page xviii	Introduction	para 2	ed	The requirement that the DIM argument to THIS_IMAGE be of default kind has been dropped but this is not mentioned amongst the changes described in the Introduction	Under bullet point "Intrinsic procedures and modules", following the sentence "In references to the intrinsic functions ALL, ANY, … a present optional dummy argument.", add a new sentence "The actual argument DIM to THIS_IMAGE is no longer required to be of default kind.".
page xix	Introduction	para 2	ed	The addition of a new interoperable type is not mentioned in the introduction.	Under bullet point "Features previously described by ISO/IEC TS 29113:2012", following the sentence "The FPTR argument to C_F_PROCPOINTER ... pointer." add a new sentence "There is a new interoperable type C_PTRDIFF_T.".
page 18	3.112.4		ed	Clarification: It was not intended that a generic in an intrinsic module should be classed as module procedure	Insert "specific" before "procedure provided by an intrinsic module". See also edit to subclause 15.2.2.2.
page 164	10.1.11	para 2	ed	There is a formatting error in paragraph 2. Paragraph 2 continues after the numbered list and the text does not become a new paragraph.	Remove label "3" and renumber the following paragraphs.
page 164	10.1.11	para 2/3	ed	Clarification: The text ", and where any final subroutine that is invoked is pure" is redundant and potentially confusing since this situation cannot occur (ref C1583).	Delete ", and where any final subroutine that is invoked is pure".
page 166	10.1.12	para 1 item (6)	ed	Functions GET_TEAM and TEAM_NUMBER should be added to the list of exceptions since the function value is not constant.	Insert "GET_TEAM," after "COMMAND_ARGUMENT_COUNT," and insert "TEAM_NUMBER," after "NUM_IMAGES,".
page 192	11.1.7.5	para 2	ed	There is a typographical error in the third line from the bottom of the page.	After "PROTECTED, SAVE" insert a comma.
pages 199-200	11.1.10.1	para 1	ed	Specifying a rank that is not supported by the processor in a <select-rank-case-stmt> is harmless because the statement would not be reached. It should be allowed so that a program can be written that is portable to a processor that does support the rank.	In constraint C1151 delete "and less than or equal to the maximum possible rank of selector". After paragraph 2 insert a new paragraph 3: The <scalar-int-constant-expr> in a <select-rank-case-stmt> may have a value greater than the maximum possible rank of selector; in this case, its block will never be executed."
page 266	13.3.1	constraint C1302	ed	The list of edit descriptors for which an optional comma may be omitted after a P edit descriptor which is shown in 13.3.1 is not consistent with the list in 13.8.5. This appears to be an oversight.	In constraint C1302, after "ES," add "EX,".
page 269	13.4	para 5	ed	Edit descriptor EX has inadvertently been omitted from the list of descriptors which relate to a real variable.	After "ES," insert "EX,".
page 271	13.7.2.1	para 1 item (3)	ed	Correction:The present wording is incorrect for negative zero (if the processor distinguishes between positive and negative zero).	Change "positive or zero" to "nonnegative".
page 272	13.7.2.3.1	para 1	ed	As for 13.4 above.	After "ES," insert "EX," twice.
page 273	13.7.2.3.2	para 7	ed	Clarification: The description of a hexadecimal-significand input field does not account for trailing	In the sentence beginning "Embedded blanks", following "number", insert "; trailing blanks are ignored".

				blanks.	
page 282	13.8.4	para 3	ed	The sign edit mode affects also list-directed and namelist editing. Hence the first sentence is incorrect. It is also redundant since the edit mode is properly specified elsewhere. For clarification it should be stated that sign mode descriptors are permitted on input but have no effect.	Delete the sentence “The SS, SP, and S edit descriptors affect only I, F, E, EN, ES, D, and G editing during the execution of an output statement”. In the second sentence, between “and S edit descriptors” and “have no effect” insert “are permitted but”.
page 283	13.8.7	para 1	ed	As for 13.4 above.	After “ES,” insert “EX,”.
page 283	13.8.8	para 2	ed	The decimal edit mode affects also list-directed and namelist editing. Hence the second sentence is incorrect. It is also redundant since the edit mode is properly specified elsewhere. Further, “EX” is missing from the list.	Delete the sentence “The decimal edit mode affects only D, E, EN, ES, F, and G editing.”.
page 299	15.2.2.2	para 3	ed	As for subclause 3.112.4 above.	After "A module procedure is" insert "a specific procedure provided by an intrinsic module or", making the whole paragraph read "A module procedure is a specific procedure provided by an intrinsic module or a procedure that is defined by a module subprogram."
