

Wissenswertes über binäre Felder

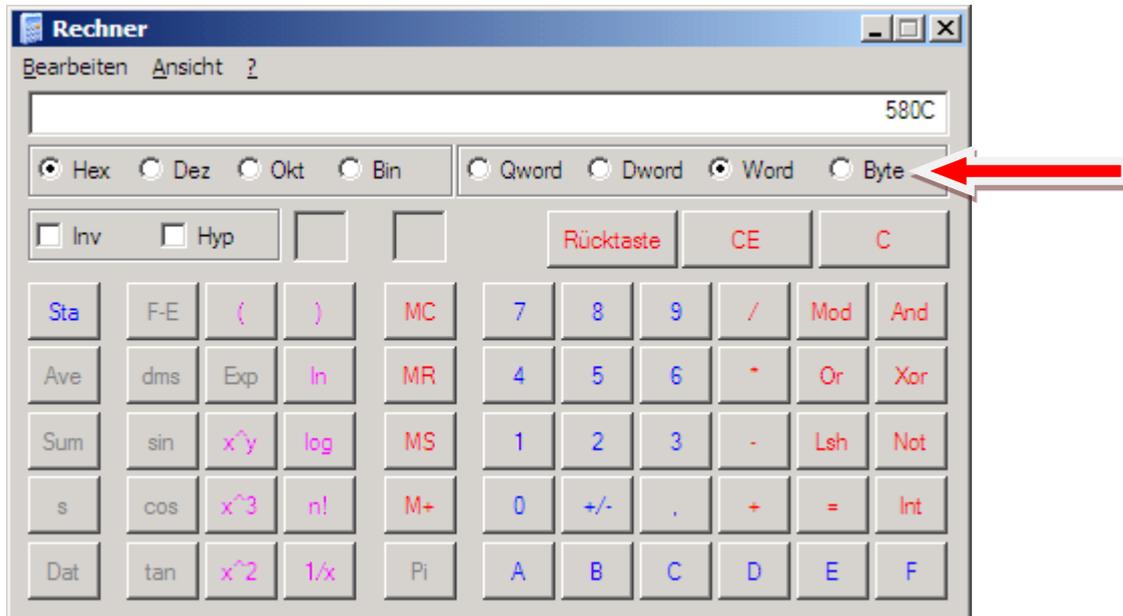
Inhaltsverzeichnis

Genauigkeit des PC-Taschenrechners_____	2
Genauigkeit des PC-Taschenrechners ab Windows 7 _____	2
Ausgangspunkt_____	3
Binäres Feld ohne Vorzeichen-Definition _____	3
Binäres Feld mit Vorzeichen-Definition_____	3
Der binäre Kreis – ohne Vorzeichen – 2 Byte Feld (PIC 9(4) binary) _____	4
Der binäre Kreis – ohne Vorzeichen – 2 Byte Feld (PIC S9(4) binary) _____	4
Umrechnen von (positiven) binären Feldern in dezimale Werte _____	5
Umrechnen von (positiven) binären Feldern in dezimale Werte ab Windows 7 _____	6
Umrechnen von (negativen) binären Feldern in dezimale Werte _____	7
Umrechnen von (negativen) binären Feldern in dezimale Werte ab Windows 7 _____	9
Umrechnen von positiven binären Feldern > 32767 in dezimale Werte ab Windows 7 (COBOL: PIC 9(04) binary)_____	10

