

## Customer Notification

# IAR EWV850

## IAR Embedded Workbench for V850

### Operating Precautions

---

**Y-IAR-EWV850-FULL-MOBILE**

**Y-IAR-EWV850-FULL**

**EWV850-KS64-EE**

**EWV850-KS32-EE**

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

"Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority- owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## Table of Contents

A) Table of Operating Precautions for the Compiler (ICCV850) and Assembler (AV850) ...	5
B) Table of Operating Precautions for the Linker (XLINK) .....	6
C) Table of Operating Precautions for the Debugger (CSV850) .....	7
D) Table of Operating Precautions for the IAR platform (EWV850) .....	8
E) Description of Operating Precautions for the Compiler (ICCV850) and Assembler (AV850) .....	9
F) Description of Operating Precautions for the Linker (XLINK) .....	20
G) Description of Operating Precautions for the Debugger (CSV850) .....	23
H) Description of Operating Precautions for the IAR Platform (EWV850) .....	32
I) List of Error messages of the ICE .....	36
J) Valid Specification .....	46
K) Revision History .....	47

### A) Table of Operating Precautions for the Compiler (ICCV850) and Assembler (AV850)

No.	Outline	Version	ICCV850 (Compiler)									
			V3.71.2	V3.80.1	V3.81.1	V4.10.1	V4.10.2	V4.10.3	V4.20.1	V4.20.2	V5.10.1	V5.10.2
<a href="#">a1</a>	If an SFR is defined twice at the same address, linking fails (EW10021)		×	×	×	✓	✓	✓	✓	✓	✓	✓
<a href="#">a51</a>	Copy of a constant structure with a zero content generates incorrect code		×	×	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">a52</a>	long long comparison with a constant always returns true		-	×	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">a53</a>	Loading of an address to an sfr could cause internal error (1)		✓	×	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">a54</a>	Loading of an address to an sfr could cause internal error (2)		✓	×	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">a55</a>	Logical AND ('&&') operation generates incorrect code when operands are swapped		×	✓	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">a56</a>	Nested loops with loop variables of different types could in some very rare cases trigger an internal compiler error		×	✓	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">a57</a>	Faulty Index Calculation for a Loop with an Index Counter		✓	✓	×	✓	✓	✓	✓	✓	✓	✓
<a href="#">a58</a>	Wrong indirect post Increment of a Result of a post Increment		×	×	×	×	×	✓	✓	✓	✓	✓
<a href="#">a59</a>	Wrong Optimization of indirect Variable increment in nested do Loops		×	×	×	×	×	✓	✓	✓	✓	✓
<a href="#">a60</a>	Extended Keyword __saddr causes an Internal Error		✓	✓	✓	×	✓	✓	✓	✓	✓	✓
<a href="#">a61</a>	Extended Keyword __saddr causes an Internal Error (2)		✓	✓	✓	×	×	✓	✓	✓	✓	✓
<a href="#">a62</a>	MISRA C 2004 Rule 10.6 not triggered		×	×	×	×	×	×	×	✓	✓	✓
<a href="#">a63</a>	Inconsistency of extended Keyword __monitor		-	-	-	×	×	×	×	×	✓	✓
<a href="#">a64</a>	32-bit switch cases can fail		×	×	×	×	×	×	×	×	×	✓
<a href="#">a65</a>	Faulty code generated on optimization level medium and higher		×	×	×	×	×	×	×	×	×	✓

×: Applicable

✓: Not applicable

-: Not checked

**B) Table of Operating Precautions for the Linker (XLINK)**

No.	Outline	Version	XLINK (linker)									
			V4.61R	V5.0.0.2	V5.2.6.19	V5.6.0.36	V6.1.1.52	V6.3.3.74	V6.3.4.78	V6.5.4.100		
<a href="#">b3</a>	Base relat. addressing with 23-Bit disp. requires new sections BREL23_X		×	×	×	×	×	×	×	×		
<a href="#">b4</a>	Specifying the alignment of a segment will result in an internal error of XLINK.		×	✓	✓	✓	✓	✓	✓	✓		
<a href="#">b5</a>	XLINK Linker doesn't completely fill up unused bytes of a segment in case there follows an empty placeholder segment		-	×	✓	✓	✓	✓	✓	✓		
<a href="#">b6</a>	Incorrect Debug Information in ELF/DWARF Output File		-	-	×	×	×	×	✓	✓		

×: Applicable

✓: Not applicable

-: Not checked

**C) Table of Operating Precautions for the Debugger (CSV850)**

No.	Outline	Version									
		V3.81.1	V3.81.3	V3.81.4	V4.10.1	V4.10.2	V4.10.3	V4.20.1	V4.20.1	V5.10.1	V5.10.2
<a href="#">c9</a>	Trace information not compensated when "BranchPC Address+ Data Access" is chosen	x	x	x	x	x	x	x	x	x	x
<a href="#">c13</a>	Support of the same name for data type and data object is not complete	x	x	x	x	x	x	x	x	x	x
<a href="#">c14</a>	Softlinks cannot be used in SWBP definition	x	x	x	x	x	x	x	x	x	x
<a href="#">c17</a>	Symbolic window partially updated.	x	x	x	x	x	✓	✓	✓	✓	✓
<a href="#">c18</a>	Selfprogramming Emulation – Reset Vector	x	x	x	x	x	✓	✓	✓	✓	✓
<a href="#">c20</a>	Error message of ICE is incorrectly displayed: Unknown Error.	x	x	x	x	x	✓	✓	✓	✓	✓
<a href="#">c40</a>	Limitation of Software-Breakpoints (V850E2M only)	x	x	x	✓	✓	✓	✓	✓	✓	✓
<a href="#">c43</a>	Not possible to Download srec/hex File from C-Spy "Memory" Window	x	x	x	✓	✓	✓	✓	✓	✓	✓
<a href="#">c44</a>	Increased Download Time for V850E2 Devices using the E1 Emulator	x	x	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">c45</a>	Memory overridden if multiple images are loaded	x	x	✓	✓	✓	✓	✓	✓	✓	✓
<a href="#">c46</a>	IECUBE2 Driver: Trace Buffer Size cannot be reduced	x	x	x	x	x	✓	✓	✓	✓	✓
<a href="#">c47</a>	IDE Crashes due to unintended triggered Interrupt	-	-	-	x	x	✓	✓	✓	✓	✓
<a href="#">c48</a>	IDCODE Input disabled in Hardware Setup	-	-	-	x	x	✓	✓	✓	✓	✓
<a href="#">c49</a>	IDE Crash changing low or high Byte of 16bit I/O Register	-	-	-	x	x	x	✓	✓	✓	✓
<a href="#">c50</a>	IDE Crash at Single-Step Command while Live-Watch-Window is open	✓	✓	✓	x	x	x	✓	✓	✓	✓
<a href="#">c51</a>	Wrong Address area for internal RAM displayed in C-SPY E1 Hardware Setup	x	x	x	x	x	x	x	✓	✓	✓
<a href="#">c52</a>	The CMOV instruction is not correctly implemented (C-SPY simulator only)	x	x	x	x	x	x	x	x	x	✓

x: Applicable

✓: Not applicable

-: Not checked

**D) Table of Operating Precautions for the IAR platform (EWV850)**

No.	Outline	Version	EWV850						
			V5.2A	V5.3	V5.4	V6.0	V6.5	V7.2	V8.1
<a href="#">d1</a>	The Source Browser window does not update properly.		×	×	×	×	×	×	×
<a href="#">d2</a>	The heap size under “Project Options” is limited.		×	✓	✓	✓	✓	✓	✓
<a href="#">d3</a>	Message box: ‘The project contains the unknown tool ‘Coder’		×	×	×	×	×	×	×
<a href="#">d4</a>	Missing function names in "Go to function" view.		-	-	×	✓	✓	✓	✓
<a href="#">d5</a>	No Source Browse Information generated		✓	✓	✓	✓	×	✓	✓
<a href="#">d6</a>	Actual Linker-MAP-File not automatically updated in Editor		-	-	×	×	×	✓	✓
<a href="#">d7</a>	Incorrect End Addresses in XCL File Template		×	×	×	×	✓	✓	✓

