PD ISO/IEC TS 18508:2015

BSI Standards Publication

Information technology — Additional Parallel Features in Fortran

bsi.

...making excellence a habit.™

Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

National foreword

This Published Document is the UK implementation of ISO/IEC TS 18508:2015.

The UK participation in its preparation was entrusted to Technical Committee IST/5, Programming languages, their environments and system software interfaces.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2015.
Published by BSI Standards Limited 2015

ISBN 978 0 580 90628 2
ICS 35.060

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 December 2015.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

TECHNICAL SPECIFICATION

ISO/IEC TS 18508

First edition
2015-12-01

Information technology — Additional Parallel Features in Fortran

Technologies de l'information — Caractéristiques parallèles supplémentaires en Fortran

Reference number
ISO/IEC TS 18508:2015(E)

Documents
Document Sharing Hub

© ISO/IEC 2015

Full document details: PD ISO/IEC TS 18508:2015

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

Contents

1 Scope . . . 1

2 Normative references . . . 3

3 Terms and definitions . . . 5

4 Compatibility . . . 7
4.1 New intrinsic procedures . . . 7
4.2 Fortran 2008 compatibility . . . 7

5 Teams of images . . . 9
5.1 Team concepts . . . 9
5.2 TEAM_TYPE . . . 9
5.3 CHANGE TEAM construct . . . 9
5.4 Image selectors . . . 11
5.5 FORM TEAM statement . . . 12
5.6 SYNC TEAM statement . . . 13

6 Failed images . . . 15
6.1 Introduction . . . 15
6.2 FAIL IMAGE statement . . . 15
6.3 CRITICAL construct . . . 16
6.4 STAT_FAILED_IMAGE . . . 16

7 Events . . . 17
7.1 Introduction . . . 17
7.2 EVENT_TYPE . . . 17
7.3 EVENT POST statement . . . 18
7.4 EVENT WAIT statement . . . 18

8 Intrinsic procedures . . . 19
8.1 General . . . 19
8.2 Atomic subroutines . . . 19
8.3 Collective subroutines . . . 20
8.4 New intrinsic procedures . . . 21
8.4.1 ATOMIC_ADD (ATOM, VALUE [, STAT]) . . . 21
8.4.2 ATOMIC_AND (ATOM, VALUE [, STAT]) . . . 21
8.4.3 ATOMIC_CAS (ATOM, OLD, COMPARE, NEW [, STAT]) . . . 22
8.4.4 ATOMIC_FETCH_ADD (ATOM, VALUE, OLD [, STAT]) . . . 22
8.4.5 ATOMIC_FETCH_AND (ATOM, VALUE, OLD [, STAT]) . . . 22
8.4.6 ATOMIC_FETCH_OR (ATOM, VALUE, OLD [, STAT]) . . . 23
8.4.7 ATOMIC_FETCH_XOR (ATOM, VALUE, OLD [, STAT]) . . . 23
8.4.8 ATOMIC_OR (ATOM, VALUE [, STAT]) . . . 24
8.4.9 ATOMIC_XOR (ATOM, VALUE [, STAT]) . . . 24
8.4.10 CO_BROADCAST (A, SOURCE_IMAGE [, STAT, ERRMSG]) . . . 24
8.4.11 CO_MAX (A [, RESULT_IMAGE, STAT, ERRMSG]) . . . 25
8.4.12 CO_MIN (A [, RESULT_IMAGE, STAT, ERRMSG]) . . . 25
8.4.13 CO_REDUCE (A, OPERATOR [, RESULT_IMAGE, STAT, ERRMSG]) . . . 26