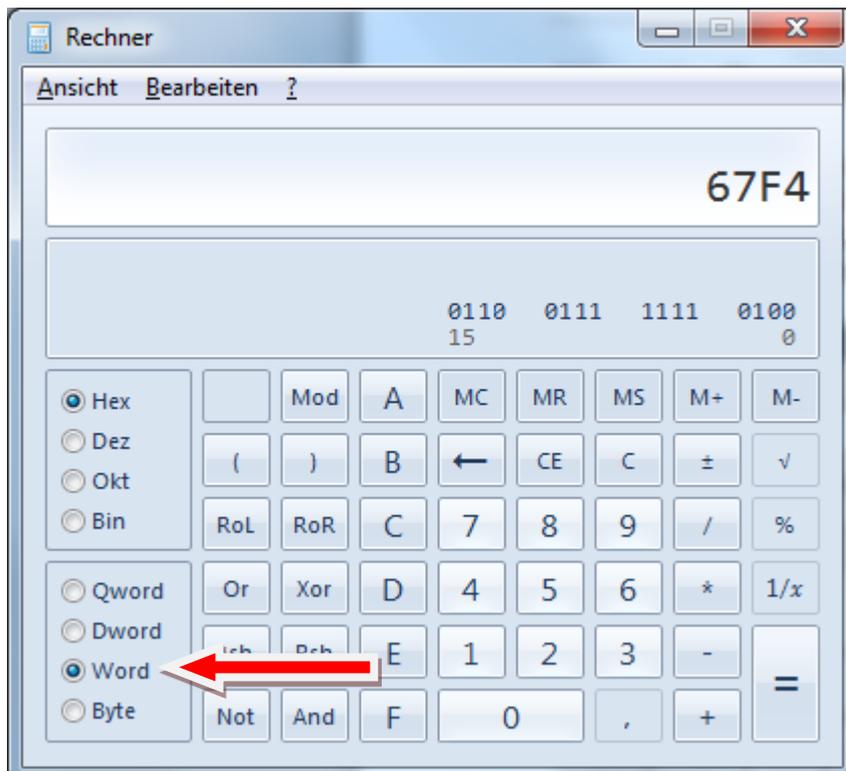
Genauigkeit des PC-Taschenrechners

PIC 9(4) oder PIC S9(4) ist ein 2-Byte langes Feld. Zwei Byte nennt man auch ein „Wort“.
PIC 9(8) oder PIC S9(8) ist ein 4-Byte langes Feld. Vier Byte nennt man auch „Doppelwort“.
PIC 9(18) oder PIC S9(18) ist ein 4-Byte langes Feld. Vier Byte nennt man auch „Quadwort“.

Auf dem Taschenrechner gibt es diese Einstellung auch:



Genauigkeit des PC-Taschenrechners ab Windows 7



Ausgangspunkt

F-BIN PIC 9(04) BINARY.

Das Feld belegt 2 Bytes.

F-BIN-99 PIC 9(02) BINARY

Das Feld belegt 2 Bytes.

Bei PIC 9(02) BINARY hat der Compiler zusätzliche Instruktionen, um das Feld künstlich zu „kürzen“. Sehr inperformant.

Binäres Feld ohne Vorzeichen-Definition

F-BIN PIC 9(04) BINARY.

Zahlenbereich bei 2 Bytes: 0 bis 65535

Darstellung: 0000 bis FFFF.

Beispiele:

$$0000_{(16)} = 0_{(10)}$$

$$0001_{(16)} = 1_{(10)}$$

$$0011_{(16)} = 17_{(10)}$$

$$324A_{(16)} = 12874_{(10)}$$

$$FFF0_{(16)} = 65520_{(10)}$$

Binäres Feld mit Vorzeichen-Definition

F-BIN PIC S9(04) BINARY.

Zahlenbereich bei 2 Bytes: -32767₍₁₀₎ bis -0₍₁₀₎ / +0₍₁₀₎ bis +32767₍₁₀₎

Darstellung: 8001₍₁₆₎ bis 8000₍₁₆₎ / 0000₍₁₆₎ bis 7FFF₍₁₆₎.

Also: das erste Bit wird zur Kennzeichnung des Vorzeichens benutzt.

$$8001_{(16)} + 1_{(16)} = 8002_{(16)} \quad / \quad -32767_{(10)} + 1_{(10)} = -32766_{(10)}$$

$$8002_{(16)} + 1_{(16)} = 8003_{(16)} \quad / \quad -32766_{(10)} + 1_{(10)} = -32765_{(10)}$$

etc. bis ...

$$8FFF_{(16)} + 1_{(16)} = 9000_{(16)} \quad / \quad -28672_{(10)} + 1_{(10)} = -28671_{(10)}$$

etc. bis ...

$$FFFD_{(16)} + 1_{(16)} = FFFE_{(16)} \quad / \quad -3_{(10)} + 1_{(10)} = -2_{(10)}$$

