

# Naming convention and parental relationships of the devices

[Naming convention](#)

[Parental relationships](#)

[Example](#)

## Naming convention

In addition to the [standard naming rules](#) that apply to all D2000 objects, the following rules apply to *Device Definitions* and *Devices*:

- The default prefix for *Device Definitions* is "DD.", for *Devices* it is "DI."
- The suffix is not allowed for *Device Definitions* or *Devices* - it must be empty.
- The prefix must end with a separating character - a *Separator* (by default a dot, it can be changed to an underscore).
- Both *Device Definitions* and *Devices* must have a unique prefix - it must be different from the prefixes of other object types (to avoid a possible naming collision).
- The *Separator* must be the same for both *Device Definitions* and *Devices*.
- It is possible to change the *Separator* in the menu setting, but it is necessary to change it for both *Device Definitions* and *Devices* at the same time (eg "DD." "DD\_" and "DI." "DI\_").
- Changing the Prefix for *Device Definitions* and *Devices*, as opposed to changing the Prefix for other types of D2000 objects, will cause **immediate renaming** of all *Device Definitions* (including changing item names to exclude the new Separator, if it was changed), *Devices*, and connected *Internal Items*. Therefore, we **do not recommend performing this operation on production**, but only on the development server, ideally without running other D2000 processes (only D2000 Server and CNF). Subsequently, we recommend restarting the D2000 system. We recommend that you back up the configuration database before renaming.
- The *Item name* in the *Structured Device Definition* cannot contain a *Separator*. Thus, by default, it can contain an underscore, but not a period.
- *Row Name* of a Device with an *Array Device Definition* cannot contain a *Separator*. Thus, by default, it can contain an underscore, but not a period.

Names of *Devices*

- The top-level *Device* (which is not connected to another *Device*) has a name in the form of a `prefix+own_name` (e.g. DI.BlockA)
- The name of a *Device* that is connected to another *Device* is changed to a name in the form `device_name+separator+item_name` (e.g. DI.BlockA.Gener1)
- If the subdevice is connected to a *Device* with a *Structured Device Definition*, then `item_name` is the *Item name* from the *Device Definition*. If the subdevice is connected to a *Device* with an *Array Device Definition*, then `item_name` is the *Name* of the corresponding row of the *Device*. When the subdevice is disconnected, its name changes. The *Separator* is replaced with an alternative separator (by default, a dot is replaced by an underscore). See the [example](#).

Names of *Internal Items*

- If a D2000 object (eg I/O tag) is connected to a *Device* item whose *Value type* is *Internal Item*, then its name changes to `device_name+separator+item_name` (e.g. DI.Gener1.Power).
- Belonging to the *Device* therefore has a higher priority than the [standard naming rules](#) for a given type of object.

Recursivity

- If a sub-device has other sub-subdevices or internal items connected, their names are also changed according to the above rules. See the [example](#).

## Parental relationships

Parental relationships of *Device Definitions*

- The parent of all *Device Definition* objects is the DEVICEDDEF.d object

Parental relationships of *Devices*

- The parent of a *Device* object becomes the respective *Device Definition* object when it is created.
- The parent of a *Device* object changes when the *Device* is connected to another *Device*. The parent of the sub-device becomes the *Device* to which the sub-device is connected.
- If a sub-device is disconnected from the *Device*, its parent will again be changed to the relevant *Device Definition*.

# Example

Consider the following object configuration, which models the following tree: Power Plant Blocks Generators

- The DI.EMO device (a power plant) has the DI.EMO.Blocks subdevice (with the *Array Device Definition*) connected in the Blocks item.
- The DI.EMO.Blocks device (blocks) is an array of two devices and has the DI.EMO.Blocks.BlockA and DI.EMO.Blocks.BlockB subdevices (with the *Array Device Definition*) connected. The *Names* of the individual rows of the array are BlockA and BlockB.
- The DI.EMO.Blocks.BlockA device (a block) is an array of two devices and has DI.EMO.Blocks.BlockA.TG1 and DI.EMO.Blocks.BlockA.TG2 subdevices (with the *Structured Device Definition*) connected. The *Names* of the individual rows of the array are TG1 and TG2.
- The DI.EMO.Blocks.BlockB device (a block) is an array of two devices and has DI.EMO.Blocks.BlockB.TG3 and DI.EMO.Blocks.BlockA.TG4 subdevices (with the *Structured Device Definition*) connected. The *Names* of the individual rows of the array are TG3 and TG4.
- The DI.EMO.Blocks.BlockA.TG1, DI.EMO.Blocks.BlockA.TG2, DI.EMO.Blocks.BlockB.TG3 and DI.EMO.Blocks.BlockA.TG4 devices (individual generators) have I/O tags connected to the Power and Running items.
- The following figure illustrates the described Devices together with their parents (for a top-level device DI.EMO, the parent is its *Device Definition*, for other devices the parent is the device to which they are connected):

