

# Application modules

**Application module** (AM) is a group of properly configured [objects](#) of D2000 which together perform some task. This group of objects is covered just one object of *Application module* type.

Group of objects can be heterogeneous against the type of objects. Each object in the configuration of application can be (but may not be) the part of just one application module. The object of *Application module* type can not be a part of another application module.

## Application module member

Any [object](#), except the objects of *Application module*, *System variable* and *User* type, can be a member of application module. Objects belonging to AM may be divided into two types:

- **Private** – they are accessible only for the user who created them.
- **Public** – they are visible also for the objects not belonging to AM (visible is only their basic configuration), they make the interface between AM and its surrounding.

## Surrounding of the application module

The surrounding of the application module consists of all the objects of application which are not a part of the application module. They are the objects do not belonging to any AM or objects belonging to other AM.

## Configuration property (CP)

Configuration property is just one item of an object configuration (e.g. monitoring in user variable). CP is represented by the path to the item of object in XML representation of object.

The objects with following configuration properties may be parameterized:

All types of objects	Parent.
User variable	Technical units, transformation palette, value, value time, log changes, save start value, manual control.
Structured variable	Array length, defined start values.
Historical value	History depth.

## Parameter of application module (PAM)

Parameter is defined by its **name** which is unique within the module. The name of parameter must comply with the limitation defined for the name of object.

Definition of parameter contains:

- **Description** of parameter – description of parameter, its meaning.
- **Obligation** - defines whether the value of parameter must be entered. It concerns only the parameter of object type and value.
- **Type** of parameter – string, logical value, integral number, real number, absolute time, relative time, object, value.
- **Default value**
- **Limitations** depending on the type of parameter (maximum, minimum, permitted types of objects, maximal length).

PAM may be connected to one or more configuration properties of member objects. Based on parameter usage there can be defined dependences between parameters. In current version, the parameter dependency occurs only in the structured variable under the condition that the array length and one or more items of structure are parameterized at the same time. In this case the dependent parameters are generated according to size that was defined at import of module.

**Note 1:** Parameters of *Absolute time* and *Relative time* are not compatible with the user variable either with the item of structure variable of *Absolute/Relative time* type. When these items are parameterized the parameter of *String* type must be connected. The format for absolute time is "hh:mi:ss.mss dd-mm-yyyy" and "ddd hh:mi:ss.mss" for relative time.

**Note 2:** The value of parameter that is connected with the parents of some object cannot be changed.

## Application module dependencies

Some application module can need another application modules in application. This dependency is defined in [configuration dialog](#) of object of *Application module* type. Dependency on another module always relates to a combination of application module - major version.

The dependencies are checked at import of application module. The import of application module is conditioned by completion of dependencies.



**Related pages:**

[Application modules - configuration dialog box](#)  
[Consistency check](#)