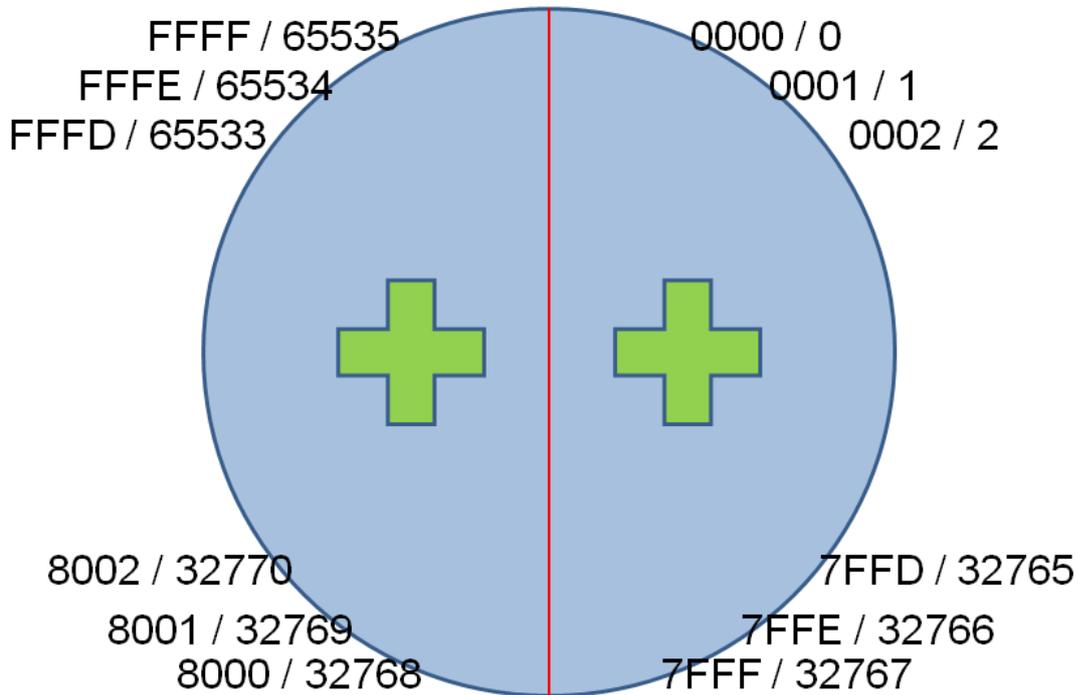
$$FFFE_{(16)} + 1_{(16)} = FFFF_{(16)} \quad / \quad -2_{(10)} + 1_{(10)} = -1_{(10)}$$

$$FFFF_{(16)} + 1_{(16)} = 0000_{(16)} \quad / \quad -1_{(10)} + 1_{(10)} = 0_{(10)}$$

Analoges gilt für 4 Byte lange binäre Felder mit der Definition PIC S9(08) BINARY.

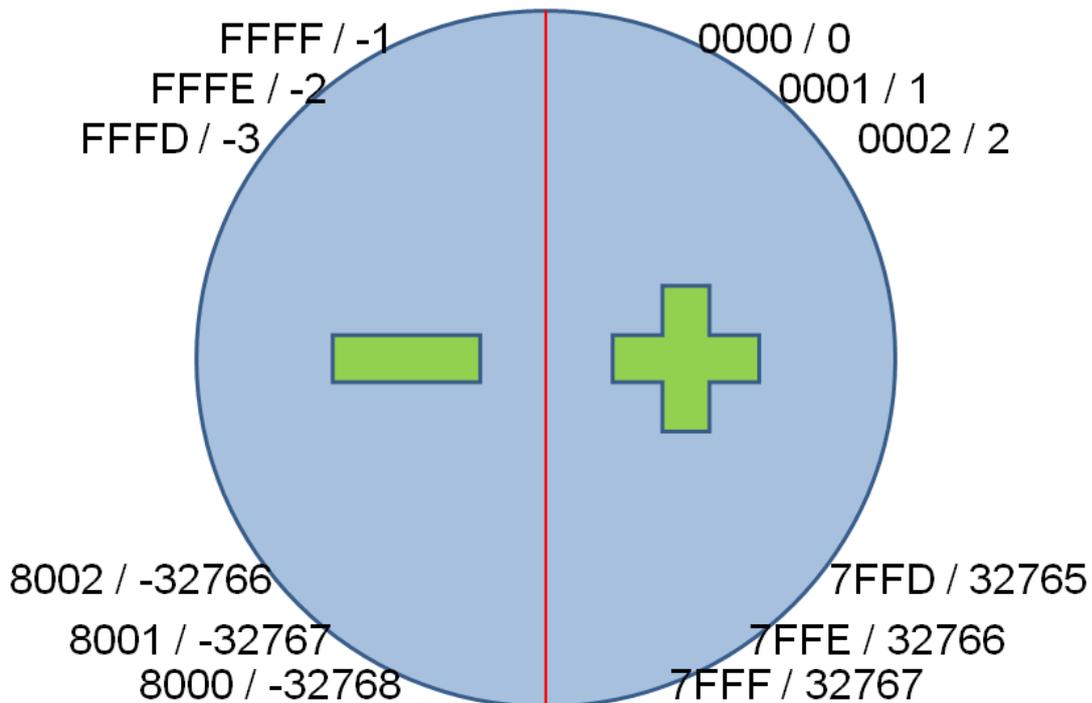
Der binäre Kreis – ohne Vorzeichen – 2 Byte Feld (PIC 9(4) binary)

Hinweis: Zu dieser Definition in COBOL gibt es keine Entsprechung in PL/1 und DB2.



