



July 4, 2018

## [Invalid Data](#)

### Invalid Data residing in Numeric fields with DISPLAY usage

#### Definition of INVALID DATA for converting to COBOL 5 or 6

The format of NUMERIC fields in COBOL for character or DISPLAY usage consists of two HEX digits making up one NUMERIC character. For instance, the HEX format for the numbers 1, 2 and 3 are F1, F2, and F3.

For unsigned number **123**, the hex format stored in the computer is **F1F2F3**.

In older IBM compilers a check was done on the last or low order digit to see if that HEX format was either Fn, of Cn, or Dn, where D indicated a negative number for the entire field or was a C or F for a positive number.

**+1234 = F1F2C3** (The C indicating a forced positive number)

The other digits are all assumed (in older compilers) to be an **E** with no implicit check. That allowed for a blank character, hex 40 or other invalid combination to be there. See below example:

'1**E**3' = 'F1**40**F3' where the middle character is a blank

#### What problem can this cause

The newer compilers (5 and on), create different results from the older compilers when this is there. The way to not be surprised is to catch these errors.

#### Frequency of this problem

According to IBM, only 25% of shops or installations encounter this problem.

#### How to identify

Without using the solution offered by Control/DCD, each installation must find their own way of identifying these errors.

One possible way is to identify all occurrences of invalid numeric DISPLAY data is to list all the fields within INPUT and OUTPUT files in a COBOL program and use an easily modifiable program to go through each of these files and look invalid data in these fields. This would catch most of these errors.

## What is Control/DCD

Control/DCD is a software tool that does analysis on COBOL programs with the following features:

1. COBOL Analysis is created for reviewing on the mainframe or on the PC or other *device* as a PDF.
2. It is also designed to be used as a pre-compiler in place of extra COBOL compilers to avoid making extra compiles that take much longer.
3. It creates Warnings and Error message like the COBOL compiler.

## How to identify using Control/DCD

An option in Control/DCD within ISPF that turns on Error Messages on and off, with a Y or N. Using an **'I'** in place of a **'Y'** option, turns on additional messages that identify where INVALID DATA is created in several ways as shown below.

- If a bad non-numeric literal with an INVALID entry is in a REDEFINES to a numeric DISPLAY entry, this is flagged as such.
- If a bad non-numeric literal with an INVALID entry is MOVED to REDEFINES to a numeric DISPLAY entry, this is flagged as such.
- If a bad literal exists in a VALUE in the DATA DIVISION and is MOVED to another field that is a REDEFINES to a numeric DISPLAY entry, this also is flagged.
- If a bad literal is MOVED to a one field, that is later MOVED to another REDEFINES of a numeric DISPLAY field, then this is flagged.

With this option on, simply run Control/DCD on this program and then go to the Error Message generated and look for possible errors.

An example of these messages generated is shown below:

```
MSG-NBR      DESCRIPTION
CSEM6KC3-I   MOVE '12T44' may reach field at (0481) via MOVE at 01346
CSEM6IB2-I   MOVE HIGH-VALUES may be INVALID for field at 00153
```

With this option on, there are also other messages generated to look to ensure that all DISPLAY NUMERIC fields are initially assigned a VALUE and warns when a REDEFINES of NUMERIC fields have a value inconsistent with the REDEFINED Numeric DISPLAY field. See [DISPLAY Numeric used before assigned a VALUE](#)

*Want to know more? Contact us today. Call 1-800-252-1400 ext. #4 or send an email to [sales@marblecomputer.com](mailto:sales@marblecomputer.com)*