEFINE DATA LOCAL
1 #FIELD1 (N5.4)
END-DEFINE
*
INPUT #FIELD1
*
DECIDE FOR EVERY CONDITION
  WHEN #FIELD1 >= 0
    WRITE '#FIELD1 is positive or zero.'
  WHEN #FIELD1 <= 0
    WRITE '#FIELD1 is negative or zero.'
  WHEN FRAC(#FIELD1) = 0
    WRITE '#FIELD1 has no decimal digits.'
  WHEN ANY
    WRITE 'Any of the above conditions is true.'
  WHEN ALL
    WRITE '#FIELD1 is zero.'
  WHEN NONE
    IGNORE
END-DECIDE
*
END
```

Output of Program DECEX2:

```
#FIELD1 42
```

After pressing ENTER:

```
Page      1
```

```
05-01-11 14:56:26
```

```
#FIELD1 is positive or zero.
#FIELD1 has no decimal digits.
Any of the above conditions is true.
```