DECIDE ON

```
DECIDE ON  { FIRST }  [VALUES] [OF] operand1
           { EVERY }

 {VALUES operand2 [[,operand2] ... [:operand2]] statement ...}

 [ANY [VALUES]  statement ... ]

 [ALL [VALUES]  statement ... ]

  NONE [VALUES] 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 FOR | IF | IF SELECTION | ON ERROR

Belongs to Function Group: *Processing of Logical Conditions*

Function

The `DECIDE ON` statement is used to specify multiple actions to be performed depending on the value (or values) contained in a variable.

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 ON` statement.

Syntax Description

Operand Definition Table:

Operand	Possible Structure					Possible Formats													Referencing Permitted	Dynamic Definition
operand1		S	A		N	A	U	N	P	I	F	B	D	T	L		G	O	yes	no
operand2	C	S	A			A	U	N	P	I	F	B	D	T	L		G	O	yes	no

* Format B of *operand5*, *operand6*, *operand7* and *operand8* may be used only with a length of less than or equal to 4.

Syntax Element Description:

FIRST/EVERY	With one of these keywords, you indicate whether only the first or every value that is found is to be processed.
operand1	**Selection Field:** As *operand1* or *operand2* you specify the name of the field whose content is to be checked.
VALUES *operand2* [[,*operand2*] ... [:*operand2*]*statement* ...	With this clause, you specify the value (*operand2*) of the selection field, as well as the *statement(s)* which are to be executed if the field contains that value. You can specify one value, multiple values, or a range of values optionally preceded by one or more values. Multiple values must be separated from one another either by the input delimiter character (as specified with the session parameter `ID`) or by a comma. A comma must not be used for this purpose, however, if the comma is defined as decimal character (with the session parameter `DC`). For a range of values, you specify the starting value and ending value of the range, separated from each other by a colon.
ANY *statement*	With `ANY`, you specify the *statement(s)* which are to be executed if any of the values in the `VALUES` clause are found. These statements are to be executed in addition to the statement specified in the `VALUES` clause.

ALL ***statement***	With ALL, you specify the *statement(s)* which are to be executed if all of the values in the VALUES clause are found. These statements are to be executed in addition to the statement specified in the VALUES clause. The ALL clause applies only if the keyword EVERY is specified.
NONE ***statement***	With NONE, you specify the *statement(s)* which are to be executed if none of the specified values are found.
END-DECIDE	The Natural reserved word END-DECIDE must be used to end the DECIDE ON statement.

Examples

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

Example 1 - DECIDE ON with FIRST Option

```
** Example 'DECEX3': DECIDE ON (with FIRST option)
************************************************************************
*
SET KEY ALL
INPUT 'Enter any PF key' /
      'and check result' /
*
DECIDE ON FIRST VALUE OF *PF-KEY
  VALUE 'PF1'
    WRITE 'PF1 key entered.'
  VALUE 'PF2'
    WRITE 'PF2 key entered.'
  ANY VALUE
    WRITE 'PF1 or PF2 key entered.'
  NONE VALUE
    WRITE 'Neither PF1 nor PF2 key entered.'
END-DECIDE
*
END
```

Output of Program DECEX3:

```
 Enter any PF key
 and check result
```

Output after pressing PF1:

```
Page      1                                                  05-01-11  15:08:50

PF1 key entered.
PF1 or PF2 key entered.
```

Example 2 - DECIDE ON with EVERY Option

```
** Example 'DECEX4': DECIDE ON (with EVERY option)
************************************************************************
DEFINE DATA LOCAL
1 #FIELD (N1)
END-DEFINE
*
INPUT 'Enter any value between 1 and 9:' #FIELD (SG=OFF)
*
DECIDE ON EVERY VALUE OF #FIELD
  VALUE 1 : 4
    WRITE 'Content of #FIELD is 1-4'
  VALUE 2 : 5
    WRITE 'Content of #FIELD is 2-5'
  ANY VALUE
    WRITE 'Content of #FIELD is 1-5'
  ALL VALUE
    WRITE 'Content of #FIELD is 2-4'
  NONE VALUE
    WRITE 'Content of #FIELD is not 1-5'
    END-DECIDE
*
END
```

Output of Program DECEX4:

```
ENTER ANY VALUE BETWEEN 1 AND 9: 4
```

After entering and confirming 4:

```
Page      1                                                  05-01-11  15:11:45

Content of #FIELD is 1-4
Content of #FIELD is 2-5
Content of #FIELD is 1-5
Content of #FIELD is 2-4
```