

Template for comments and secretariat observations

Date: 2018-05-15

Document: ISO/IEC DIS 1539-1

Project: 72320

MB/ NC ¹	Line number (e.g. 17)	Clause/ Subclause (e.g. 3.1)	Paragraph/ Figure/ Table/ (e.g. Table 1)	Type of comment ²	Comments	Proposed change	Observations of the secretariat
JPO 1	[207:4]	11.6.1	3	ed	<p>A FORM TEAM statement includes the effect of executing a SYNC MEMORY statement for the following reasons.</p> <p>The effects of executing a SYNC MEMORY statement are to divide the execution sequence on the executing image into two segments, which is common among all image control statements, and ensure the order of memory operations by the executing image before and after the SYNC MEMORY statement.</p> <p>According to the fourth sentence of paragraph 5 in clause 11.6.9 (page 213, line 15), "The segments that executed before the FORM TEAM statement on an active image of this team precede the segments that execute after the FORM TEAM statement on another active image of this team. "</p> <p>This sentence means that execution of a FORM TEAM statement makes the segments that executed before the statement on all active images of the current team precede the segments that execute after the statement on all active images of this team.</p> <p>Since segment ordering ensures memory operation order, a FORM TEAM statement includes the effect of executing a SYNC MEMORY statement. (A FORM TEAM statement virtually includes the effect of executing a SYNC ALL statement, which includes the effect of executing a SYNC MEMORY statement.)</p>	Delete "FORM TEAM" from the sentence "All image control statements except CRITICAL, END CRITICAL, EVENT POST, EVENT WAIT, FORM TEAM, LOCK, and UNLOCK include the effect of executing a SYNC MEMORY statement (11.6.5).".	
JPO	[207:4]	11.6.1	3	ed	A CRITICAL statement seems to virtually include	Delete "CRITICAL" from the sentence "All image	

¹ **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

² **Type of comment:** **ge** = general **te** = technical **ed** = editorial

Template for comments and secretariat observations

Date: 2018-05-15

Document: ISO/IEC DIS 1539-1

Project: 72320

MB/ NC ¹	Line number (e.g. 17)	Clause/ Subclause (e.g. 3.1)	Paragraph/ Figure/ Table/ (e.g. Table 1)	Type of comment ²	Comments	Proposed change	Observations of the secretariat
2					<p>the effect of executing a SYNC MEMORY statement for the following reasons.</p> <p>In the case that image M executes a CRITICAL construct, image M shall not commence any memory operation in the segment corresponding to the CRITICAL construct before image M executes the CRITICAL statement.</p> <p>According to the fifth sentence of paragraph 3 in clause 11.1.6 (p. 187, line 26) , "Otherwise, if image M completes execution of the construct by failing, and image T is the next to execute the construct, the previous segment on image M precedes the segment on image T."</p> <p>This sentence means that a processor practically has to make all memory operations in the segments before the CRITICAL statement on image M precede any memory operation in the segment corresponding to the CRITICAL construct on image M, because when an image fails is generally unpredictable in advance and it will be impossible to complete delayed memory operations by a failed image in general.</p> <p>Therefore, a processor shall virtually ensure the order of memory operations before and after a CIRITICAL statement by the image that executes the CRITICAL statement. This means that a CRITICAL statement virtually includes the effect of executing a SYNC MEMORY statement.</p>	<p>control statements except CRITICAL, END CRITICAL, EVENT POST, EVENT WAIT, FORM TEAM, LOCK, and UNLOCK include the effect of executing a SYNC MEMORY statement (11.6.5).".</p>	
JP0 3	[215:35] [216:6]	11.6.11	3, 5, 9, 10	ed	<p>The descriptions of the value of the <i>stat-variable</i> are not consistent between the paragraphs:</p> <ul style="list-style-type: none"> - a processor-dependent positive value 	<p>These descriptions should be unified (at least, "positive" should be explicitly stated) unless there are special intentions.</p>	