Der binäre Kreis – ohne Vorzeichen – 2 Byte Feld (PIC S9(4) binary)

Hinweis: PL/1: fixed bin(15), DB2: smallint

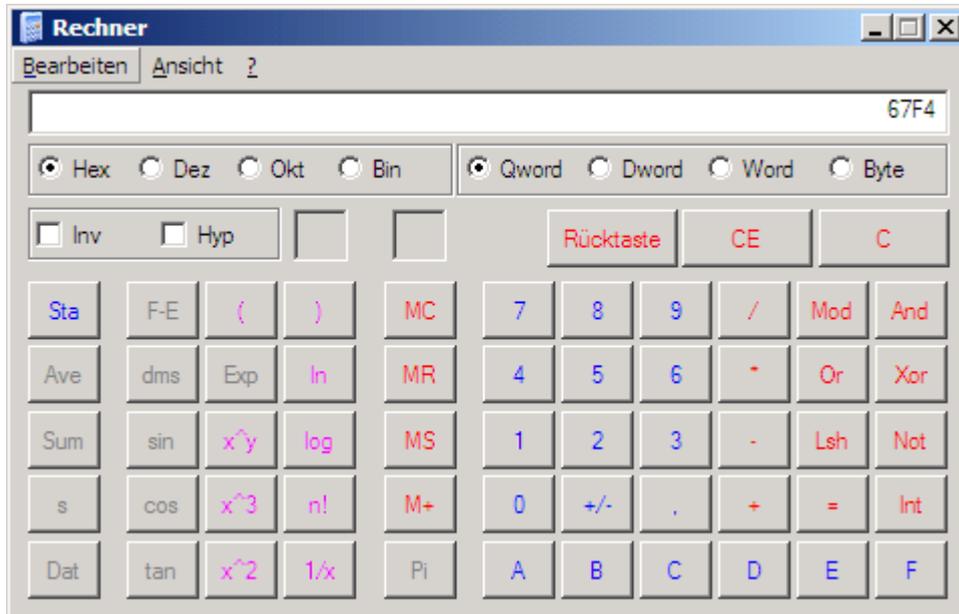


Umrechnen von (positiven) binären Feldern in dezimale Werte

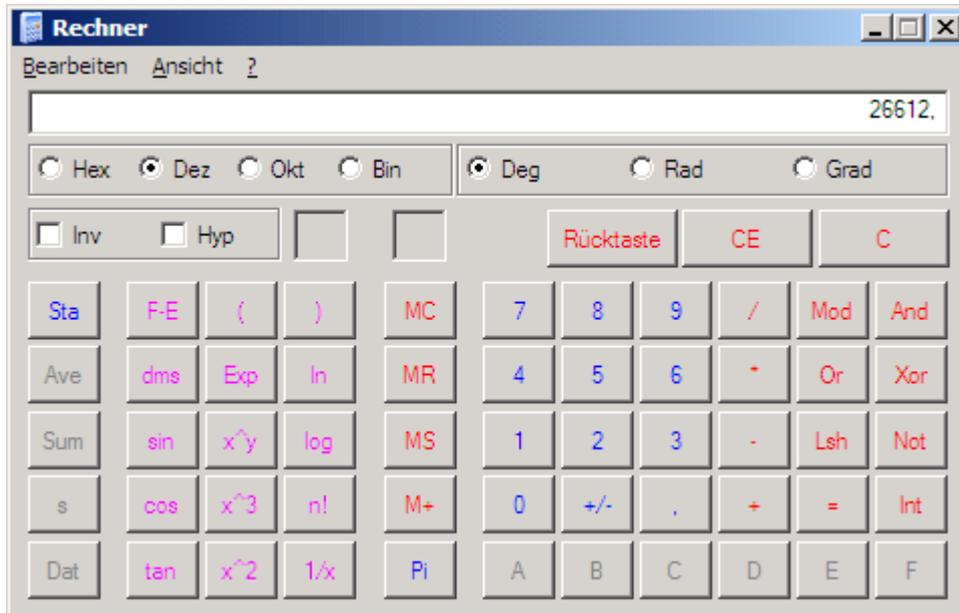
unsere Ausgangsdefinition:

```
F-BIN      PIC S9(04) BINARY.
```

Nehmen wir an, wir haben im Dump für dieses Feld den (positiven) Wert $67F4_{(16)}$ gefunden.
Eingabe in Rechner:



Umschalten auf dezimal:



Wir haben also den Wert $26612_{(10)}$.