© ISO/IEC 2015 – All rights reserved

AVA Documents
Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

8.4.14 CO_SUM (A [, RESULT_IMAGE, STAT, ERRMSG]) . . . 27
8.4.15 EVENT_QUERY (EVENT, COUNT [, STAT]) . . . 27
8.4.16 FAILED_IMAGES ([TEAM, KIND]) . . . 28
8.4.17 GET_TEAM ([LEVEL]) . . . 28
8.4.18 IMAGE_STATUS (IMAGE, [TEAM]) . . . 29
8.4.19 STOPPED_IMAGES ([TEAM, KIND]) . . . 29
8.4.20 TEAM_NUMBER ([TEAM]) . . . 30
8.5 Modified intrinsic procedures . . . 31
8.5.1 ATOMIC_DEFINE and ATOMIC_REF . . . 31
8.5.2 IMAGE_INDEX . . . 31
8.5.3 MOVE_ALLOC . . . 31
8.5.4 NUM_IMAGES . . . 32
8.5.5 THIS_IMAGE . . . 32

9 Required editorial changes to ISO/IEC 1539-1:2010(E) . . . 33
9.1 General . . . 33
9.2 Edits to Introduction . . . 33
9.3 Edits to clause 1 . . . 33
9.4 Edits to clause 2 . . . 34
9.5 Edits to clause 4 . . . 35
9.6 Edits to clause 6 . . . 35
9.7 Edits to clause 8 . . . 36
9.8 Edits to clause 9 . . . 39
9.9 Edits to clause 13 . . . 39
9.10 Edits to clause 16 . . . 43
9.11 Edits to annex A . . . 44
9.12 Edits to annex C . . . 44

Annex A (informative) Extended notes . . . 45
A.1 Clause 5 notes . . . 45
A.1.1 Example using three teams . . . 45
A.1.2 Accessing coarrays in sibling teams . . . 45
A.1.3 Reducing the codimension of a coarray . . . 46
A.2 Clause 6 notes . . . 47
A.2.1 Example involving failed images . . . 47
A.3 Clause 7 notes . . . 49
A.3.1 EVENT_QUERY example . . . 49
A.3.2 EVENT_QUERY example that tolerates image failure . . . 50
A.3.3 EVENTS example . . . 52
A.4 Clause 8 notes . . . 53
A.4.1 Collective subroutine examples . . . 53
A.4.2 Atomic memory consistency . . . 53

© ISO/IEC 2015 – All rights reserved
AVA Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and nongovernmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

In other circumstances, particularly when there is an urgent market requirement for such documents, the joint technical committee may decide to publish an ISO/IEC Technical Specification (ISO/IEC TS), which represents an agreement between the members of the joint technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/IEC TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/IEC TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TS 18508:2015 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC22, *Programming languages, their environments and system software interfaces*.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

Introduction

The system for parallel programming in Fortran, as standardized by ISO/IEC 1539-1:2010, defines simple syntax for access to data on another image of a program, synchronization statements for controlling the ordering of execution segments between images, and collective allocation and deallocation of memory on all images.

The existing system for parallel programming does not provide for an environment where a subset of the images can easily work on part of an application while not affecting other images in the program. This complicates development of independent parts of an application by separate teams of programmers. The existing system does not provide a mechanism for a processor to identify what images have failed during execution of a program. This adversely affects the resilience of programs executing on large systems. The synchronization primitives available in the existing system do not provide a convenient mechanism for ordering execution segments on different images without requiring that those images arrive at a synchronization point before either is allowed to proceed. This introduces unnecessary inefficiency into programs. Finally, the existing system does not provide intrinsic procedures for commonly used collective and atomic memory operations. Intrinsic procedures for these operations can be highly optimized for the target computational system, providing significantly improved program performance.

This Technical Specification extends the facilites of Fortran for parallel programming to provide for grouping the images of a program into nonoverlapping teams that can more effectively execute independently parts of a larger problem, for the processor to indicate which images have failed during execution and allow continued execution of the program on the remaining images, for a system of events that can be used for fine grain ordering of execution segments, and for collective and atomic memory operation subroutines that can provide better performance for specific operations involving more than one image.

The facility specified in this Technical Specification is a compatible extension of Fortran as standardized by ISO/IEC 1539-1:2010, ISO/IEC 1539-1:2010/Cor 1:2012, and ISO/IEC 1539-1:2010/Cor 2:2013.

