4.1.2 Combined Directive

Synopsis

Multiple attributes can be specified by one combined declarative directive, which is analogous to type declaration statements using the "::" punctuation.

Syntax

[F]	`!$xmp` *combined-directive*	**is**	*combined-attribute [, combined-attribute]...* `::` *combined-decl [, combined-decl]...*
[C]	`#pragma xmp` *combined-directive*	**is**	*combined-attribute [, combined-attribute]...* `::` *combined-decl [, combined-decl]...*

combined-attribute is one of:

`nodes`
`template`
`distribute (`*dist-format [, dist-format]...* `) onto` *nodes-name*
`align (` *align-source [, align-source]...* `) ` ▮
▮ `with` *template-name* `(`*align-subscript [, align-subscript]...* `)`
`shadow (` *shadow-width [, shadow-width]...* `)`
[F] `dimension (` *explicit-shape-spec [, explicit-shape-spec]...* `)`

and *combined-decl* is one of:

nodes-decl
template-decl
array-name

Description

A combined directive is interpreted as if an object corresponding to each *combined-decl* is declared in a directive corresponding to each *combined-attribute*, where all restrictions of each directive, in addition to the following ones, are applied.

Restrictions

- The same kind of *combined-attribute* must not appear more than once in a given *combined-directive*.
- If the `nodes` attribute appears in a *combined-directive*, each *combined-decl* must be a *nodes-decl*.
- If the `template` or `distribute` attribute appears in a *combined-directive*, each *combined-decl* must be a *template-decl*.
- If the `align` or `shadow` attribute appears in a *combined-directive*, each *combined-decl* must be an *array-name*.
- `[F]` If the `dimension` attribute appears in a *combined-directive*, any object to which it applies must be declared using either the `template` or the `nodes` attribute.