

M3T-NC30WA V.5.30 Release 02 ご使用上のお願い

M16C/60, M16C/30, M16C/Tiny, M16C/20, M16C/10およびR8C/Tinyシリーズ用 Cコンパイラパッケージ M3T-NC30WA V.5.30 Release 02 の使用上の注意事項を連絡します。

- 関数の実引数にconst修飾型オブジェクトのアドレスを渡す場合の注意事項

1. 該当製品

M3T-NC30WA V.5.30 Release 02

2. 内容

実引数がconst修飾型オブジェクトのアドレスである関数呼び出しの記述に対し、正しい型の実引数を記述しても誤った警告メッセージが表示されます。なお、誤った警告が表示されても、生成されたコードに問題はありません。

2.1 発生条件

以下の条件をすべて満たす場合に発生します。

- 呼び出される関数の仮引数がconst修飾型へのポインタ型である。
- (1)の仮引数に対応する実引数に、以下のいずれかの値を記述している。
 - const修飾型オブジェクトにアドレス演算子(&)を適用した結果
 - const修飾型の配列の名称
- (2)のconst修飾型オブジェクト、またはconst修飾型の配列要素は、(1)の仮引数のポインタが指す型と同じ型を持つ。

2.2 発生例

```

const int  ten = 10;          /* 発生条件(3) */
const int  arr[2] = { 1, 2 }; /* 発生条件(3) */

void subr(const int *);      /* 発生条件(1)および(3) */

void mainr(void)
{
    subr(&ten);              /* 発生条件(2) */
    subr(arr);              /* 発生条件(2) */
}

```

メッセージ表示例

[Warning(ccom):example.c,line 8] assignment from const pointer to non-const pointer

====> subr(&ten);

[Warning(ccom):example.c,line 9] assignment from const pointer to non-const pointer

====> subr(arr);

3. 回避策

誤って警告される実引数の前に、仮引数の型へ型変換するキャスト演算子を挿入してください。

```

void mainr(void)
{
    subr((const int *)&ten); /* キャスト演算子を挿入 */
    subr((const int *)arr);  /* キャスト演算子を挿入 */
}

```

4. 恒久対策

本問題は、次期バージョンアップ時に改修する予定です。

[免責事項]

過去のニュース内容は発行当時の情報をもとにしており、現時点では変更された情報や無効な情報が含まれている場合があります。ニュース本文中のURLを予告なしに変更または中止することがありますので、あらかじめご承知ください。