It is the intention of ISO/IEC JTC 1/SC22 that the semantics and syntax specified by this Technical Specification be included in the next revision of ISO/IEC 1539-1 without change unless experience in the implementation and use of this feature identifies errors that need to be corrected, or changes are needed to achieve proper integration, in which case every reasonable effort will be made to minimize the impact of such changes on existing implementations.

This Technical Specification is organized in 8 clauses:

Scope	Clause 1
Normative references	Clause 2
Terms and definitions	Clause 3
Compatibility	Clause 4
Teams of images	Clause 5
Failed images	Clause 6
Events	Clause 7
Intrinsic procedures	Clause 8
Required editorial changes to ISO/IEC 1539-1:2010(E)	Clause 9

It also contains the following nonnormative material:

Extended notes	Annex A

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

1 Scope

This Technical Specification specifies the form and establishes the interpretation of facilities that extend the Fortran language defined by ISO/IEC 1539-1:2010, ISO/IEC 1539-1:2010/Cor 1:2012, and ISO/IEC 1539-1:2010/Cor 2:2013. The purpose of this Technical Specification is to promote portability, reliability, maintainability, and efficient execution of parallel programs written in Fortran, for use on a variety of computing systems.

This Technical Specification does not specify formal data consistency model. Developing the formal data consistency model is left until the integration of these facilities into ISO/IEC 1539-1.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

(Blank page)

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

2 Normative references

The following referenced standards are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 1539-1:2010, *Information technology—Programming languages—Fortran—Part 1:Base language*

ISO/IEC 1539-1:2010/Cor 1:2012, *Information technology—Programming languages—Fortran—Part 1:Base language TECHNICAL CORRIGENDUM 1*

ISO/IEC 1539-1:2010/Cor 2:2013, *Information technology—Programming languages—Fortran—Part 1:Base language TECHNICAL CORRIGENDUM 2*

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

(Blank page)

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 1539-1:2010 and the following apply. The intrinsic module ISO_FORTRAN_ENV is extended by this Technical Specification.

3.1
active image
image that has not failed or initiated termination

3.2
asynchronous progress
ability of an image to reference or define a coarray on another image without waiting for that image to execute any particular statement or class of statement

3.3
established coarray
⟨in a team⟩ coarray that is accessible using a coindexed designator (5.1)

3.4
team
set of images that can readily execute independently of other images (5.1)

3.4.1
current team
⟨of an image⟩ team specified by the most recently executed CHANGE TEAM statement of an active CHANGE TEAM construct, or initial team if no CHANGE TEAM construct is active (5.1)

3.4.2
initial team
team, consisting of all images, that began execution of the program (5.1)

3.4.3
parent team
⟨of a team⟩ current team during the execution of the CHANGE TEAM statement that established the team (5.1)

3.4.4
team number
integer value identifying a team within its parent team (5.1)

3.5
failed image
image that has ceased participating in program execution but has not initiated termination (6.1)

3.6
stopped image
image that has initiated normal termination

3.7
event variable
scalar variable of type EVENT_TYPE (7.2) in the intrinsic module ISO_FORTRAN_ENV

3.8
team variable
scalar variable of type TEAM_TYPE (5.2) in the intrinsic module ISO_FORTRAN_ENV

© ISO/IEC 2015 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

(Blank page)

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

4 Compatibility

4.1 New intrinsic procedures

This Technical Specification defines intrinsic procedures in addition to those specified in ISO/IEC 1539-1:2010. Therefore, a Fortran program conforming to ISO/IEC 1539-1:2010 might have a different interpretation under this Technical Specification if it invokes an external procedure having the same name as one of the new intrinsic procedures, unless that procedure is specified to have the EXTERNAL attribute.

4.2 Fortran 2008 compatibility

This Technical Specification specifies an upwardly compatible extension to ISO/IEC 1539-1:2010, as modified by ISO/IEC 1539-1:2010/Cor 1:2012 and ISO/IEC 1539-1:2010/Cor 2:2013.