×: Applicable

✓: Not applicable

-: Not checked



## E) Description of Operating Precautions for the Compiler (ICCV850) and Assembler (AV850)

No. a1	If an SFR is defined twice at the same address, linking fails
	<p><u>Details:</u></p> <p>Example:</p> <pre> __IO_REG16_BIT( TMC00, 0xFFFFF57A, __READ_WRITE ) __IO_REG8_BIT( TMC00L, 0xFFFFF57A, __READ_WRITE ) __IO_REG8_BIT( TMC00H, 0xFFFFF57B, __READ_WRITE ) </pre> <p>The SFR area is defined twice at the same address with different names. This causes XLINK to fail if accesses are made through both names in modules that are to be linked together.</p> <p><u>Workaround:</u></p> <p>Replace the macro calls in the io_XXX.h-file that specifies the SFR with the problem with an anonymous union. For example, in 'io_v850e_df3114.h' replace the macro calls above by the following union:</p> <pre> __near __no_init volatile __READ_WRITE union {     struct {         union {             unsigned char TMC00L;             __BITS8 TMC00L_bit;         };         union {             unsigned char TMC00H;             __BITS8 TMC00H_bit;         };     };     union {         unsigned short TMC00;         __BITS16 TMC00_bit;     }; } @ 0xFFFFF57A; </pre>

<b>No. a51</b>	<b>Copy of a constant structure with a zero content generates incorrect code</b>
	<p><u>Details:</u></p> <p>This optimization problem occurs when a copy is performed from a constant struct with a zero content. In that case, the compiler will insert a call to library-routine ?SetMemoryWords (or –Halfword or –Byte), which does not restore registers correctly.</p> <p><u>Workaround:</u></p> <p>Use the ANSI-C library function memcpy() for the initialization instead.</p>

<b>No. a52</b>	<b>long long comparison with a constant always returns true</b>
	<p><u>Details:</u></p> <p>The following comparison of a long long variable with a constant always returns true:</p> <pre>unsigned long long value = 128;  int main(void) {     if( value &gt;= 1477)     {         return 1;     }     else     {         return 0;     } }</pre> <p>The compiler tries to use an addition for the comparison. The addition uses R0 as destination, but in V850 R0 is the zero register.</p> <p><u>Workaround:</u></p> <p>None.</p>

<b>No. a53</b>	<b>Loading of an address to an sfr could cause internal error (1)</b>
	<p><u>Details:</u></p> <p>Compilation of the following code generates a tool internal error:</p> <p>Internal Error: [CoreUtil/General]: Size mismatch for "MOV -1497,r1", inserted as 6 bytes, assembled as 4 bytes</p> <pre>#include &lt;io70f3724.h&gt; #define MASK_DMA_ADRESS      0x03FFFFFF  void Pbus_TxStart(void) {     unsigned int    L;      L = (unsigned int)&amp;UA2TX &amp; MASK_DMA_ADRESS; } </pre> <p><u>Workaround:</u></p> <pre>const static char volatile __no_bit_access * p = &amp;UA2TX;  void Pbus_TxStart(void) {     unsigned int    L;      L = (unsigned int)p &amp; MASK_DMA_ADRESS; } </pre>
<b>No. a54</b>	<b>Loading of an address to an sfr could cause internal error (2)</b>
	<p><u>Details:</u></p> <p>Following code sent by the customer leads into a tool internal error:</p> <pre>DSA0H = (unsigned short)((((unsigned long)&amp;ADA0CR1) &amp; 0x03ffffff) &gt;&gt; 16); </pre> <p>Internal Error: [CoreUtil/General]: Size mismatch for "MOV -3566,r1", inserted as 6 bytes, assembled as 4 bytes</p> <p>DSA0H and ADA0CR1 are 16-bit SFR registers in V850ES/Jx3-L (uPD70F3842) CPU</p> <p><u>Workaround:</u></p> <p>(see workaround item 53 above)</p>

<b>No. a55</b>	<b>Logical AND ('&amp;&amp;') operation generates incorrect code when operands are swapped</b>
	<p><u>Details:</u></p> <p>In some cases two tests on the form <code>expr1 &gt; c1 &amp;&amp; expr2 &lt; c2</code> or <code>expr1 &lt; c1    expr2 &gt; c2</code> could be optimized as a range test even though <code>expr1</code> is not identical to <code>expr2</code>.</p> <p>The following code is not compiled correctly if the order of '&amp;&amp;' operands is swapped and optimization is low or higher.</p> <p>With <code>optimization=none</code>, the code is compiled correctly.</p> <pre>// Next lines are compiled correctly if ((cindex&gt;0) &amp;&amp; (i&lt;3))     __asm("NOP");           // Statement is reached  // Next lines are not compiled correctly with optimize level low if ((i&lt;3) &amp;&amp; (cindex&gt;0))     __asm("NOP");           // Statement is not reached</pre> <p><u>Workaround:</u></p> <p>None.</p>

<b>No. a56</b>	<b>Nested loops with loop variables of different types could in some very rare cases trigger an internal compiler error</b>
	<p><u>Details:</u></p> <p>Nested loops with loop variables of different types could in some very rare cases trigger an internal error with high optimization:</p> <pre>"ToolInternalError: Internal Error:[CoreUtil/General]: integral sub"</pre> <p><u>Workaround:</u></p> <p>Use same type for the loop-variable in both the inner and outer loop when nested loops are used. See example below.</p> <pre>#define NULL (void *) 0 typedef unsigned short datum; #define BLKSIZE 32  datum *memTestDevice(volatile datum * baseAddress, unsigned long nBytes) {     unsigned long          n;     datum                Save[BLKSIZE];     unsigned long offset;      for( n = 0; n &lt; nBytes; n++ )     {         for (offset = n*BLKSIZE; offset &lt; (n+1)*BLKSIZE; offset++)         {             baseAddress[offset] = Save[offset-n*BLKSIZE];         }     }     return (NULL); }</pre>

No. a57	Faulty Index Calculation for a Loop with an Index Counter
	<p><u>Details:</u></p> <p>Index calculation for a loop with an index counter where the loop count is known can in rare cases generate faulty code on optimization level high-speed (-Ohs).</p> <pre>typedef unsigned char U8; typedef unsigned int  U16;  U8  g1; U32 g2; U32 array[10];  void test( void ) {     U8 loc1;      loc1 = g1;     do {         --loc1;     } while ((loc1 &lt; g1) &amp;&amp; (array[loc1] == g2)); }</pre> <p><u>Workaround:</u></p> <p>Avoid high speed optimization, either on the command line or use pragma optimize=size on the function.</p>

No. a58	<b>Wrong indirect post Increment of a Result of a post Increment</b>
	<p><u>Details</u></p> <p>Independent of the selected optimization level the compiler generates wrong code for the indirect post increment of a result of a post increment</p> <p><u>Example</u></p> <pre> #include &lt;stdio.h&gt; #include &lt;assert.h&gt;  char    c[2] = {'a','b'}; char *pc[2] = {&amp;c[0],&amp;c[1]}; char **ppc = &amp;pc[0];  int test(void) {     char cc_ret;     cc_ret = *(*ppc++)++;     assert(pc[0]==pc[1]);     return (int)cc_ret; } </pre> <p><u>Workaround</u></p> <p>Use separate statements for post increment:</p> <pre> int workaround (void) {     ...     cc_ret = *(*ppc);    /* problem */     (*ppc)++;     ppc++;     ... } </pre>

No. a59	Wrong Optimization of indirect Variable increment in nested do Loops
	<p><u>IAR Reference:</u> EW23965</p> <p><u>Details</u></p> <p>On high optimization, a variable (v) can be optimized incorrectly if *v is incremented with a constant value inside a do loop, *the do loop has a computable trip count, *the do loop is surrounded by another loop with a computable trip count, and *v is not used inside either of the two loops, except for the increment.</p> <p><u>Example</u></p> <pre> #include &lt;stdio.h&gt; #include &lt;assert.h&gt;  int i, i0, i1, i2, i3, i4, i5, i6, i7, int main(void) {     i = i3 = i4 = 0;     i0 = 2;     do {         i1 = 2;         while (i1--) {             for (i2 = 0; i2 &lt; 2; i2++) {                 i5 = 2;                 do {                     i6 = 2;                     while (i6--) {                         for (i7 = 0; i7 &lt; 2; i7++)                             i++;                     }                 } while (--i5);             }         }     } while (--i0);     assert (i == 64);     printf("i= %d\n",i);     return 0; } </pre> <p><u>Workaround</u></p> <p>None. It is listed as known problem in release notes V4.10.2</p>



