

How do I use conditional compilation?

Author: Veryant Support

Saved From: <http://support.veryant.com/support/phpkb/question.php?ID=27>

Question:

How do I use compiler directives to conditionally include or exclude sections of code?

Answer:

Here are some examples of conditional compilation:Â

EVALUATE and WHEN in a bug test program:

```
id division.
program-id. chinese.
data division.
working-storage section.
>> EVALUATE CASE
>> WHEN "TEST-1"
  77 chinese-item pic n(2) value nx"4EAC".
>> WHEN "TEST-2"
  77 chinese-item pic n(1) value nx"4EAC".
>> WHEN "TEST-3"
  77 chinese-item pic n(1) usage national.
>> END-EVALUATE
procedure division.
main-logic.
  display standard graphical window.
  display chinese-item.
  accept omitted.
```

Compile with:

```
java -Discobol.compiler.const.CASE=TEST-1 com.iscobol.compiler.Pcc chinese.cbl
to include the TEST-1 case.
```

An alternative would be to add the following to the top of chinese.cbl:

```
>> DEFINE CASE AS "TEST-3"
```

and write the program as:

```
id division.
program-id. chinese.
data division.
working-storage section.
>> IF CASE IS = "TEST-1"
  77 chinese-item pic n(2) value nx"4EAC".
>> ELSE
>> IF CASE IS = "TEST-2"
  77 chinese-item pic n(1) value nx"4EAC".
```

```

>> ELSE
>> IF CASE IS = "TEST-3"
    77 chinese-item pic n(1) usage national.
>> END-IF
>> END-IF
>> END-IF
procedure division.
main-logic.
    display standard graphical window.
    display chinese-item.
    accept omitted.

```

In the above, CASE is a preprocessor constant set to either "TEST-1", "TEST-2" or "TEST-3". The program could also be written to just test whether a constant is defined, as follows:

```

id division.
program-id. chinese.
data division.
working-storage section.
>> IF TEST-1 IS DEFINED
    77 chinese-item pic n(2) value nx"4EAC".
>> ELSE
>> IF TEST-2 IS DEFINED
    77 chinese-item pic n(1) value nx"4EAC".
>> ELSE
>> IF TEST-3 IS DEFINED
    77 chinese-item pic n(1) usage national.
>> END-IF
>> END-IF
>> END-IF
procedure division.
main-logic.
    display standard graphical window.
    display chinese-item.
    accept omitted.

```

and then compiled with

```
java -Discobol.compiler.const.TEST-1=1 com.iscobol.compiler.Pcc chinese.cbl
```

Here is an excerpt from the isCOBOL APS User Guide (which is available from the Windows Start menu or at [this link](#)).

Compiler directives

The compiler directives can alter the behavior of the compiler.

DEFINE directive:

The DEFINE directive allows to define a compiler constant.

```
>> DEFINE ConstantName AS ContantValue [OVERRIDE]
```

Syntax

1. ConstantName is the name of the constant to be set.
2. ConstantValue is the value of the constant.

General rules

1. The Constant is defined and set to the ConstantValue.
2. Use the OVERRIDE clause to re-define a previously set constant.

ERROR Directive:

When the ERROR directive is encountered, a message is written to the STDERR and the compiler exits.

```
>> ERROR String
```

Syntax

1. String is a text string delimited by quotes.

EVALUATE directive:

The EVALUATE directive allows to check the value of a constant to include or exclude fragments of source code.

```
>> EVALUATE ConstantName
```

```
  { >> WHEN String Statements-1 } ...
  [ >> WHEN OTHER Statements-2 ]
```

```
>> END-EVALUATE
```

Syntax

1. ConstantName is a constant defined in the configuration file as iscobol.compiler.ConstantName or using >> DEFINE compiler directive.
2. String is a text string delimited by quotes.
3. Statements-1 e Statements-2 are lines of COBOL code.

General rules

1. When String matches the value of ConstantName, lines in Statements-1 are included.
2. When no match is found, lines in Statements-2 are included.

IF directive:

The IF directive allows to check if a constant is set, to include or exclude fragments of source code.

```
>> IF ConstantName IS [ NOT ] DEFINED Statements-1
```

```
      {[ NOT ] = ConstantValue}
      {[ NOT ]           {[ NOT ] > ConstantValue}
```

```
  [ >> ELSE Statements-2 ]
```

```
>> END-IF
```

Syntax

1. ConstantName is a constant defined in the configuration file as iscobol.compiler.ConstantName or using >> DEFINE compiler directive.
2. Statements-1 e Statements-2 are lines of COBOL code.

General rules

1. Testing IF DEFINED, when ConstantName is defined, no matter the value, lines in Statements-1 are included. Otherwise, lines in Statements-2 are included.
2. Testing IF EQUAL, GREATER or LESS, when ConstantName is defined, and its value matches the IF condition, lines in Statements-1 are included. Otherwise, lines in Statements-2 are included.