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub

Full document details: PD ISO/IEC TS 18508:2015

(Blank page)

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

5 Teams of images

5.1 Team concepts

A team of images is a set of images that can readily execute independently of other images. Syntax and semantics of *image-selector* (R624 in ISO/IEC 1539-1:2010) are extended to determine how cosubscripts are mapped to image indices for sibling or ancestor team references. Initially, the current team consists of all images and this is known as the initial team. Except for the initial team, every team has a unique parent team. A team is divided into new teams by executing a FORM TEAM statement. Each new team is identified by an integer value known as its team number. Information about the team to which the current image belongs can be determined by the processor from the collective value of the team variables on the images of the team.

During execution, each image has a current team, which is only changed by execution of CHANGE TEAM and END TEAM statements. Executing a CHANGE TEAM statement changes the current team to the team specified by the *team-variable*, and execution of the corresponding END TEAM statement restores the current team back to that immediately prior to execution of the CHANGE TEAM statement.

A nonallocatable coarray that is neither a dummy argument, host associated with a dummy argument, declared as a local variable of a subprogram, nor declared in a BLOCK construct is established in the initial team. An allocated allocatable coarray is established in the team in which it was allocated. An unallocated allocatable coarray is not established. A coarray that is an associating entity in a *coarray-association* of a CHANGE TEAM statement is established in the team of its CHANGE TEAM construct. A nonallocatable coarray that is a dummy argument or host associated with a dummy argument is established in the team in which the procedure was invoked. A nonallocatable coarray that is a local variable of a subprogram or host associated with a local variable of a subprogram is established in the team in which the procedure was invoked. A nonallocatable coarray declared in a BLOCK construct is established in the team in which the BLOCK statement was executed. A coarray dummy argument is not established in any ancestor team even if the corresponding actual argument is established in one or more of them.

5.2 TEAM_TYPE

TEAM_TYPE is a derived type with private components. It is an extensible type with no type parameters. Each component is fully default-initialized. A scalar variable of this type describes a team. TEAM_TYPE is defined in the intrinsic module ISO_FORTRAN_ENV.

A scalar variable of type TEAM_TYPE is a team variable. The default initial value of a team variable shall not represent any valid team.

5.3 CHANGE TEAM construct

The CHANGE TEAM construct changes the current team to which the executing image belongs.

R501 *change-team-construct* **is** *change-team-stmt*
block
end-change-team-stmt

R502 *change-team-stmt* **is** [*team-construct-name*:] CHANGE TEAM (*team-variable* ■
■ [, *coarray-association-list*] [, *sync-stat-list*])

R503 *coarray-association* **is** *codimension-decl* => *coselector-name*

© ISO/IEC 2015 – All rights reserved

Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

R504 *end-change-team-stmt* **is** END TEAM [(*sync-stat-list*)] [*team-construct-name*]

R505 *team-variable* **is** *scalar-variable*

C501 (R501) A branch within a CHANGE TEAM construct shall not have a branch target that is outside the construct.

C502 (R501) A RETURN statement shall not appear within a CHANGE TEAM construct.

C503 (R501) An *exit-stmt* or *cycle-stmt* within a CHANGE TEAM construct shall not belong to an outer construct.

C504 (R501) If the *change-team-stmt* of a *change-team-construct* specifies a *team-construct-name*, the corresponding *end-change-team-stmt* shall specify the same *team-construct-name*. If the *change-team-stmt* of a *change-team-construct* does not specify a *team-construct-name*, the corresponding *end-change-team-stmt* shall not specify a *team-construct-name*.

C505 (R503) The *coarray-name* in the *codimension-decl* shall not be the same as any *coselector-name* in the *change-team-stmt* or the same as a *coarray-name* in another *codimension-decl* in the *change-team-stmt*.

C506 (R505) A *team-variable* shall be of type TEAM_TYPE (5.2).

