Implicit Data Region
There is an implicit data region in each subprogram that starts before the first executable statement and ends after the last executable statement.

Declaraive Data Directive
A declarative data directive is used to specify that data is to be allocated in device memory for the duration of the implicit data region of the subprogram.

Fortran

C

procedure.

A device present directive finds an existing device copy of

Device Present Directive
A device present directive finds an existing device copy of data allocated in a data region surrounding the call to this

A device present directive finds an existing device copy of data allocated in a data region surrounding the call to this

Fortran

C

!$acc data region [clause [ [ ] clauses] ...] new-line

structured block

Fortran

C

Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.

PGI Accelerator Directive Syntax

Only one directive-name per directive statement.

Clause order is not significant and may be repeated unless otherwise specified. Where applicable, clause list arguments are comma-separated variable names, array names, or subarrays with subscript ranges.
**Wait Directive**

A wait directive causes the program to wait until all asynchronous activities associated with the given handle are complete, or until all asynchronous activities are complete if no handle is specified.

```
C
#pragma acc wait ([handle])
```

**Fortran**

```fortran
!$acc wait ([/handle])
```

---

**DataClauses**

The description applies to the clauses used on compute regions, data regions, and to the standalone declarative data directives and executable update directives.

- `copy (list)`
  - Copies the data in list on the device and copies the data from the host to the device when entering the region and copies the data from the device to the host when exiting the region.

- `copyin (list)`
  - Allocates the data in list on the device and copies the data from the host to the device when entering the region.

- `copyout (list)`
  - Allocates the data in list on the device and copies the data from the device to the host when exiting the region.

- `local (list)`
  - Allocates the data in list on the device, but does not automatically copy the data between the host and device.

- `device() (list)`
  - Only the list entries must be pointer variables that contain device addresses, such as from CUDAmalloc.

- `deviceptr() (list)`
  - Declares that the data in list are to be allocated only on the device, and are accessible only within compute regions.

- `update_device() (list)`
  - Copies the data from the host to the already-allocated space on the device for the data in list, when entering a nested data region, compute region, or at the update directive.

- `update_host() (list)`
  - Copies the data from the already-allocated space on the device to the host for the data in list, when exiting a nested data region, compute region, or at the update directive.

- `mirror() (list)`
  - Fortran-only. Specifies that the data in list should mirror the allocation on the host; does not automatically copy data.

- `redirected() (list)`
  - Data in list must be dummy arguments. Specifies that data in list must already be allocated on the device by the caller.

- `device resident() (list)`
  - Specifies which device number to use. Override in the program with a call to acc_get_device.

- `acc_set_device_num(devicenum, devicetype)`
  - Sets the device type to use to execute compute regions; this assumes that the code has been compiled with PGI Unified Binary. The valid device types are acc_device_host or acc_device_nvidia.

- `acc_get_num_devices(devicetype)`
  - Returns the number of devices of the specified type; the only valid device type is acc_device_nvidia.

- `acc_set_device(devicetype)`
  - Sets the device type to use to execute compute regions; this is used in accelerator compute regions; the code has been compiled with PGI Unified Binary. The valid device types are acc_device_nvidia or acc_device_host.

- `acc_get_device()`
  - Returns the device type that will be used to execute the next accelerator region.

- `acc_set_device_num(devicetype)`
  - Sets the device number of the given type to use to execute accelerator regions.

- `acc_get_num_devices(devicetype)`
  - Returns the device number of the given type that will be used to execute the next accelerator compute region.

- `acc_async_wait(handle)`
  - Returns zero or .FALSE. otherwise.

- `acc_wait(handle)`
  - Waits until all asynchronous activities associated with the given handle have been completed; returns zero or .TRUE. otherwise.

- `acc_async_wait(handle)`
  - Waits until all asynchronous activities associated with the given handle have been completed.

- `acc_launch()`
  - Initializes the runtime library for that device type; the default is to initialize when the first accelerator region is entered.

- `acc_shutdown()`
  - Disconnects the program from the accelerator device; the next accelerator region will have to reinitialize the device.

- `acc_on_device(devicetype)`
  - This is used in accelerator compute regions to take different execution paths depending on whether the program is running on an accelerator or on the host. The argument must be a compile-time constant.

---

**Environment Variables**

```
ACC_DEVICE device
ACC_DEVICE NUM num
ACC_NOTIFICATION num
ACC_DEVICE_NUM num
```