

Case Statement

Description

Provides a framework for conditional branching. Evaluates one or more test expressions, then executes the relevant code at the corresponding internal subroutine label. A Case statement must be introduced with **Begin Case** and terminated with **End Case**.

Syntax

Begin Case [**Case** expression statements ...] **End Case**

Parameters

The Case statement has the following parameters.

Parameter	Description
<i>Expression</i>	<p>If <i>expression</i> evaluates to true the statements are executed, and no subsequent Case statements are evaluated. Program control then proceeds past End Case.</p> <p>When used with Case, an expression of 1 (one) always equates to true. It can be used to specify unconditional execution of the statements. If used in this way, this should be the last Case statement. This Case will always be executed, if none of the other Case statements evaluate to true.</p>
<i>Statements</i>	Any valid BASIC+ statement or statements.

See also

[If](#), [On...GoSub](#), [On...GoTo](#)

Example

```
$insert Logical
/* The program control branches to statement label EQUAL if A is equal to B,
to label LESSTHAN if A is less than B,
and to NEITHER if neither of the above is true. */
Begin Case
  Case A EQ B
    GoSub EQUAL
  Case A LT B
    GoSub LESSTHAN
  Case OTHERWISE$
    GoSub NEITHER
End Case
/* If the inventory quantity is less than 10, the program calls the subroutine EXPEDITE.
If QTY is more than or equal to 10 but less than 100, the program calls the subroutine REORDER.
Otherwise the program continues without calling either subroutine. */
Begin Case
  Case QTY LT 10
    Call EXPEDITE
  Case QTY LT 100
    Call REORDER
End Case
```