No. a60	<b>Extended Keyword <code>__saddr</code> causes an Internal Error</b>
	<p><u>IAR Reference:</u> EW24051</p> <p><u>Details</u></p> <p>Using the extended keyword <code>__saddr</code> causes a compiler internal error:</p> <p>Tool Internal Error: Internal Error: [TBeZeroedVariableGenerator::GenerateVariable]: Unhandled memory attribute</p> <p><u>Example</u></p> <pre>__saddr unsigned int dummy;</pre> <p><u>Workaround</u></p> <p>None. Install the service pack V4.10.2 to fix this issue.</p>
No. a61	<b>Extended Keyword <code>__saddr</code> causes an Internal Error (2)</b>
	<p><u>IAR Reference:</u> EW24056</p> <p><u>Details</u></p> <p>Using the extended keyword <code>__saddr</code> to define a variable not initialized by '0' causes a compiler internal error:</p> <p>Tool Internal Error: Internal Error: [TBeZeroedVariableGenerator::GenerateVariable]: Unhandled memory attribute</p> <p><u>Example</u></p> <pre>__saddr unsigned int dummy=0x1234;</pre> <p><u>Workaround</u></p> <p>Do the initialization by application function:</p> <pre>__saddr unsigned int dummy;  void init_var(void) {     dummy = 0x1234 }</pre>

No. a62	<b>MISRA C 2004 Rule 10.6 not triggered</b>
	<p><u>IAR Reference:</u> EW24733</p> <p><u>Details</u></p> <p>The compiler does not check MISRA-C 2004 rule 10.6 correctly. It bases the check on the usage of the constant instead of on the type of the constant.</p> <p><u>Example:</u></p> <pre>#define UNSIGNED_CHAR_C 0x12 #define UNSIGNED_SHORT_C 0x1234 #define UNSIGNED_LONG_C 0x12345678  unsigned char var1 = UNSIGNED_CHAR_C; /* Error [Pm127] */ unsigned short var2 = UNSIGNED_SHORT_C; /* Error [Pm127] */ unsigned long var3 = UNSIGNED_LONG_C; /* no Error */</pre> <p>In above example error Pm127 should be triggered three times instead of only twice.</p> <p><u>Workaround</u></p> <p>None.</p>
No. a63	<b>Inconsistency of extended Keyword <code>__monitor</code></b>
	<p><u>IAR Reference:</u> EW25971</p> <p><u>Details</u></p> <p>Using IAR function object attributes (like <code>__monitor</code>) with member functions of template classes defined outside the class definition does not work properly. Specifying the attribute both on the declaration and the definition of the function results in a nonsensical error message ("declaration is incompatible with ...").</p> <p><u>Example:</u></p> <pre>template &lt;typename T, unsigned long Size&gt; class buffer {     __monitor void clear(); };  template &lt;typename T, unsigned long Size&gt; __monitor void buffer&lt;T, Size&gt;::clear() {     // ... }</pre> <p><u>Workaround</u></p> <p>None; it will be fixed in next update.</p>

No. a64	<b>32-bit switch cases can fail</b>  <i><u>IAR Reference:</u></i> EWV850-457  <i><u>Details</u></i>  Switch cases that end up using value tables with delta changes between cases can fail if a delta change is larger than a signed 16-bit value.  <i><u>Example:</u></i>  None  <i><u>Workaround</u></i>  None; it will be fixed in next update.
No. a65	<b>Faulty code generated on optimization level medium and higher</b>  <i><u>IAR Reference:</u></i> EWV850-455  <i><u>Details</u></i>  On medium and higher optimization levels, the compiler tries to optimize instructions that loads a register with an immediate value by finding another register that already contains that value. This generates faulty code if the lowest 8 bits have the same value.  <i><u>Example:</u></i>  None  <i><u>Workaround</u></i>  None; it will be fixed in next update.

## F) Description of Operating Precautions for the Linker (XLINK)

No. b3	<b>Base relat. addressing with 23-Bit disp. requires new sections BREL23_X (V850E2M only)</b>
	<p><u>Details:</u></p> <p>From compiler V3.71.1 and Linker V4.61R onwards the memory model “medium” is added to support the load/store instructions with 23-Bit displacement of V850E2M-core devices.</p> <p>If you are using a customized XCL file based on the old standard please add the BREL23_X sections as follow:</p> <pre>-Z (CONST) BREL_CBASE, BREL_C, BREL23_C=xxx - yyy -Z (CONST) SADDR7_ID, SADDR8_ID, NEAR_ID, BREL_ID, BREL23_ID, HUGE_ID=xxx - yyy -Z (DATA) BREL_BASE, BREL_I, BREL_Z, BREL_N, BREL23_I, BREL23_Z, BREL23_N=xxx - yyy</pre>
No. b4	<b>Specifying the alignment of a segment will result in an internal error of XLINK.</b>
	<p><u>Details:</u></p> <p>Specifying the alignment of a segment with -Z(type)segment1 align=ranges will result in an internal error of XLINK.</p> <p><u>Example:</u></p> <p>Following commands will fail:</p> <pre>-Z (CODE) FAL_Text   4=00000C20-002FFFFB -Z (CODE) FAL_Const   4=00000C20-002FFFFB -Z (CODE) EEL_Text   4=00000C20-002FFFFB -Z (CODE) EEL_Const   4=00000C20-002FFFFB</pre> <p><u>Workaround:</u></p> <p>Use XLINK version V4.61S or later</p>

No. b5	<b>XLINK Linker doesn't completely fill up unused bytes of a segment in case there follows an empty placeholder segment</b>
	<p><u>Details:</u></p> <p>XLINK Linker doesn't completely fill up unused bytes of a segment in case there follows an empty placeholder segment (e.g. BREL_CBASE) Prerequisite is that at the same time the previous segment will not end on a 4-Byte aligned address. In this case there will be a gap in the checksum calculation.</p> <p><u>Example:</u></p> <p>xcl-File:</p> <pre>-HFF -h0-2FFFFB -J4,crc32  -Z(CONST)NEAR_C=000003E0-00007FFF -Z(CONST)BREL_CBASE,BREL_C=000003E0-0000FFFF -Z(CONST)HUGE_C=000003E0-0002FFFFB -Z(CODE)CSTART,RCODE,ICODE,TRAPVEC,DIFUNCT=000003E0-0002FFFFB -Z(CONST)SADDR7_ID,SADDR8_ID,NEAR_ID,BREL_ID,HUGE_ID=000003E0-0002FFFFB -Z(CONST)CLTVEC=000003E0-0002FFFFB</pre> <p>Linker generates the following MAP-File:</p> <pre>SEGMENT SPACE START ADDRESS END ADDRESS SIZE TYPE ALIGN ===== INTVEC 00000000 - 00000007 8 com 2 ?FILL1 00000008 - 000003DF 3D8 rel 0 NEAR_C 000003E0 - 000003E0 1 rel 0 BREL_CBASE 000003E4 rel 2 CSTART 000003E4 - 0000040F 2C rel 1 RCODE 00000410 - 0000041B C rel 1</pre> <p>Symbol Checksum Memory Start End Initial value</p> <pre>----- __checksum 0xd40f2f0d CODE 00000000 - 000003E0 0x00000000 (#0x00000000) CODE 000003E4 - 0002FFFFB</pre> <p>NEAR_C ends on a misaligned address and the following empty placeholder segment BREL_CBASE starts on a 4-Byte aligned address. The three bytes in between (at 0x3E1, 0x3E2 and 0x3E3) will not be filled up correctly and there will be a gap (three bytes at 0x3E1) in the checksum calculation.</p> <p><u>Workaround:</u></p> <p>One possible workaround is to place a code/const segment before the empty segment. Another workaround is to give the segment content (like a NOP instruction).</p>

No. b6	Incorrect Debug Information in ELF/DWARF Output File
	<p data-bbox="331 309 715 342"><u>IAR Reference:</u> EW25695</p> <p data-bbox="331 383 416 412"><u>Details</u></p> <p data-bbox="331 452 1457 595">When generating output in the ELF/DWARF output format, XLINK does not handle the debug information for variadic functions correctly. The first non-parameter variable in a variadic function is erroneously output as a parameter and the function is not output as being variadic.</p> <p data-bbox="331 636 496 665"><u>Workarounds</u></p> <p data-bbox="331 667 1366 734">None. Will be fixed in next XLINK update (-&gt; <a href="#">IAR Download latest XLINK version</a>).</p>

## G) Description of Operating Precautions for the Debugger (CSV850)

<b>No. c9</b>	<b>Trace information not compensated when “BranchPC Address+ Data Access” is chosen</b>
	<p><u>Details:</u> When choosing the Trace mode “BranchPC Address+ Data Access”, the trace information doesn’t display the Data Accesses instructions. In V3.20A, the trace is not anymore compensated; but still conformed to the information given by the target.</p> <p><u>Workaround:</u> Select “All PC Addr+Data Addr”.</p>
<b>No. c13</b>	<b>Support of the same name for data type and data object is not complete</b>
	<p><u>Details:</u> If the same name is used for a data-object and for a data-type, this data-object cannot be displayed in the watch window. An error is displayed instead of the data-object’s value:</p> <p>[syntax error, unexpected TYPE_NAME] column 1</p> <p>Example:</p> <pre>struct same_name {     int i;     short s;     char c; };  struct same_name S_1; struct same_name *same_name;</pre> <p><u>Workarounds:</u></p> <ol style="list-style-type: none"> <li>1) Use different name for the data –object and –type.</li> <li>2) Enter the physical address of the data-object and the corresponding type-cast to the watch window instead of the symbol name: Example: (struct same_name*) 0xFFFFA008</li> </ol>