page 328	15.6.2.1	para 6	ed	The restriction defined in subclause 18.3.7 para 2 item (6) does not appear in the constraints on interoperable procedure interfaces in subclause 15.6.2.1.	After constraint C1556 add a new constraint: C1556a If <proc-language-binding-spec> is specified for a procedure, each dummy argument of type CHARACTER with the ALLOCATABLE or POINTER attributes shall have deferred character length.”
page 390	16.9.97 IMAGE_INDEX	para 2	ed	Correction. IMAGE_INDEX cannot be an inquiry function, which would be allowed in a constant expression, because its value depends on the number of images in the current team (or a team specified by TEAM or TEAM_NUMBER). For example, if X is declared thus REAL X[*] IMAGE_INDEX(X,[18]) has the value 18 if the number of images in the team is more than 17, and zero otherwise.	Change “Inquiry function” to “Transformational function”.
page 390	16.9.97 IMAGE_INDEX	para 6	ed	Clarification. Subclause 5.3.4 requires that an image index be not greater than the number of images in a team.	After “on any image” add “, provided the number of images is at least 213”
page 416	16.9.155 RANDOM_INIT	REPEAT-ABLE	ed	In the definition of argument REPEATABLE, the last sentence says "If it has the value false, the seed is set to a processor-dependent, unpredictably different value on each call." The inclusion of the word "unpredictably" has the effect of forbidding a vendor from documenting how their pseudorandom number generator works. We believe that this was not intended.	In line 4 of the definition delete "unpredictably".

page 416	16.9.155 RANDOM_INIT	REPEAT-ABLE	ed	In the current definition, it is unclear whether the value of the seed is the same only for the current execution or is repeatable over different runs on the same processor with the same number of images. We think the latter was intended, which is very useful when testing or verifying code.	At the end of the third sentence, add "on the same processor and the number of images is the same".
page 416	16.9.155 RANDOM_INIT	paras 3 & 4	te	Subclause 16.7, para 4 says "It is processor dependent whether each image uses a separate random number generator, or if some or all images use common random number generators." We believe that the reason for this wording is to allow implementations to use a separate seed on each image, a single seed on each set of images (perhaps those on single nodes) or a single seed on all images (perhaps on a shared-memory machine with a modest number of images). The present wording for IMAGE_DISTINCT makes sense only for the case of a separate seed on each image - which image of a set sharing a seed calls RANDOM_INIT has no bearing. Does a single call of RANDOM_INIT initialize all the seeds? The current wording suggests that only one seed is initialized. Subclause 16.7, para 4 also says "If RANDOM_INIT or RANDOM_SEED is called in a segment A, and RANDOM_INIT, RANDOM_SEED, or RANDOM_NUMBER is called in segment B, then segments A and B shall be ordered." If only one seed is initialized, the calls must be in a CRITICAL construct or its equivalent, which would be very inefficient. It would be much better to say that a single call initializes all the seeds and that it does not matter which image makes the call.	In the description of the REPEATABLE argument, in the third sentence delete “from the same image”. In the description of the IMAGE_DISTINCT argument, delete the third and fourth sentences, ("If it has the value true ... RANDOM_INIT.") and replace them by "If it has the value true, the seeds on the images of the current team that use separate random number generators and on the sets of images of the current team that use common random number generators are set to values that are all different. If it has the value false, these seeds are all set to the same value". In paragraph 4 (Example) replace the sentence by “The following statement initializes the pseudorandom number generator so that all the seeds are different:”
page 435	16.9.197 UCOBOUND	para 5	te	UCOBOUND returns a misleading result for any coarray that was established outside the team. The final upper cobound returned by UCOBOUND should be the largest value that the final cosubscript can have in a valid set of cosubscripts. We require the correct value to be produced by NUM_IMAGES, so should do the same here. The change does not affect the efficiency of CHANGE TEAM or the mapping of a cosubscript list to an image index.	Change "the team current when COARRAY was established" to "the current team".
page 443	17.1	para 2		IEEE_SUPPORT_FLAG, and IEEE_SUPPORT_HALTING are incorrectly described as inquiry functions. Also add a clarification and remove tautologous text.	After “Which other exceptions are supported” insert “in the scoping unit”. Delete the first two instances of “inquiry”. Delete “and return true from IEEE_SUPPORT_FLAG(IEEE_UNDERFLOW, X)”.