C507 (R502) No *coselector-name* shall appear more than once in a *change-team-stmt*.

C508 (R503) A *coselector-name* shall be the name of an accessible coarray.

The values of the *team-variable*s on the active images of the team shall be those of team variables defined by execution of the same FORM TEAM statement on all active images of the team or be the values of team variables for the initial team. The current team for the statements of the CHANGE TEAM *block* is the team specified by the value of the *team-variable*. The current team is not changed by a redefinition of the team variable during execution of the CHANGE TEAM construct. A CHANGE TEAM construct completes execution by executing its END TEAM statement.

A coselector name identifies a coarray. The coarray shall be established when the CHANGE TEAM statement begins execution. A *codimension-decl* in a *coarray-association* associates a coarray with an established coarray during the execution of the block. This coarray is an associating entity (8.1.3.2, 8.1.3.3, 16.5.1.6 of ISO/IEC 1539-1:2010). Its name is an associate name that has the scope of the construct. It has the declared type, dynamic type, type parameters, rank, and bounds of the established coarray. Its corank and cobounds are those specified in the *codimension-decl*.

Within a CHANGE TEAM construct, a coarray that is not an associating entity has the corank and cobounds that it had when it was established.

An allocatable coarray that was allocated when execution of a CHANGE TEAM construct began shall not be deallocated during execution of the construct. An allocatable coarray that is allocated when execution of a CHANGE TEAM construct completes is deallocated if it was not allocated when execution of the construct began.

The CHANGE TEAM and END TEAM statements are image control statements. All active images of the current team shall execute the same CHANGE TEAM statement. When a CHANGE TEAM statement is executed, there is an implicit synchronization of all active images of the team containing the executing image that is identified by *team-variable*. On each active image of the team, execution of the segment following the statement is delayed until all other active images of the team have executed the same statement the same number of times since execution last began in the team that was current before execution of the CHANGE TEAM statement. When a CHANGE TEAM construct completes execution, there is an implicit synchronization of all active images in the current team. On each active image of the team, execution of the segment following the END TEAM statement is delayed until all other active images of the team have executed the same construct the same number of times since execution last began in the team that was current before execution of the corresponding CHANGE TEAM

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

statement.

NOTE 5.1

Deallocation of an allocatable coarray that was not allocated at the beginning of a CHANGE TEAM construct, but is allocated at the end of execution of the construct, occurs even for allocatable coarrays with the SAVE attribute.

5.4 Image selectors

The syntax rule R624 *image-selector* in subclause 6.6 of ISO/IEC 1539-1:2010 is replaced by:

R624 *image-selector* **is** *lbracket cosubscript-list* ■
■ [, *team-identifier*] [, STAT = *stat-variable*] *rbracket*

R624a *team-identifier* **is** TEAM_NUMBER = *scalar-int-expr*
or TEAM = *team-variable*

C509 (R624) *stat-variable* shall not be a coindexed object.

If TEAM= appears in a coarray designator, *team-variable* shall be defined with a value that represents the current or an ancestor team. The coarray shall be established in that team or an ancestor of that team and the cosubscripts determine the image index in that team.

If TEAM_NUMBER = appears in a coarray designator and the current team is not the initial team, the *scalar-int-expr* shall be defined with the value of a team number for one of the teams that were formed by execution of the FORM TEAM statement for the current team. The coarray shall be established in an ancestor of the current team and the cosubscripts determine the image index in the team identified by TEAM_NUMBER. If TEAM_NUMBER = appears in a coarray designator and the current team is the initial team, the value of *scalar-int-expr* is ignored.

Execution of a statement containing an *image-selector* with a STAT= specifier causes the *stat-variable* to become defined. If the designator is part of an operand that is evaluated, and the object designated is on a failed image, the *stat-variable* is defined with the value STAT_FAILED_IMAGE in the intrinsic module ISO_FORTRAN_ENV; otherwise, it is defined with the value zero.