Umrechnen von (positiven) binären Feldern in dezimale Werte ab Windows 7

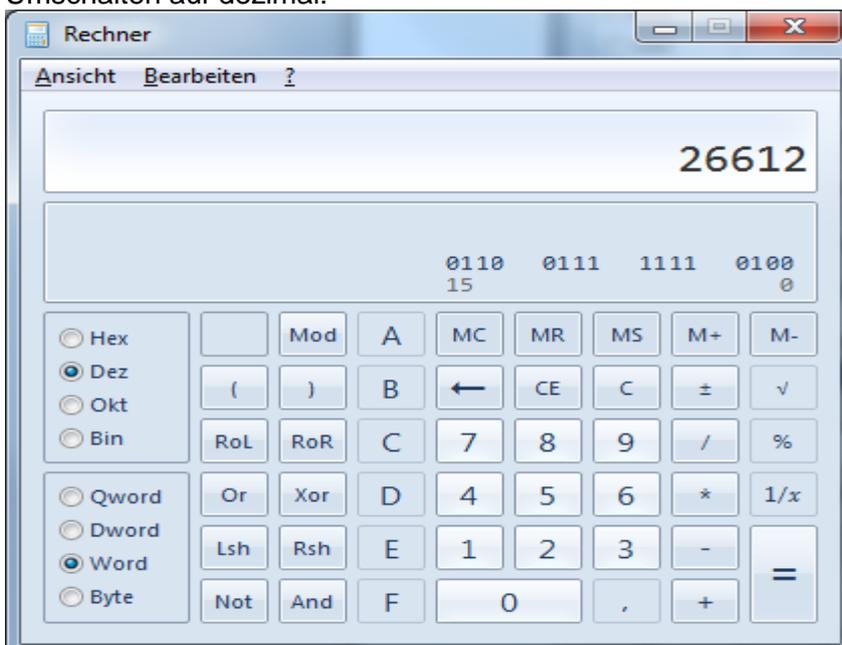
unsere Ausgangsdefinition:

```
F-BIN      PIC S9(04) BINARY.
```

Nehmen wir an, wir haben im Dump für dieses Feld den (positiven) Wert $67F4_{(16)}$ gefunden. Eingabe in Rechner, der auf **Ansicht „Programmierer“** eingestellt ist:



Umschalten auf dezimal:



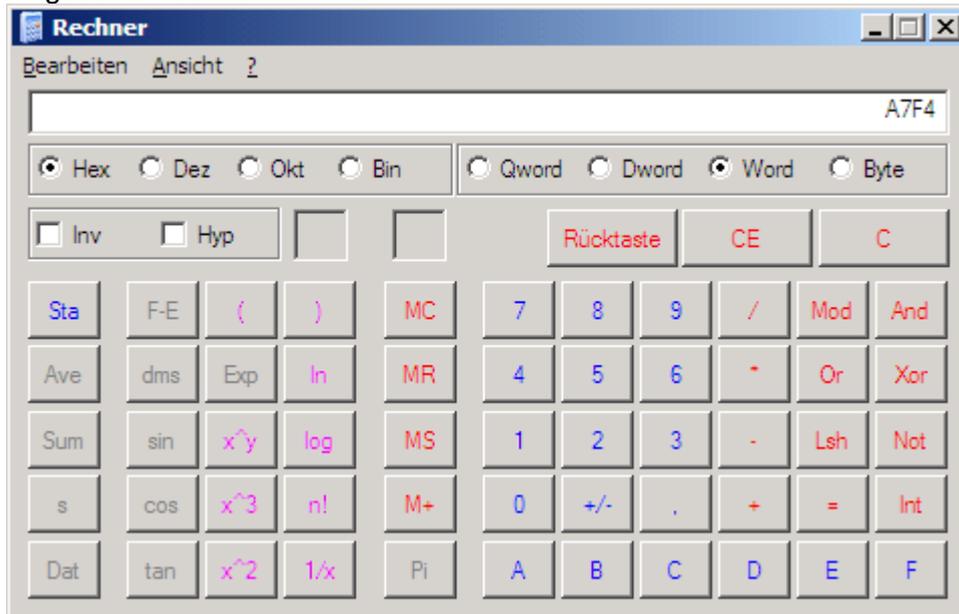
Wir haben also den Wert $26612_{(10)}$.