<b>No. c14</b>	<b>Softlinks cannot be used in SWBP definition</b>
	<p><u>Details:</u> Softlinks cannot be used to define a code breakpoint by source location:  e.g.: \$PROJ_DIR\$\source\main.c, row 26.</p> <p>If Softlinks are used, the error message “The file does not exist” is displayed, although the path and the filename are correct.</p> <p><u>Workaround:</u>  This issue is listed as improvement proposal for future versions.</p>
<b>No. c17</b>	<b>Symbolic window partially updated.</b>
	<p><u>Details:</u>  When scrolling in the symbolic window it is possible that only half of the window be updated.</p> <p><u>Workaround:</u>  None</p>



No. c18	Self Programming Emulation – Reset vector.
	<p><u>Details:</u></p> <p>When the Flash Self-programming Emulation is enabled, if a reset occurs, PC is initialized to 0x4. This implies that the jump to the start program as to be at the address 0x4 instead of 0x0.</p> <p><u>Workaround:</u></p> <p>Customised the reset vector ?creset in cstartup.s85 file by introducing 2 nops, then include this customised file in the project:</p> <pre>?creset:     NOP     NOP     MOV    __program_start, LP     JMP    [LP]      ENDMOD</pre>
No. c20	Error message of ICE is incorrectly displayed: Unknown Error.
	<p><u>Details:</u></p> <p>The errors which the emulator return while debugging are not clearly output by IAR IDE. Only the error number is displayed, the description is missing. Instead one can read “Unknown message”</p> <p>e.g.:</p> <pre>unknown error 3124 (0x0c34)</pre> <p><u>Workaround:</u></p> <p>None. The error number and the description of the error messages of the NEC ICE are listed in this document: <a href="#">List of Error messages of the ICE</a></p>

<b>No. c40</b>	<b>Limitation of Software-Breakpoints for QB-V850MINI (V850E2M only)</b>
	<p><u>Details:</u></p> <p>Currently the number of software breakpoints for QB-V850MINI in conjunction with debugging of V850E2M-core devices is limited to 4, due to a limitation of the debug control unit on the device.</p> <p><u>Workaround:</u></p> <p>None.</p>
<b>No. c43</b>	<b>Not possible to Download srec/hex File from C-Spy "Memory" Window</b>
	<p><u>Details:</u></p> <p>The C-Spy function "Memory Restore...", "Memory Save..." and "Memory Fill..." function in debugger "Memory" window does not work correctly. While downloading a larger file, either S-Record or HEX image, it may lead into error message "Encountered an improper argument"</p> <p><u>Workaround:</u></p> <p>Possible workaround to avoid this problem is to use XLINK's pure binary image link functionality. This allows to download code + raw-data at one time and avoids usage of the "Memory" window download functionality.</p>
<b>No. c44</b>	<b>Increased Download Time for V850E2 Devices using the E1 Emulator</b>
	<p><u>Details:</u></p> <p>At specific conditions the download time in case of using V850E2 devices together with the E1 emulator may increase up to 20min depending on the application code size.</p> <p><u>Workaround:</u></p> <p>There is no workaround, but only the recommendation to disable fill up of unused code to minimize the application code size. The issue will be fixed in next service pack scheduled for e/o January 2013.</p>

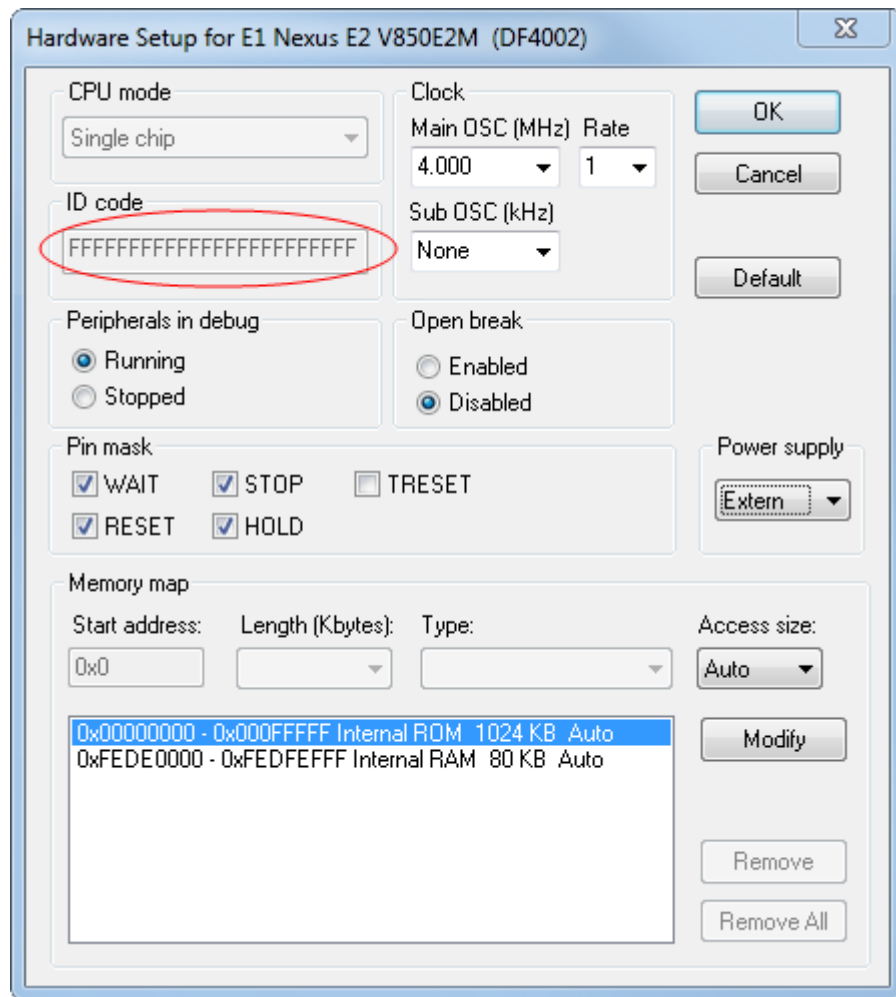
<b>No. c45</b>	<b>Memory overridden if multiple images are loaded</b>
	<p><u>Details:</u></p> <p>The memory maybe overwritten if multiple images are loaded.</p> <p><u>Workaround:</u></p> <p>There is no workaround.</p>
<b>No. c46</b>	<b>IECUBE2 Driver: Trace Buffer Size cannot be reduced</b>
	<p><u>IAR Reference:</u> EW24057</p> <p><u>Details:</u></p> <p>The IECUBE2 trace buffer size is fixed to maximum value of 512KB and cannot be reduced by user.</p> <p><u>Workaround:</u></p> <p>There is no workaround.</p>
<b>No. c47</b>	<b>IDE Crashes due to unintended triggered Interrupt</b>
	<p><u>IAR Reference:</u> EW24057</p> <p><u>Details:</u></p> <p>If an interrupt is repeatedly triggered by mistake, the Embedded Workbench IDE may crash.</p> <p><u>Workaround:</u></p> <p>To be sure that no unintended interrupts are triggered please put a guard on every possible unused interrupt vector.</p>

**No. c48** ID CODE Input disabled in Hardware Setup

IAR Reference: F140428B

Details:

Input of ID code is disabled in hardware setup dialogue:



Workaround:

There is no workaround.