¹ **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

² **Type of comment:** **ge** = general **te** = technical **ed** = editorial

Template for comments and secretariat observations

Date: 2018-05-15

Document: ISO/IEC DIS 1539-1

Project: 72320

MB/ NC ¹	Line number (e.g. 17)	Clause/ Subclause (e.g. 3.1)	Paragraph/ Figure/ Table/ (e.g. Table 1)	Type of comment ²	Comments	Proposed change	Observations of the secretariat
	[216:27] [216:34]				(in paragraphs 3 and 5) - a processor-dependent positive integer value (in paragraph 9) - a processor-dependent value (in paragraph 10)		
JPO 4	[498:29]	18.5.5.5	2	ed	The underlined part of the following restriction can be misunderstood to mean that the first parameter "dv" of CFI_establish must not be used as a C actual argument associated with a Fortran dummy argument, which seems contrary to the descriptions in Case (i) and Case (ii) in paragraph 5 Examples (page 499). "It shall not have the same value as either a C formal parameter that corresponds to a Fortran actual argument or <u>a C actual argument that corresponds to a Fortran dummy argument.</u> "	It should be clearly described that a pointer to a C descriptor established by CFI_establish can be a C actual argument associated with a Fortran dummy argument.	
JPO 5	[500:23]	18.5.5.7	2	ed	The underlined part of the following restriction can be misunderstood to mean that the first parameter "result" of CFI_section must not become a C actual argument associated with a Fortran dummy argument if the attribute member of the target of "result" does not have the value CFI_attribute_pointer, which seems contrary to the example in C.12.12. "If the value of result is the same as either a C formal parameter that corresponds to a Fortran actual argument or <u>a C actual argument that corresponds to a Fortran dummy argument, the attribute member shall have the value CFI_attribute_pointer.</u> "	It should be clearly described that a pointer to a C descriptor updated by CFI_section whose attribute member does not have the value CFI_attribute_pointer can be a C actual argument associated with a Fortran dummy argument.	

¹ **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

² **Type of comment:** **ge** = general **te** = technical **ed** = editorial

Template for comments and secretariat observations

Date: 2018-05-15

Document: ISO/IEC DIS 1539-1

Project: 72320

MB/ NC ¹	Line number (e.g. 17)	Clause/ Subclause (e.g. 3.1)	Paragraph/ Figure/ Table/ (e.g. Table 1)	Type of comment ²	Comments	Proposed change	Observations of the secretariat
					In the example in paragraph 1 of C.12.12 (page 608-609), the value of the first argument "&array" of CFI_section is the same as that of the C actual argument that corresponds to the Fortran dummy argument of the subroutine set_all, and the attribute member of the C descriptor "array" has the value CFI_attribute_other, not CFI_attribute_pointer.		
JP0 6	[531:18]	A.2	1	ed	There is a typo in the third item: the value of a reference <u>to to</u> a coindexed object on a failed image (5.3.6);	Delete one "to" from the sentence.	
JP0 7	[532:42]	A.2	1	ed	The following item is missing: whether the processor has the ability to detect that an image has failed (11.5, 16.10.2.26);	It should be added.	
JP0 8	[559:12]	C.6.8	2	ed	The closing parenthesis is missing in the following statement: images_spare = MAX (INT (0.01*NUM_IMAGES ()), 0, MIN (NUM_IMAGES () - 10, 1)	It should be: images_spare = MAX (INT (0.01*NUM_IMAGES ()), 0, MIN (NUM_IMAGES () - 10, 1))	
JP0 9	[559:42] [560:9]	C.6.8	2	ed	The statements "EXIT simulation" on the 3rd line from the bottom of p.559 and on the 9th line of p.560 violate the restriction C1167 because they belong to the CHANGE TEAM construct.	A GOTO statement branching to the statement "END TEAM simulation" can be used instead. In the latter case, "EXIT iter" can also be used alternatively.	
JP1 0	[584:15]	C.10.3	1	ed	The dummy argument "callback" is not declared in the subroutine "forward_invoke".	The following statement should be added in the specification-part of the subroutine: TYPE(callback_record), POINTER :: callback	

¹ **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

² **Type of comment:** **ge** = general **te** = technical **ed** = editorial