Umrechnen von (negativen) binären Feldern in dezimale Werte

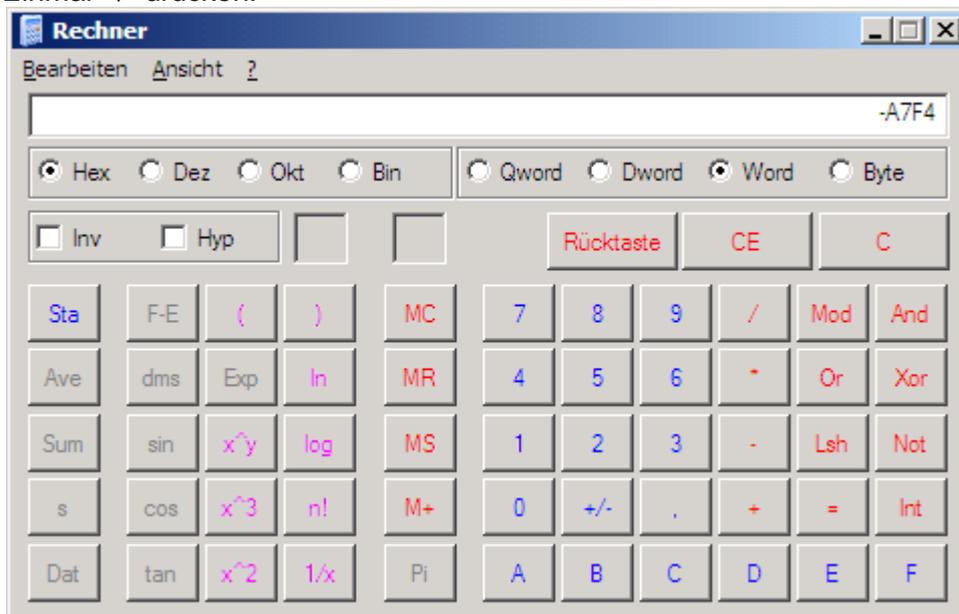
Hier müssen wir den PC-Taschenrechner austricksen, damit er richtig auf negativ umrechnet.

Nehmen wir an, wir haben im Dump für dieses Feld den (negativen) Wert $A7F4_{(16)}$ gefunden. Zur Erinnerung: Das erste Bit zeigt an, ob es eine negative Zahl ist. Ist die erste Hexa-Zahl 8 oder größer, ist die Zahl also negativ.

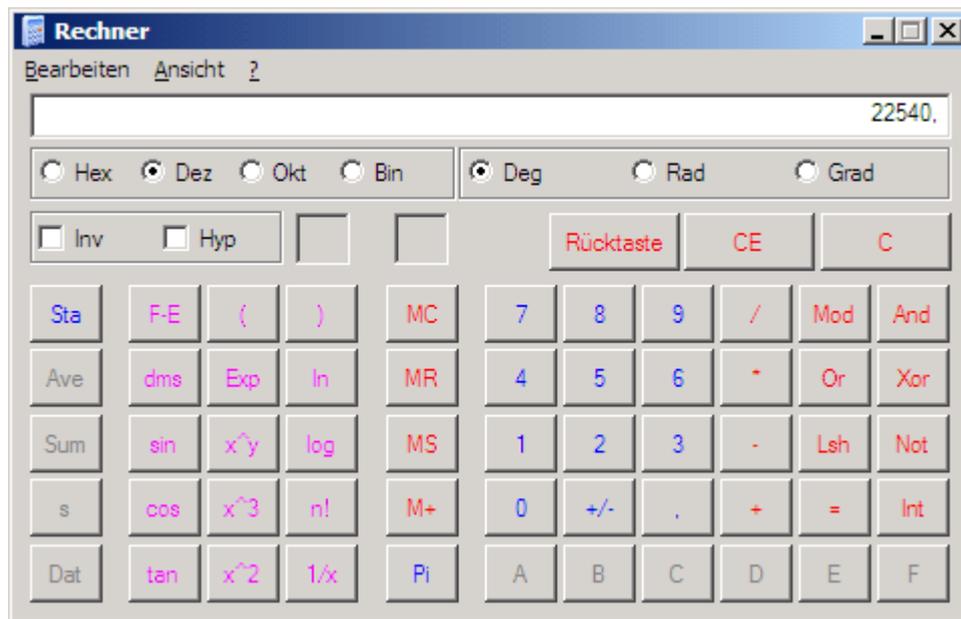
Eingabe in Rechner:



Einmal +/- drücken:



Nach dezimal wechseln:



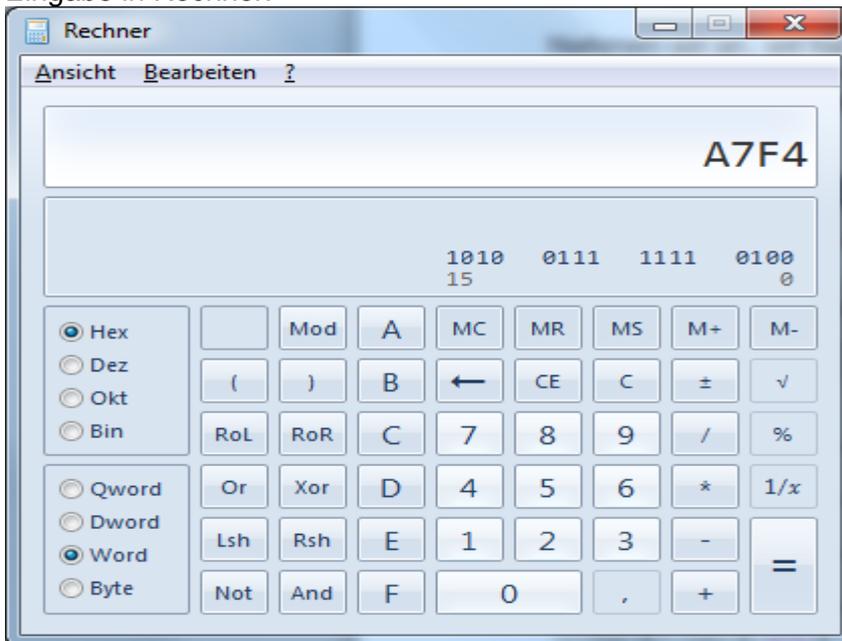
$A7F4_{(16)}$ ist dann $-22540_{(10)}$!

Umrechnen von (negativen) binären Feldern in dezimale Werte ab Windows 7

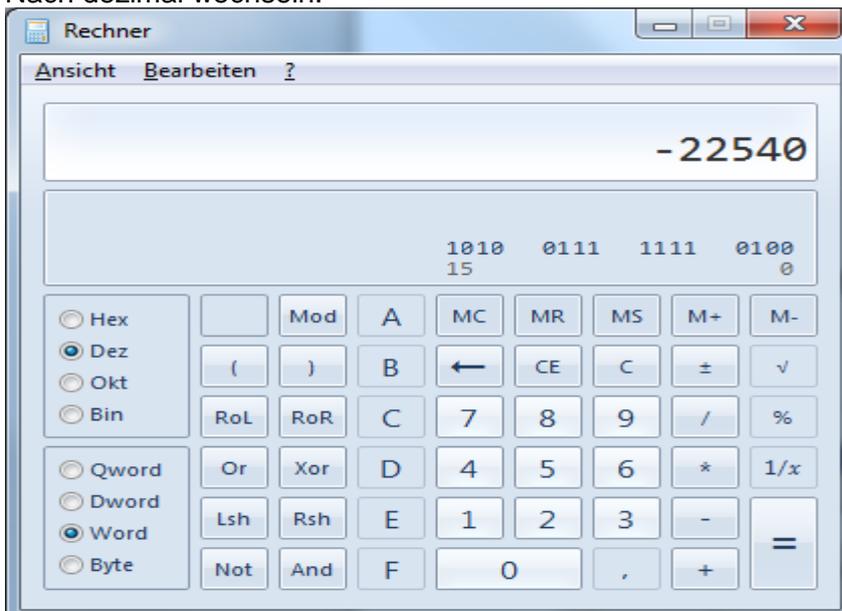
Mit der neuen Windowsversion wurde durch die Ansicht „Programmierer“ in Calc eine Einstellung aufgeliefert, die eine trickreiche Umrechnung in negative binäre Werte überflüssig macht.

Nehmen wir an, wir haben im Dump für dieses Feld den (negativen) Wert $A7F4_{(16)}$ gefunden. Zur Erinnerung: Das erste Bit zeigt an, ob es eine negative Zahl ist. Ist die erste Hexa-Zahl 8 oder größer, ist die Zahl also negativ.