Object name	Parent
[DI.EMO]	DD.PowerPlant
[DI.EMO.Blocks]	DI.EMO
[DI.EMO.Blocks.BlockA]	DI.EMO.Blocks
DI.EMO.Blocks.BlockA.TG1	DI.EMO.Blocks.BlockA
DI.EMO.Blocks.BlockA.TG2	DI.EMO.Blocks.BlockA
[DI.EMO.Blocks.BlockB]	DI.EMO.Blocks
DI.EMO.Blocks.BlockB.TG3	DI.EMO.Blocks.BlockB
DI.EMO.Blocks.BlockB.TG4	DI.EMO.Blocks.BlockB

- The following figure illustrates the names of the I/O tags connected to the Power and Running items of the individual TG1..TG4 turbogenerators together with the membership in the device:

Object name	Member of device
DI.EMO.Blocks.BlockA.TG1.Power	DI.EMO.Blocks.BlockA.TG1
DI.EMO.Blocks.BlockA.TG1.Running	DI.EMO.Blocks.BlockA.TG1
DI.EMO.Blocks.BlockA.TG2.Power	DI.EMO.Blocks.BlockA.TG2
DI.EMO.Blocks.BlockA.TG2.Running	DI.EMO.Blocks.BlockA.TG2
DI.EMO.Blocks.BlockB.TG3.Power	DI.EMO.Blocks.BlockB.TG3
DI.EMO.Blocks.BlockB.TG3.Running	DI.EMO.Blocks.BlockB.TG3
DI.EMO.Blocks.BlockB.TG4.Power	DI.EMO.Blocks.BlockB.TG4
DI.EMO.Blocks.BlockB.TG4.Running	DI.EMO.Blocks.BlockB.TG4

- If the DI.EMO.Blocks.BlockA.TG2 subdevice was disconnected from the DI.EMO.Blocks.BlockA, its name would change to DI.EMO.Blocks.BlockA.TG1 (the dot separator would be replaced by an alternate underscore separator). It is so obvious that it is now a top-level device (there is only one separator in the prefix in the name). At the same time, the parent of the device would change to a *Device Definition* - see the last line of the following figure:

Object name	Parent
[DI.EMO]	DD.PowerPlant
[DI.EMO.Blocks]	DI.EMO
[DI.EMO.Blocks.BlockA]	DI.EMO.Blocks
DI.EMO.Blocks.BlockA.TG2	DI.EMO.Blocks.BlockA
[DI.EMO.Blocks.BlockB]	DI.EMO.Blocks
DI.EMO.Blocks.BlockB.TG3	DI.EMO.Blocks.BlockB
DI.EMO.Blocks.BlockB.TG4	DI.EMO.Blocks.BlockB
DI.EMO.Blocks.BlockA.TG1	DD.Generator

- Similarly, the names of the I/O tags connected to the Power and Running items of the DI.EMO\_Blocks\_BlockA\_TG1 would change - see the last two lines of the following figure:

Object name	Member of device
DI.EMO.Blocks.BlockA.TG2.Power	DI.EMO.Blocks.BlockA.TG2
DI.EMO.Blocks.BlockA.TG2.Running	DI.EMO.Blocks.BlockA.TG2
DI.EMO.Blocks.BlockB.TG3.Power	DI.EMO.Blocks.BlockB.TG3
DI.EMO.Blocks.BlockB.TG3.Running	DI.EMO.Blocks.BlockB.TG3
DI.EMO.Blocks.BlockB.TG4.Power	DI.EMO.Blocks.BlockB.TG4
DI.EMO.Blocks.BlockB.TG4.Running	DI.EMO.Blocks.BlockB.TG4
DI.EMO_Blocks_BlockA_TG1.Power	DI.EMO_Blocks_BlockA_TG1
DI.EMO_Blocks_BlockA_TG1.Running	DI.EMO_Blocks_BlockA_TG1

**i** Related pages:

[Device definition](#)

[Device](#)