					Delete “and return true from the corresponding inquiry function”. Delete “and return true from IEEE_SUPPORT_HALTING(FLAG)”.
page 443	17.1	para 4	ed	Correct the wording for support of rounding modes and remove redundant text.	Delete “and return true from IEEE_SUPPORT_DATATYPE(X) (17.11.48)”. Delete “and return true from the corresponding inquiry function”. After “In the case of IEEE_ROUNDING,” change “return true for” to “support”.
page 446	17.3	para 10	ed	The second sentence “The inquiry function … supported” is already stated (and more correctly after the proposed edit above) in subclause 17.1 paragraph 2.	Delete the sentence ““The inquiry function … supported.”.
page 447	17.4	para 4	ed	Support for rounding is already covered in subclause 17.1 paragraph 4.	Delete the sentence ““The inquiry function IEEE_SUPPORT_ROUNDING… mode.”.
page 447	17.4	para 4	ed	Correction: IEEE rounding modes are separate from I/O rounding modes.	Delete the sentence, “The inquiry function IEEE_SUPPORT_IO… 2011.”.
page 448	17.6	para 1	ed	The function of IEEE_SUPPORT_HALTING has already been described in subclause 17.1 paragraph 2.	Delete the sentence, “The inquiry function IEEE_SUPPORT_HALTING… available.”.
pages 490-491	18.3.7	paras 5 to 7 and NOTE 18.22	ed	The term "simply contiguous dummy variable" is undefined. Moreover as described in NOTE 18.22 it is inconsistent with the current standard.	In paragraph 5 change "simply contiguous dummy argument" to "dummy argument which has the CONTIGUOUS attribute, or is an assumed-length CHARACTER array and not a pointer, assumed-shape, or assumed-rank,". Make the same change in paragraph 6. Delete paragraph 7. Delete NOTE 18.22 Add a new NOTE 18.22: "If an interoperable C procedure whose Fortran interface has a dummy argument which has the CONTIGUOUS attribute, or is an assumed-length CHARACTER array and not a pointer, assumed-shape, or assumed-rank, is invoked from C, the invoking routine is responsible for the contents of the C descriptor which therefore might not describe a contiguous data object."
page 492	18.5.3	para 2	ed	Correction. The definition of “base_addr” is self-contradictory. The value of an object can be defined simultaneously to be a null pointer and not a null pointer.	In the description of base_addr change “null pointer. If the object has zero size” to “null pointer; otherwise, if the object has zero size”.
page 492	18.5.3	para 2	ed	Correction. The definition of “version” is incomplete. It does not allow for different procedures in a program being processed with different versions, nor for a structure being defined by Fortran.	Delete the description of version, i.e. “The value is equal … descriptor was established.” and replace it by “If the descriptor was established in a C function, it is the value of CFI_VERSION in ISO_Fortran_binding.h when that source file was translated. Otherwise, it is the version of ISO_Fortran_binding.h with which the descriptor established by Fortran is compatible.”.
page 494	18.5.4	para 8	ed	Clarification. The term “intrinsic type” is not defined in	Change "Otherwise, the value for an intrinsic type shall be positive." to "Otherwise,

				the C standard. More precise wording is proposed.	the value of a macro listed in Table 18.4 is positive."
page 498	18.5.5.5	para 2	ed	Correction. A type cannot be equal to a macro but it can have the value of a type code, which we believe is what was intended.	In the definition of parameter "type" change “shall be one of” to “shall have the value of one of”.
page 498	18.5.5.5	para 2	te	As defined, "type" does not cover all possible interoperable types.	In the definition of parameter "type", after “18.4” insert “or a positive value corresponding to an interoperable C type”.
page 499	18.5.5.5	para 2	ed	Typographical error.	In the definition of parameter "elem_len", change “the type” to “type” in code font.
page 543	C.1	para 1	ed	In the bullet point beginning “All transformational functions”, references to functions from the modules IEEE_ARITHMETIC and IEEE_EXCEPTIONS are inappropriate. Some of them were mis-characterized in Fortran 2003 as inquiry functions and were re-classified as transformational functions in Fortran 2008 Technical Corrigendum 4. Allowing these transformational functions in constant and specification expressions involved no change to the language because they were previously inquiry functions and therefore allowed in such expressions. In Fortran 2003, it was unstated whether the functions in ISO_C_BINDING were pure or not (pure function references being permitted in specification expressions). This issue was not addressed until Fortran 2008. So this reference is correct.	In the bullet point beginning “All transformational functions” change “modules… modules” to “module”, delete the following “IEEE_ARITHMETIC… and,” making the whole bullet point read: “All transformational functions from the intrinsic module ISO_C_BINDING can be used in specification expressions.”