<b>No. c49</b>	IDE Crash changing low or high Byte of 16bit I/O Register
	<p><u>IAR Reference:</u> EW25740</p> <p><u>Details:</u></p> <p>The IDE sometimes crashes if the high or low byte of an I/O Register is modified manually in the Register Window.</p> <p><u>Workaround:</u></p> <p>Change complete I/O register.</p>

<b>No. c50</b>	IDE Crash at Single-Step Command while Live-Watch-Window is open
	<p><u>IAR Reference:</u> EW25121</p> <p><u>Details:</u></p> <p>The IDE sometimes crashes if you issue multiple single-step commands while the Live Watch Window is open.</p> <p><u>Workaround:</u></p> <p>None</p>

**No. c51** Wrong Address area for internal RAM displayed in C-SPY E1 Hardware Setup

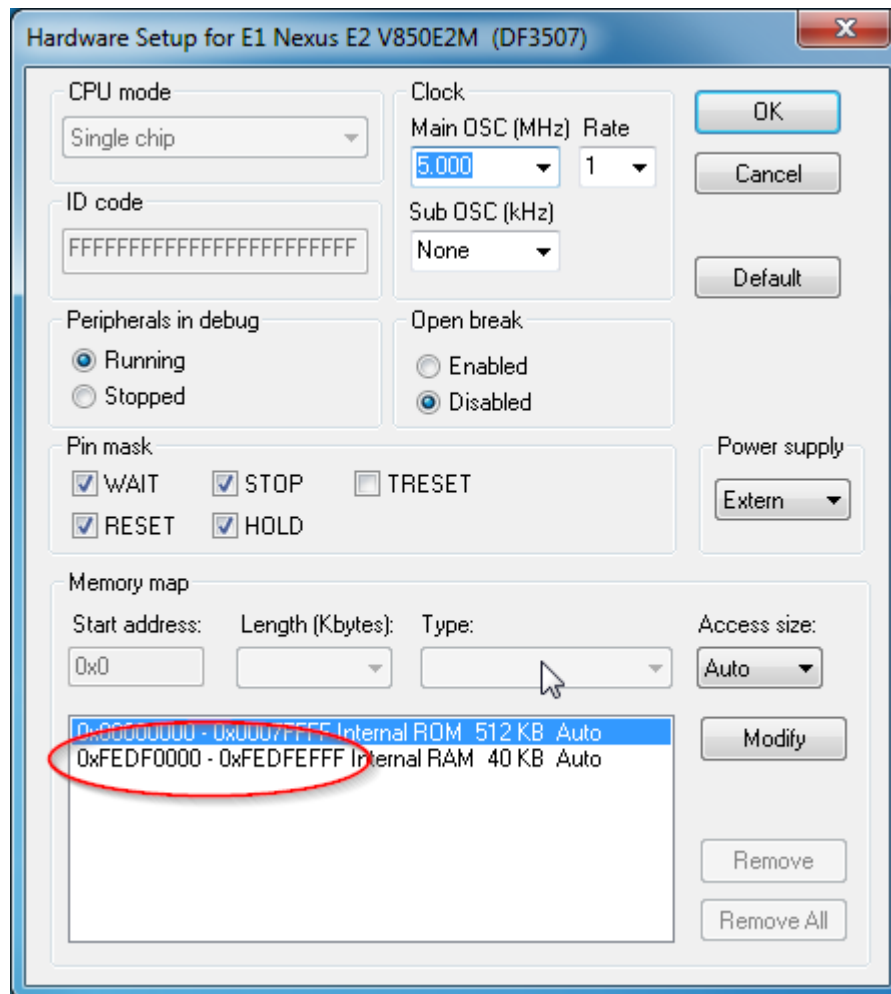
IAR Reference: EW25775

Details:

Wrong address area for the internal RAM is listed in the C-SPY E1 Hardware Setup dialog box.

Example:

Internal RAM  $\mu$ PD70F3507: 0xFEDF6000-0xFEDFFFFFF (40KB)



Workaround:

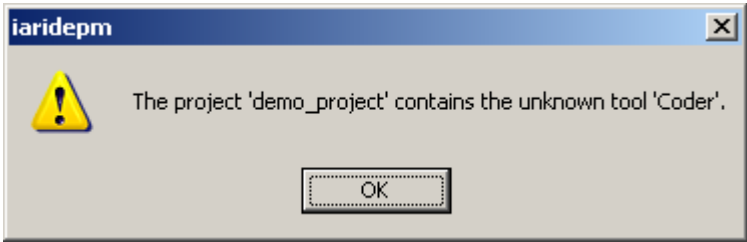
None.

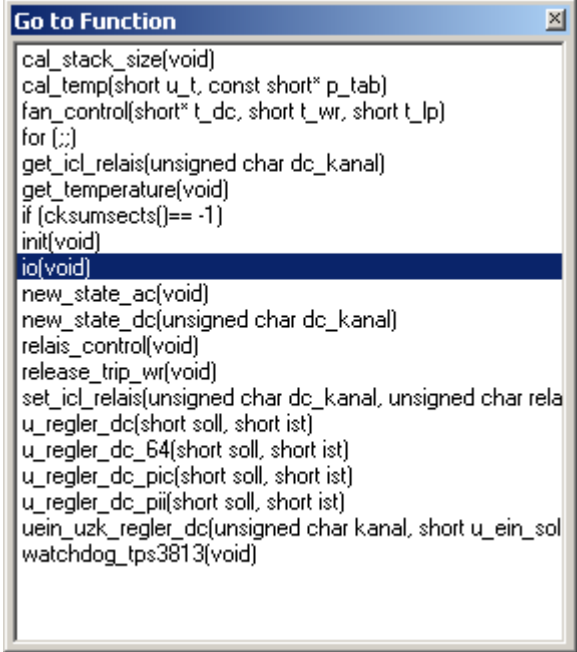
<b>No. c52</b>	<p>The CMOV instruction is not correctly implemented (C-SPY simulator only)</p> <p><u>IAR Reference:</u> EWV850-460</p> <p><u>Details:</u></p> <p>The CMOV instruction is not correctly implemented in the C-SPY simulator.</p> <p><u>Workaround:</u></p> <p>None</p>
----------------	---

**H) Description of Operating Precautions for the IAR Platform (EWV850)**

<b>No. d1</b>	<b>The source Browser window doesn't update properly.</b>
	<p><u>Details:</u></p> <p>After deletion of a file from a project and after new compilation, the file's symbols are still listed in the source browser window.</p> <p><u>Workaround:</u></p> <p>Force a re-collection of the information by turning the option "Tools"-&gt;"Project"-&gt;"Generate browse information" on and off.</p>
<b>No. d2</b>	<b>The heap size under “Project Options” is limited</b>
	<p><u>Details:</u></p> <p>When setting the heap size in the IDE under “Project Options”, only an integer between 0 and 65535 is accepted.</p> <p><u>Workaround:</u></p> <p>Set the Heap size in the linker directive file.</p>



No. d3	Message box: 'The project contains the unknown tool 'Coder''
	<p><u>Details:</u></p> <p>When you open a project file, which was created with an Embedded Workbench version with IAR visualSTATE plug-in, with an Embedded Workbench version without IAR visualSTATE plug-in following message appears:</p>  <p><u>Workarounds:</u></p> <ol style="list-style-type: none"><li>1) Click OK. This setting will not influence your project.</li><li>2) Install the IAR visualSTATE evaluation edition which is available from the IAR website.</li><li>3) If you do not want to install visualSTATE, please edit the IAR project file (.ewp) and delete this block, which appears twice in the project file:</li></ol> <pre>&lt;settings&gt;   &lt;name&gt;Coder&lt;/name&gt;   &lt;archiveVersion&gt;0&lt;/archiveVersion&gt; &lt;/data&gt; &lt;/settings&gt;</pre>

No. d4	<p><b>Missing function names in "Go to function" view</b></p> <p><u>Details:</u></p> <p>Not all functions of a C-module will be shown in the editor "Go to function" view. Instead lines of C-code commands may appear.</p>  <p><u>Workaround:</u></p> <p>None.</p>
No. d5	<p><b>No Source Browse Information generated</b></p> <p><u>Details:</u></p> <p>Source browsing information is not generated if the source code contains more than 100 occurrences of the extended keyword <code>__no_bit_access</code>. As a result the source code pop-up menu functions 'Goto Definition' and 'Goto Declaration' will not work as well as the function list for each module is empty (-&gt; button <code>f0</code> )</p> <p><u>Workaround:</u></p> <p>None. Please install the update patch V4.10.2</p>

No. <b>d6</b>	<b>Actual Linker-MAP-File not automatically updated in Editor</b>
	<p><u>IAR Reference:</u> EW24451</p> <p><u>Details</u></p> <p>Although the option 'Scan for changed Files' is enabled in EW tool options, a linker map file in HTML format is not automatically updated.</p> <p><u>Workarounds</u></p> <p>Use text format or update the file manually.</p>

No. <b>d7</b>	<b>Incorrect End Addresses in XCL File Template</b>
	<p><u>IAR Reference:</u> EW24772</p> <p><u>Details</u></p> <p>Devices with 64 Kbytes internal RAM have an incorrect end address calculated for the BREL_CBASE and BREL_C segments.</p> <p><u>Workarounds</u></p> <p>Correct end addresses manually or use the the XCL file templates included in EWV850 SP V4.80.3 or later.</p>

## I) List of Error messages of the ICE

### Fatal Error

<b>0x0100</b>	<b>Fatal err (communication error)</b> Cannot communicate with ICE. Please confirm the installation of the device driver for the PC interface board. > The driver may not be correctly installed. Reinstall the driver.
<b>0x0101</b>	<b>Fatal err (hostname not found)</b> Cannot find initialization file (expc.ini).
<b>0x0102</b>	<b>Fatal err (net-inf file not found)</b> Host name not found.
<b>0x0103</b>	<b>Fatal err (data send timeout)</b> Data transfer to ICE is timed out. > Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.
<b>0x0104</b>	<b>Fatal err (exec timeout)</b> Data receive from ICE is timed out. > Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.
<b>0x0105</b>	<b>Fatal err (missing device file read)</b> Failed in reading device file (dxxxx.800). > Necessary files may be damaged. Reinstall the device file.
<b>0x0106</b>	<b>Fatal err (illegal receive data)</b> Illegal data received. > Check the power of the in-circuit emulator, cable connections, and setting of the interface board and restart the debugger.
<b>0x0107</b>	<b>Fatal err (illegal pipe handle)</b> Error departure rose with communication with the in circuit emulator.
<b>0x0108</b>	<b>Fatal err (hostname error)</b> Cannot read the network information file just
<b>0x0109</b>	<b>Fatal err (USB communication error)</b> Abnormality occurred in the communication of USB > Please end the debugger, verify the power source of in-circuit-emulator, and the connection cable. Restart the debugger.
<b>0x010a</b>	<b>Fatal err(another EXEC has already operated)</b> The tool that uses EXEC cannot be started simultaneously. > Stop operated tool that uses EXEC.
<b>0x01a0</b>	<b>Fatal err (monitor timeout)</b> Monitor timeout occurred. > Please verify whether there is no abnormality in the signal and the clock pulse of RESET, WAIT and HLDRQ etc.
<b>0x01a1</b>	<b>Fatal err (exec file read err)</b> Failed in reading the exec file.

<b>0x01a2</b>	<b>Fatal err (BK board no connect)</b> Break board is not connected.
<b>0x01a3</b>	<b>Fatal err (EM board no connect)</b> Emulation board is not connected.
<b>0x01a4</b>	<b>Fatal err (illegal board set)</b> Board configuration of ICE is not consistent.
<b>0x01a5</b>	<b>Fatal err (POD/EM1 board no connect)</b> POD/EM1 board is not connected.
<b>0x01a6</b>	<b>Fatal err (exec running)</b> EXEC is running. > Stop operated tool that uses EXEC.
<b>0x01a7</b>	<b>Fatal err (micro program read err)</b> Failed in reading micro program file.
<b>0x01a8</b>	<b>Fatal err (ini file not found)</b> Failed in reading initialization file (expc.ini).
<b>0x01a9</b>	<b>Fatal err (packet send-buffer size over)</b> Packet transmission buffer size over.
<b>0x01aa</b>	<b>Fatal err (RPRM no connect)</b> Terminal emulator is not loaded.
<b>0x01ab</b>	<b>Fatal err (flash F/W file read err)</b> Failed in reading the flash F/W file.
<b>0x01ac</b>	<b>Fatal err (illegal device file)</b> Device file format type error.
<b>0x01ad</b>	<b>Fatal err (old device driver)</b> Device driver is older version. > Install newest driver.
<b>0x01ae</b>	<b>Fatal err (init file error)</b> Failed in reading init file.
<b>0x01b2</b>	<b>Fatal err (ICE firmware is older version)</b> MINICUBE2 firmware is older version. > Install newest MINICUBE firmware
<b>0x0600</b>	<b>Fatal err (buffer allocate err)</b> Not enough memory for buffer. > There is not enough system memory. Close the applications being executed and the open files
<b>0x0601</b>	<b>Fatal err (win32 resource insufficiency)</b> Resource of the operating system becoming insufficient.
<b>0x0c00</b>	<b>Fatal err (Monitor file read error)</b> Monitor file read error. > Necessary files may be damaged. Reinstall that files.
<b>0x0c01</b>	<b>Fatal err (Register function monitor timeout)</b> During access of register, CPU did time out. ➤ Check the clock signal, etc. The register value may not be correct.
<b>0x0c02</b>	<b>Fatal err (Memory function monitor timeout)</b> During access of memory, CPU did time out.

	<p>&gt; Check the HOLD signal, WAIT signal, clock signal, etc. The memory value may not be correct.</p>
<b>0x0c03</b>	<p><b>Fatal err (Sfr function monitor timeout)</b>  During access of sfr register, CPU did time out.  &gt; Check the HOLD signal, WAIT signal, clock signal, etc. The IOR value may not be correct.</p>
<b>0x0ca0</b>	<p><b>Fatal err (Exec I/F abort error)</b>  Cannot communicate with ICE.  &gt; Please confirm the power of ICE, connection of the interface cable, or I/O address of the PC interface board.</p>
<b>0x0ca1</b>	<p><b>Fatal err (monitor file missing)</b>  Monitor file not found.  &gt; Necessary files may be damaged. Reinstall that files.  &gt; Please check that the device corresponds to using emulator.</p>
<b>0x0ca2</b>	<p><b>Fatal err (specified device file doesn't correspond to OCD)</b>  Device files other than OCD emulator was used.  &gt; A device file should check in the thing corresponding to OCD emulator. Or install newest device file.</p>
<b>0x0ca3</b>	<p><b>Fatal err (EXEC is too old)</b>  Unknown flag exists in OCD information part.  Using EXEC is too old.  &gt; Use newest EXEC.</p>
<b>0x0ca4</b>	<p><b>Fatal err (specified device file doesn't correspond to IECUBE)</b>  Device files other than IECUBE was used.  &gt; A device file should check in the thing corresponding to IECUBE.</p>

## User system abnormality

<b>0x0200</b>	<b>User system err (verify err)</b> Verification error occurred. Failed in writing memory. > In case of OCD emulator, MINICUBE2 or IECUBE, please check setting of main and sub- clock in the hardware set-up window
<b>0x02a0</b>	<b>User system err (bus hold)</b> Bus hold error. > CPU is in the bus-hold status. Reset the debugger.
<b>0x02a1</b>	<b>User system err (stand by mode)</b> Standby mode.
<b>0x02a2</b>	<b>User system err (cannot break)</b> Cannot compulsory break.
<b>0x02a3</b>	<b>User system err(reset continuation)</b> Reset under continuation. In case of MINICUBE2, when the RESET pin in MINICUBE2 doesn't become high-level when reset is released, it error occurs. Following causes are thought: *The connection with the target is not correct. *The reset circuit of the target doesn't operate normally.
<b>0x0c20</b>	<b>User system err (Guard area access)</b> Guarded area cannot be accessed.
<b>0x0c21</b>	<b>User system err (NOREADY)</b> Memory was unready status.
<b>0x0c22</b>	<b>User system err (NOREADY cancel break)</b> Memory unready status was canceled.
<b>0x0c23</b>	<b>User system err (bus hold)</b> Bus hold under continuation. > Check the setting of the target board, or mask the HOLD pin.
<b>0x0c24</b>	<b>User system err (break reset error)</b> It cannot shift to debug mode. > Check the clock signal. This may be caused by a stopped clock or a slow clock. > In case of MINICUBE2, it is generated when there is no response from the monitor program after releasing reset.
<b>0x0c25</b>	<b>User system err (mask rom area)</b> Flash macro service ROM was accessed or stepped in.
<b>0x0c26</b>	<b>User system err (FLMD terminal write protect)</b> FLMD terminal is in a write-protected state. > FLMD is not in the write-enabled status. Check the status of the FLMD0 and FLMD1 pins.
<b>0x0c27</b>	<b>User system err(security flag disabled)</b> Security flag is in a write-protected state. > The security flag of the flash memory has disabled writing, block erasure, or chip erasure. Nothing can be written to the flash memory.
<b>0x0c28</b>	<b>User system err(flash write disabled)</b>

	Internal RAM is not enough, the writing to flash memory is not made. > The internal RAM size is less than 4 KB and flash self-programming cannot be executed.
<b>0x0c29</b>	<b>User system err (blank check failed)</b> The blank check of flash memory failed
<b>0x0c2a</b>	<b>User system err (erase failed)</b> The erasing of flash memory failed.
<b>0x0c2b</b>	<b>User system err (write failed)</b> The writing of flash memory failed.
<b>0x0c2c</b>	<b>User system err (verify failed)</b> The internal verification of flash memory failed.
<b>0x0c2d</b>	<b>User system err (PLL lock failed)</b> The writing of flash memory failed because could not lock PLL.
<b>0x0c2e</b>	<b>User system err (no response from flash macro service)</b> No response from flash macro service.
<b>0x0c2f</b>	<b>User system err (return unjust value from flash macro service)</b> Return unjust value from flash macro service.
<b>0x0c30</b>	<b>User system err (necessary to release flash SFR prohibition setting)</b> Necessary to release flash SFR prohibition setting. In case of OCD emulator, MINICUBE2 or IECUBE, please check setting of the main and sub-clock in the hardware set-up window..
<b>0x0c31</b>	<b>User system err (break failed by stop mode)</b> Could not break by stop mode. > Please release STOP mode or reset CPU.
<b>0x0c32</b>	<b>User system err (no support device mode)</b> This device mode is not supported to write. > Please use single-chip-mode 0 to write to Flash memory.
<b>0x0c33</b>	<b>User system err (tried disable on chip debug)</b> Tried disable on chip debug. > Please do not write 0 to the most significant bit of ID code.
<b>0x0c34</b>	<b>User system err (tried to write reserved area)</b> Tried to write reserved area by on chip debug.
<b>0x0c35</b>	<b>User system err (temporary program write error)</b> Temporary program write error > Please check to specify a correct device file.
<b>0x0c36</b>	<b>User system err (IROM size is illegal for flash self emulation)</b> Flash self emulation function cannot be made enable for internal ROM size is set to other than the default size.
<b>0x0c3a</b>	<b>User system err (no support data flash)</b> This device doesn't support Data Flash. > Please confirm whether used device file supports Data Flash.
<b>0x0c3b</b>	<b>User system err (flash environmental is other than data flash)</b> Because it is accessing Code Flash area, it cannot access Data Flash area.



**Status Error**

<b>0x0300</b>	<b>Status err (user program is running)</b> User program is running.
<b>0x0301</b>	<b>Status err (user program not running)</b> User program is being broken.
<b>0x0302</b>	<b>Status err (trace is working)</b> User program is being traced.
<b>0x0303</b>	<b>Status err (no trace data)</b> Not traced.
<b>0x0304</b>	<b>Status err (trace memory invalid)</b> Trace memory is not set.
<b>0x0306</b>	<b>Status err (no trace block)</b> No trace block exists.
<b>0x0307</b>	<b>Status err (illegal event set num)</b> No event condition exists.
<b>0x0308</b>	<b>Status err (no timer measurement)</b> No timer measurement is done.
<b>0x0309</b>	<b>Status err (no trigger frame)</b> No trigger frame exists.
<b>0x030a</b>	<b>Status err (timer off)</b> Tracer is being stopped.
<b>0x030d</b>	<b>Status err (timer on)</b> Timer is running.
<b>0x030e</b>	<b>Status err (mem range err)</b> Memory copy area is overlapped.
<b>0x030f</b>	<b>Status err (already set)</b> Trace has been already set.
<b>0x0310</b>	<b>Status err (no condition event num)</b> Event condition is not set.
<b>0x0311</b>	<b>Status err (full timer num)</b> Too many valid timer event conditions.
<b>0x0312</b>	<b>Status err (no timer num)</b> Specified timer event is not set.
<b>0x0313</b>	<b>Status err (map range err)</b> Illegal map range. > Check the setting in "Memory Mapping (mapping setting area)" in the hardware set-up window
<b>0x0314</b>	<b>Status err (delay event-mode set err)</b> Only trace delay mode can set with delay trigger.
<b>0x0315</b>	<b>Status err (delay mode is full)</b> Delay trigger cannot set without trace delay mode.

<b>0x0316</b>	<b>Status err (over mapping num)</b> Too many valid mapping.
<b>0x03a0</b>	<b>Status err (target power off)</b> Target is not turned on. > Check the target power supply. Check the cable connecting the in-circuit emulator and target board.
<b>0x03a1</b>	<b>Status err (step executive)</b> Step execution is being done.
<b>0x03a2</b>	<b>Status err (realtime measure running)</b> Timer and Tracer are running.
<b>0x03a3</b>	<b>Status err (mixed events specified)</b> Existed together appointed the event.
<b>0x0c40</b>	<b>Status err (invalid address condition)</b> Status of effective event conditions cannot be changed.
<b>0x0c42</b>	<b>Status err (escape break, cannot run)</b> Monitor has failed in shift in the debugging mode. Please reset the CPU.
<b>0x0c43</b>	<b>Status err (emulator access failed)</b> Can not communicate with ICE. (IE-V850E1-CD-NW) Can not communicate with ICE. Please confirm the power of ICE, connection of the interface cable.(IECUBE)
<b>0x0c44</b>	<b>Status err (trace packet data missing)</b> trace packet data missing.
<b>0x0c45</b>	<b>Status error (power off reset effective)</b> Inside of Power off reset emulation cannot carry out program execution.
<b>0x0c46</b>	<b>user system error (flash self emulation effective)</b> It cannot modify internal ROM size and internal RAM size for the flash self emulation function is made enable.
<b>0x0caf</b>	<b>Status err (exceed trace block)</b> Trace block can not be stepped over.

## Parameter Error

<b>0x0400</b>	<b>Param err (illegal data)</b> Illegal condition. > Settings of the used in-circuit emulator and those of the Configuration dialog box may not match.
<b>0x0401</b>	<b>Param err (timer overflow)</b> Result of timer measurement overflowed.
<b>0x0402</b>	<b>Param err (pass counter overflow)</b> Too many event conditions with path count.
<b>0x0403</b>	<b>Param err (address range err)</b> Too many address range conditions.
<b>0x0404</b>	<b>Param err (event num overflow)</b> Too many simultaneously-usable-event conditions.
<b>0x0407</b>	<b>Param err (initial data area overflow)</b> Too many initialization data.
<b>0x0408</b>	<b>Param err (search data overflow)</b> Too large search data (> 16 byte).
<b>0x0409</b>	<b>Param err (search data area overflow)</b> Too large search data (> search range).
<b>0x040a</b>	<b>Param err (sequential overflow)</b> Too many Linking-event conditions.
<b>0x04a0</b>	<b>Param err (trigger event overflow)</b> Too many Trigger-event.
<b>0x04a1</b>	<b>Param err (emulation mem insufficiency)</b> Not enough memory for emulation.
<b>0x04a2</b>	<b>Param err (bus size overflow)</b> Too many partition of bus size.
<b>0x04a3</b>	<b>Param err (Execution event overflow)</b> Too many execution-event conditions.
<b>0x04a4</b>	<b>Param err (Data Access event overflow)</b> Too many bus-event conditions.
<b>0x04a5</b>	<b>Param err (external data overflow)</b> Too many External data.
<b>0x0c60</b>	<b>Param err (invalid event condition)</b> Event before execution cannot be set up other than break conditions.
<b>0x0c61</b>	<b>Param err (Cant use hardware breakpoint)</b> Can not register event numbers which can not be used for hardware break.
<b>0x0c62</b>	<b>Paramerr (event is used hardware breakpoint)</b> Event numbers reserved for hardware breaks can not be used.

## Device Dependent Error

<b>0x0c63</b>	<b>Param err (trace event set error)</b> Event link conditions cannot set.
<b>0x0c64</b>	<b>Param err (rom emulation area error)</b> Too many ROM-emulation-RAM areas.
<b>0x0c65</b>	<b>Param err (event busy)</b> The event which is appointed presently is in the midst of using.
<b>0x0c66</b>	<b>Param err (not emulation mameory area)</b> Emulation memory area was appointed.
<b>0x0c67</b>	<b>Param err (block size error)</b> Writing of flash memory during block is not made.
<b>0x0c69</b>	<b>Param err (data flash area out of range)</b> It tried to access out of Data Flash area. Or it tried to access Data Flash area from other areas. > Please specify address and length to become in Data Flash area. Or please specify address and length not accessed Data Flash area.
<b>0x0c70</b>	<b>Device depend err (DCU access error)</b> There is a possibility of having made a mistake in the selection of the device file. Please select the device file which corresponds to the target chip.
<b>0x0c71</b>	<b>Device depend err (reset request failed)</b> Please verify the clock pulse. You can think clock stop and the low-speed clock. In case of MINICUBE2, When the RESET pin in MINICUBE2 doesn't become low-level when reset is released, it error occurs. Following causes are thought: *The connection with the target is not correct. *The reset circuit of the target doesn't operate normally.
<b>0x0c72</b>	<b>Device depend err (monitor area access failed)</b> Failure of monitor area access. Following causes are thought. *The connection with the target is not correct. *The selection by UART or CSI is wrong. *The operation frequency input in the hardware set-up window and the operation frequency of the device on target are different.
<b>0x0c73</b>	<b>Device depend err (monitor execution failed)</b> Failure of monitor execution. In case of MINICUBE2, When the ID code is changed while executing the user program, it is generated.