The denotation of a *stat-variable* in an *image-selector* shall not depend on the evaluation of any entity in the same statement. The value of an expression shall not depend on the value of any *stat-variable* that appears in the same statement. The value of a *stat-variable* in an *image-selector* shall not be affected by the execution of any part of the statement, other than by whether the image specified by the *image-selector* has failed.

NOTE 5.2

The image selector in `b[i]` identifies the current team. The image selector in `b[i,team_number=1]` identifies a sibling team. The image selector in `b[i,TEAM=ancestor]` identifies the team `ancestor`.

NOTE 5.3

The expression `x[i,stat=is(f(n))]`, where `is` is an array and `f(n)` is a function that returns an integer value is an example of the denotation of a *stat-variable* in an *image-selector* that depends on the evaluation of an entity in the same statement.

NOTE 5.4

The use of a STAT=specifier in an image selector allows a test to be made for a failed image in a reference where the use of a processor-dependent result could cause error termination or an incorrect execution path. Where there is no such possibility, it may be preferable to rely on the STAT=specifier in the next image control statement.

© ISO/IEC 2015 – All rights reserved

AVA Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015

NOTE 5.5

In the following code, the vector *a* of length N*P is distributed over P images. Each image has an array A(0:N+1) holding its own values of *a* and halo values from its two neighbors. The images are divided into two teams that execute independently but periodically exchange halo data. Before the data exchange, all images (of the initial team) must be synchronized and for the data exchange the coindices of the initial team are needed.

```
USE, INTRINSIC :: ISO_FORTRAN_ENV, ONLY: TEAM_TYPE
TYPE(TEAM_TYPE) :: INITIAL, BLOCK
REAL :: A(0:N+1)[*]
INTEGER :: ME, P2
INITIAL = GET_TEAM()
ME = THIS_IMAGE()
P2 = NUM_IMAGES()/2
FORM TEAM(1+(ME-1)/P2,BLOCK)
CHANGE TEAM(BLOCK,B[*]=>A)
   DO
      ! Iterate within team
        :
      ! Halo exchange across team boundary
      SYNC TEAM(INITIAL)
      IF(ME==P2  ) B(N+1) = A(1)[ME+1,TEAM=INITIAL]
      IF(ME==P2+1)   B(0) = A(N)[ME-1,TEAM=INITIAL]
      SYNC TEAM(INITIAL)
   END DO
END TEAM
```

5.5 FORM TEAM statement

R506 *form-team-stmt* **is** FORM TEAM (*team-number*, *team-variable* ■
■ [, *form-team-spec-list*])

R507 *team-number* **is** *scalar-int-expr*

R508 *form-team-spec* **is** NEW_INDEX = *scalar-int-expr*
or *sync-stat*

C510 (R506) No specifier shall appear more than once in a *form-team-spec-list*.

The FORM TEAM statement creates one or more teams from the active images of the current team. The value of *team-number* shall be positive and identifies the new team to which the executing image will belong. The team number of each new team within the current team is the *team-number* value of the FORM TEAM statements executed by its images. Successful execution of a FORM TEAM statement assigns the *team-variable* on each participating image a value that specifies the new team that the image will belong to.

The value of the *scalar-int-expr* in a NEW_INDEX= specifier specifies the image index that the executing image will have in the team specified by *team-number*. It shall be positive and less than or equal to the number of images in the team. Each image with the same value for *team-number* shall have a different value for the NEW_INDEX= specifier. If the NEW_INDEX= specifier does not appear, the image index that the executing image will have in the team specified by *team-number* is a processor-dependent value that shall be positive, unique within the team, and not greater than the number of images in the team.

The FORM TEAM statement is an image control statement. If the FORM TEAM statement is executed on one image, the same statement shall be executed on all active images of the current team. When a FORM TEAM statement is executed, there is an implicit synchronization of all active images in the current team. On these images, execution of the segment following the statement is delayed until all other active images in the current

© ISO/IEC 2015 – All rights reserved
Documents
Document Sharing Hub
Full document details: PD ISO/IEC TS 18508:2015