Eingabe in Rechner:



Nach dezimal wechseln:

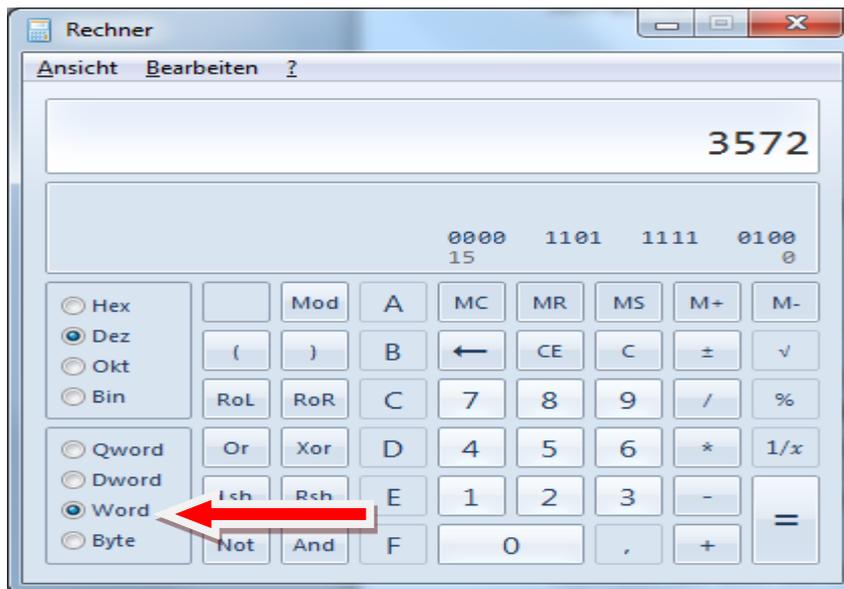


$A7F4_{(16)}$ ist dann $-22540_{(10)}$

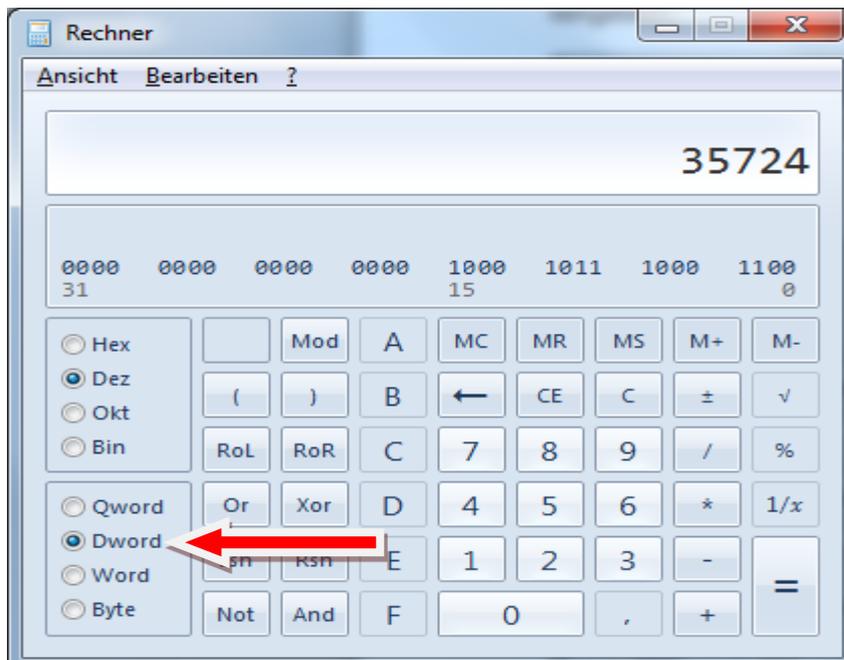
Umrechnen von positiven binären Feldern > 32767 in dezimale Werte ab Windows 7 (COBOL: PIC 9(04) binary)

Dass die Ansicht „Programmierer“ die automatische Erkennung von Vorzeichen beinhaltet heißt aber auch, dass COBOL-Felder, die größer als 32767 sind, mit der Definition PIC 9(04) binary nicht richtig dargestellt werden können. Zur Erinnerung: Zahlen > 32767 haben mindestens eine 8 auf dem ersten Halbbyte.

Nehmen wir also die $35724_{(10)}$. Diese Zahl können wir bei einem 2-Byte-Feld nicht (mehr) eingeben:



Wir müssen dann den Umweg über eine „längere“ Zahl gehen:



Binäre Felder im Hauptspeicher

Und dann erhalten wir das „richtige“ Ergebnis:

