



**CompactRISC™**

---

**CompactRISC CR16 Development Toolset**  
**Release Update 2.1.6**  
**Release Letter**

---

**Part Number: 433521772-001**

**February 1999**

# PREFACE

This release letter describes update version 2.1.6 for the CompactRISC CR16 Development Toolset. This version fixes several problems in version 2.1 of the CompactRISC CR16 Development Toolset which was released in September 1998.

This release letter includes: product description, installation procedure, and lists of known and fixed software errors.

For technical support and additional information, visit our Web site at:

<http://www.CompactRISC.com>

or contact the National Semiconductor support center in your area, as listed on the back of this document.

# CONTENTS

---

<b>1.0</b>	<b>RELEASE PACKAGE CONTENTS .....</b>	<b>5</b>
<b>2.0</b>	<b>PRODUCT DESCRIPTION .....</b>	<b>6</b>
<b>3.0</b>	<b>INSTALLATION PROCEDURE .....</b>	<b>7</b>
3.1	BEFORE YOU INSTALL .....	7
3.2	WINDOWS 95 / NT 4.0 INSTALLATION .....	7
3.3	UNINSTALL .....	7
<b>4.0</b>	<b>FIXED SOFTWARE ERRORS .....</b>	<b>8</b>
4.1	COMPILER .....	8
4.2	DEBUGGER .....	10
<b>5.0</b>	<b>KNOWN SOFTWARE ERRORS.....</b>	<b>11</b>
5.1	COMPILER .....	11



## 1.0 RELEASE PACKAGE CONTENTS

---

Part/Lit Number	Description
440521772-001	One self-extracting executable file, containing release update package 2.1.6 for the CompactRISC CR16 Development Toolset version 2.1.
433521772-001	This release letter.

## 2.0 PRODUCT DESCRIPTION

---

This package contains an update of the CompactRISC CR16 Compiler and CR16 Debugger.

Component	Description
<code>crcc.exe</code>	Compilation driver. Generates CR16 assembly files, object files, or executable object files.
<code>cpp.exe</code>	ANSI-C preprocessor.
<code>cc1.exe</code>	ANSI-C optimizing compiler.
<code>specs</code>	Compilation option control file.
<code>crdb.exe</code>	Source-level graphic debugger for CR16 programs.
<code>crdb.hlp</code>	Debugger help file.

## 3.0 INSTALLATION PROCEDURE

---

### 3.1 BEFORE YOU INSTALL

During installation, you are asked for the location of the `CRDIR` directory, i.e., the directory where you installed the CompactRISC CR16 Development Toolset Version 2.1.

We recommend closing any running application, before you start the installation.

### 3.2 WINDOWS 95 / NT 4.0 INSTALLATION

Carry out the following steps:

1. Execute the self-extracting installation file:  
`CompactRISCupdate216.exe`.
2. Follow the instructions in the installation program.
3. The installation program copies the new files to their locations in your CompactRISC CR16 Development Toolset directory. The program also creates a backup directory, located in the `CRDIR` directory, that contains all the old files.

### 3.3 UNINSTALL

To remove the release update package, first run the CompactRISC CR16 Development Toolset Version 2.1 Uninstall program, then manually remove the `CRDIR/Backup` directory.

## 4.0 FIXED SOFTWARE ERRORS

---

The following errors have been fixed in this release.

### 4.1 COMPILER

#### Issue 1603 (CR16A only)

**Problem** When optimizing for speed, the save emulation optimization option was also used. As save-emul is a space optimization, this might have slightly reduced the speed.

#### Issue 1710 (CR16A only)

**Problem** A false warning was given when casting a function pointer to unsigned short.

#### Issue 1712

**Problem** Wrong code was generated for a linked list of far pointers.

#### Issue 1713

**Problem** The comparison of far pointers against (`__far void *`)0 was incorrect.

#### Issue 1727

**Problem** The debug information for the filename was inaccurate (produced a double backslash as a dir separator instead of a single one).

#### Issue 1735

**Problem** Wrong code (wrong return) was generated for an empty interrupt routine.



## Issue 1736

**Problem** Bit operation on a far bitfield caused an internal error.

## Issue 1737

**Problem** Wrong code was generated when a function argument was zero, and the called function expected a wide (far/large) pointer.

## Issue 1738

**Problem** Reference to far array of structures was wrong.

## Issue 1748 (CR16A only)

**Problem** When optimizing for speed, the address of an argument that is passed on stack was not calculated properly. This happened when the number of arguments passed to a function was above four.

## Issue 1757

**Problem** It was impossible to define a structure bigger than 8K (this also led to wrong result when `sizeof` was applied to such a structure).

## Issue 1759

**Problem** The `-mlarge` compilation option was missing in the list file.

## Issue 1760

**Problem** The `__FILE__` macro did not work properly when the file name included a backslash (`cpp.exe`).

## 4.2 DEBUGGER

### Issue 1052

**Problem** If you changed `Core` or `Radix` in the debugger `Config` menu, while the focus was not in the main debugger window, the check mark was not updated, but remained in its previous position.

Although it was stated that this issue was fixed for previous version, it was, in fact, fixed in this release.

### Issue 1177

**Problem** A remote `Reset` command, during Virtual I/O execution, did not function correctly.

### Issue 1287

**Problem** The debugger did not support verbose mode for displaying all `DbgCom` calls and the Target Monitor (TMON) protocol, although this was announced as an enhancement in the previous release letter.

### Issue 1591

**Problem** Execution did not abort after defining a conditional breakpoint.

### Issue 1613

**Problem** When updating the setup window for the performance simulator, the log file was always reopened (i.e., erased) even if its path was not changed.

### Issue 1498

**Problem** When using the show performance window with function granularity, the entries update window was not functional.

## 5.0 KNOWN SOFTWARE ERRORS

---

The following errors have not been fixed in this release.

### 5.1 COMPILER

#### Issue 1754

**Problem** It is not possible to define constant address references over 64K.

**Example** `(*(__far char *) (0x11000)) = 5;`

would write to a truncated address.

**Workaround** Turn off the optimization flag, or use:

```
const __far char * cptr = (__far char *)0x11000;
*cptr = 5;
```

#### Issue 1755

**Problem** The compiler produces the wrong code for pointer to far array of nested structures.

**Example**

```
struct exam {
    int mark;
    char subject[20];
};
struct desc {
    struct exam exams[2];
    int average;
};
__far struct desc pupil_info [10];
__far struct desc *ptr;
int index1, sum;
main(){
    ptr = &pupil_info[1];
    sum = ptr->exams[index1].mark; /* this assignment would be
                                   compiled incorrectly */
}
```

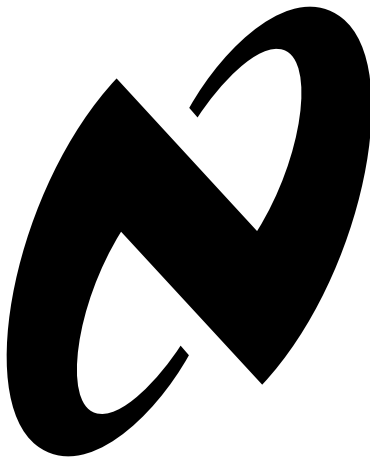
**Workaround**    Use a temporary variable ptr1:

```
main(){
    __far struct exam *ptr1;
    ptr = &pupil_info[1];
    ptr1 = &(ptr->exams[index1]);
    sum = ptr1->mark;
}
```



*National Semiconductor supplies a comprehensive set of service and support capabilities. Complete product information and design support is available from National's customer support centers.*

*To receive sales literature and technical assistance, contact the National support center in your area.*



**Americas**

**Fax: 1-800-737-7018**

**Email: support@nsc.com**

**Tel: 1-800-272-9959**

**Europe**

**Fax: +49 (0)1 80 5 30 85 86**

**Email: europe.support@nsc.com**

**Deutsch**

**Tel: +49 (0)1 80 5 30 85 85**

**English**

**Tel: +49 (0)1 80 5 32 78 32**

**Japan**

**Fax: 81-3-5620-6179**

**Tel: 81-3-5620-6175**

**Asia Pacific**

**Fax: 65-250-4466**

**Email: sea.support@nsc.com**

**Tel: 65-254-4466**

*Visit us on the Worldwide Web at*

*<http://www.national.com>*

*or*

*<http://www.CompactRISC.com>*