**IECUBE Starting Error**

<b>0x0c74</b>	<b>Device depend err (CPU resource access failed)</b> Failure of CPU core resource access > There is a possibility of having made a mistake in the selection of the device file.
<b>0x0c75</b>	<b>Device depend err (illegal debug mode)</b> Transited to illegitimate debugging mode. > Please CPU reset.
<b>0x0c76</b>	<b>Device depend err (DCU access init error)</b> Initial condition when starting the DCU access is abnormal.
<b>0x0c77</b>	<b>Device depend error (DCU access error(verify))</b> Please verify the device file. > Please check the connection of DCK,DMS,DDI,DDO,DRSTZ
<b>0x0c78</b>	<b>Device depend error (trace memory read error)</b> Failed in reading the trace data.

**J) Valid Specification**

Item	Date published	Document No.	Document Title
1	October 2018	UIDEV850-4	IDE Project Management and Building Guide
2	May 2013	CV850-9	V850 IAR C/C++ Compiler Reference Guide
3	October 2010	AV850-4	V850 IAR Assembler Reference Guide
4	October 2010	MV850-2	V850 IAR Embedded Workbench Migration Guide
5	October 2015	UCSV850-2	V850 IAR C-SPY Debugging Guide
6	September 2016	XLINK-650	IAR Linker and Library Tools Reference Guide
7	January 2011	EWMISRAC1998-4	IAR Embedded Workbench MISRA C 1998 Reference Guide
8	January 2011	EWMISRAC:2004-3	IAR Embedded Workbench MISRA C 2004 Reference Guide

## K) Revision History

Item	Date published	Document No.	Comments
1	28-Jan-2005	CESCN0009V01	First release
2	28-Apr-2005	CESCN0009V02	Item a6, Item b1
3	27-May-2005	CESCN0009V03	Item a7, Item c7, c8
4	30-June-1005	CESCN0009V04	Item a8-10
5	8-Aug-2005	CESCN0009V05	Item <a href="#">c9</a> -c10
6	5-Jan-2006	CESCN0009V06	Item a11-c11
7	23-Mar-2006	CESCN0009V07	Item a12-a13, modified c11
8	12-Mai-2006	CESCN0009V08	Added new Compiler release and Item a14-c12
9	13-Jun-2006	CESCN0009V09	Item a15 added – update webpage Links.
10	29-Jun-2006	CESCN0009V10	Item <a href="#">d1</a> added – Update of item c12
11	24-Jul-2006	CESCN0009V11/ U18064EE2V6IF00	Item a16 added. Document was renamed
12	15-Aug-2006	U18064EE3V6IF00	Item a17 added.
13	27-Feb-2007	U18064EE4V6IF00	Items a18, a19, <a href="#">c13</a> , <a href="#">c14</a> , c15 added
14	26-Mar-2007	U18064EE5V6IF00	Item a20 added
15	29-Mar-2007	U18064EE6V6IF00	Corrected symbols “✕✓” in the tables
16	06-Jun-2007	U18064EE7V6IF00	Added new compiler release and items c16 and <a href="#">c17</a> Modified Item c15
17	18-Jul-2007	U18064EE8V6IF00	Item a21, a22, a23, b2, <a href="#">c18</a> and c19 added
18	27-Jul-2007	U18064EE9V6IF00	Item a24 added. Corrected a21 and a23. Compiler patch V3.40b added
19	04-Sep-2007	U18064EEAV6IF00	Item a25 added
19	19-Sep-2007	U18064EEBV6IF00	Items a26, <a href="#">c20</a> added. List of error messages of ICE added
20	25-Oct-2007	U18064EECV6IF00	Items a27, c21, c22 added
21	22-Feb-2008	U18064EEDV6IF00	Items a28, c23 added. C-Spy patch V3.40C added
22	02-May-2008	U18064EEEV6IF00	Items a29, a30, a31, a32, a33 added
23	12-Jun-2008	U18064EEFV6IF00	Items a1-a14, c1-c8,c10-c11 removed Items c24,c25,c26,c27,c28 added Embedded Workbench update EWV850 V3.50a
22	08-Jul-2008	U18064EEGV6IF00	Items <a href="#">a34</a> , <a href="#">a35</a> added
23	01-Oct-2008	U18064EEHV6IF00	Items <a href="#">a36</a> , <a href="#">a37</a> , <a href="#">a38</a> , <a href="#">a39</a> added
24	07-Nov-2008	U18064EEIV6IF00	Items <a href="#">a40</a> and c29 added
25	26-Nov-2008	U18064EEJV6IF00	Items <a href="#">a41</a> , c30 and <a href="#">d2</a> added
26	22-Apr-2009	U18064EEKV6IF00	Items c31, c32, <a href="#">d3</a> added
27	04-May-2009	U18064EELV6IF00	Item <a href="#">a42</a> added

28	20-May-2009	U18064EEMV6IF00	Item <a href="#">a43</a> added
29	18-Jun-2009	U18064EENV6IF00	Item <a href="#">c33</a> , c34 added
30	06-Jul-2009	U18064EEOV6IF00	Item <a href="#">c33</a> modified, Item c35 and <a href="#">a44</a> added
31	30-Nov-2009	U18064EEPv6IF00	Item a45 and <a href="#">c36</a> added
32	25-Jan-2010	U18064EEQV6IF00	Added new compiler release. Items a15, b1, c12, c15 removed. Item <a href="#">c37</a> added.
33	12-Mar-2010	U18064EERV6IF00	Items a16, a17 and a25 removed. Items a46, a47, a48 and a49 added
34	07-April-2011	R20TU0002ED2600	Items a21, a23, a24, c19, c22 and c23 removed. Items a50, <a href="#">b3</a> , <a href="#">b4</a> , <a href="#">c38</a> , <a href="#">c39</a> , <a href="#">c40</a> , c41, c42 and <a href="#">d4</a> added to the document
35	20-August-2012	R20TU0002ED2710	Items a18, a19, a20, a22, a26, a27, a28, a29, a30, a31, a32, a33, a50, b2, c16, c21, c24, c25, c26, c27, c28, c29, c30, c31, c32, c34 and c35 removed. Items <a href="#">a51</a> , <a href="#">a52</a> , <a href="#">a53</a> , <a href="#">a54</a> , <a href="#">a55</a> , <a href="#">a56</a> , <a href="#">b5</a> and <a href="#">c43</a> added to the document
36	19-November-2012	R20TU0002ED2701	Item <a href="#">a57</a> added
37	11-January 2013	R20TU0002ED2702	Item <a href="#">c44</a> added, specification document update
38	16-January 2013	R20TU0002ED2703	Embedded Workbench Update V3.81.4 Item c33, c36, and c37 removed
39	03-June-2013	R20TU0002ED2704	Embedded Workbench Update V4.10.1 Items a34 - a41 removed Item <a href="#">a58</a> , <a href="#">a59</a> and <a href="#">c45</a> added
40	15-July- 2013	R20TU0002ED2705	Service Pack V4.10.2 released Items <a href="#">a60</a> and <a href="#">d5</a> added Items a42, a43, a44 , c38 and c39 removed
41	01-August-2013	R20TU0002ED2706	Items <a href="#">a61</a> and <a href="#">c46</a> added
42	08-November-2013	R20TU0002ED2707	Item <a href="#">c47</a> added.
43	11-February-2014	R20TU0002ED2708	Item <a href="#">d6</a> added.
44	14-April-2014	R20TU0002ED2709	Missing Description of item <a href="#">a59</a> added Update of specification table
45	30-June-2014	R20TU0002ED2710	Service Pack V4.10.3 released Items <a href="#">a62</a> , <a href="#">c48</a> , and <a href="#">d7</a> added Items a45- a49, c41 and c42 removed
46	21-September-2015	R20TU0002ED2711	XLINK V6.3.3.74 update. Item <a href="#">b6</a> added
47	08-October-2015	R20TU0002ED2712	Item <a href="#">c49</a> added
48	04-November-2015	R20TU0002ED2713	Embedded Workbench Update V4.20.1 Items <a href="#">c50</a> and <a href="#">c51</a> added
49	16- March-2016	R20TU0002ED2714	Item <a href="#">a63</a> added
50	05-October-2016	R20TU0002ED2715	Embedded Workbench Update V4.20.2
51	08-January-2019	R20TU0002ED2716	Embedded Workbench Update V5.10.1
52	03-April-2019	R20TU0002ED2717	Embedded Workbench Update V5.10.2 Items <a href="#">a64</a> , <a href="#">a65</a> and <a href="#">c52</a> added



Before using this material, please visit our website to confirm using the most current document available: [Most current version of this document](#)

In case of any technical question related to the Embedded Workbench for V850, please feel free to contact the Renesas [Software-Tool-Support Team](#).

