Create an application module

Application module (AM) is created in several basic steps.

1. Create and configure an object of Application module type

Create a new object of *Application module* type with the name **TelZoznam** in D2000 CNF.

Object of *Application module* type is configured in a configuration window Application modules - configuration dialog box.

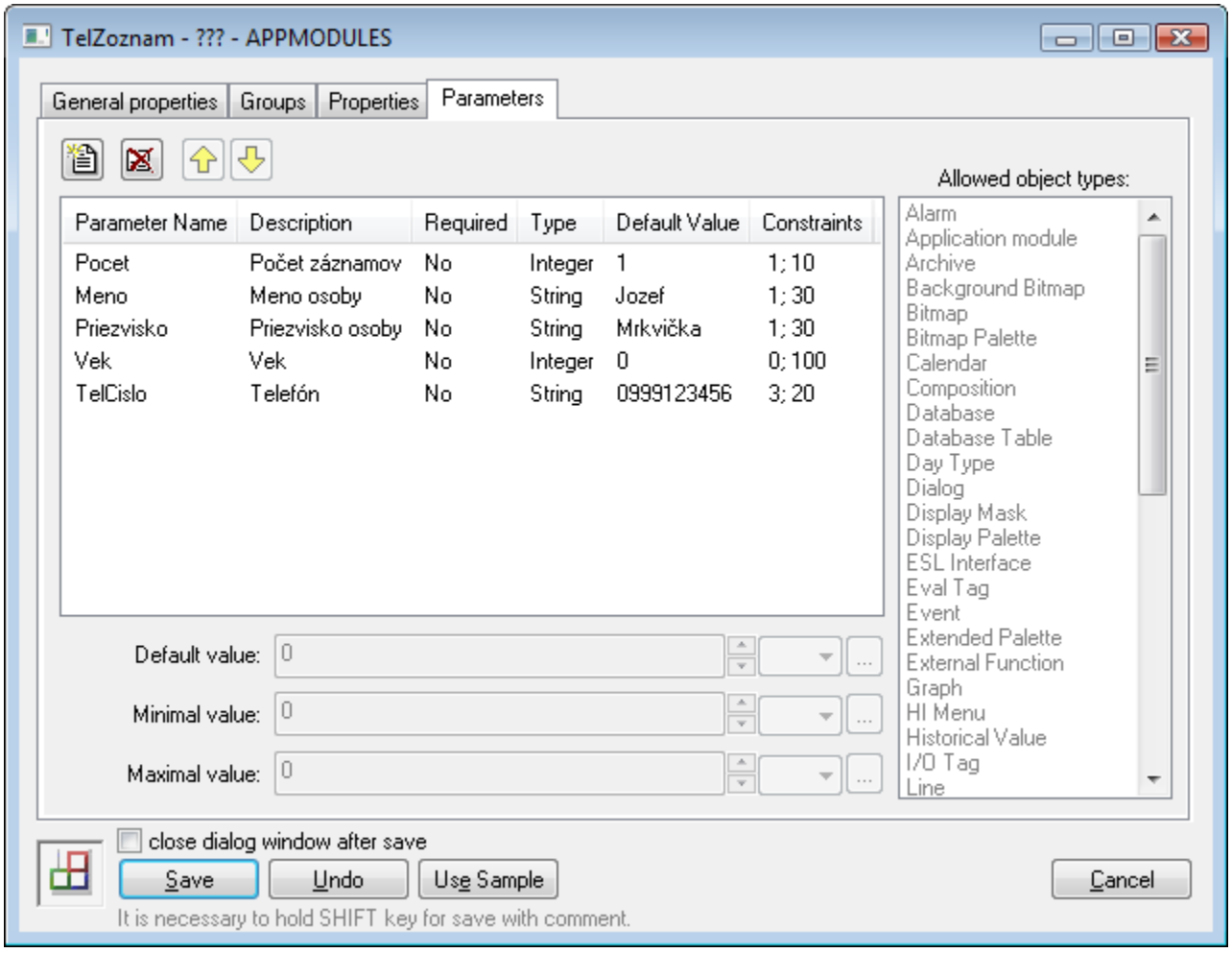
2. Definition of application module parameters

Tab **Parameters** in configuration dialog window enables to create, edit or remove any parameters of application module. The properties of parameters are described in detail in configuration of application module.

Task: Create the following parameters for module **TelZoznam**.

- *Pocet*: integer type, minimum 1, maximum 10
- *Meno*: string type, length from 1 up to 30 characters
- *Priezvisko*: string type, length from 1 up to 30 characters
- *Vek*: integer type, minimum 0, maximum 100
- *TelCislo*: string type, length from 3 up to 20 characters

If the description of parameter is entered it will show instead of parameter name at import of AM. In addition, you have to define a default value that will be preset to parameter at import of AM.



3. Assign the objects to application module

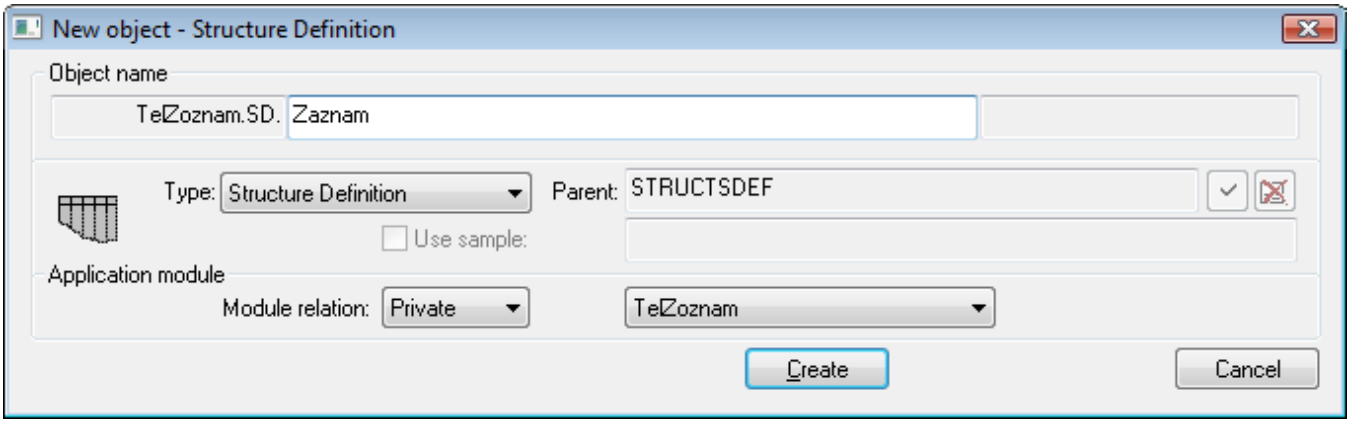
To assign the objects click on an item **Add into application module** in popup menu opened over the selected object in selection window of D2000 CNF.
The object may be assigned to application module as private or public.
The second way to integrate the objects into application module is at their creating. You can choose the application module and module relation in this dialog box.

If you want to remove an unnecessary object click on the item **Remove from application module** in popup menu.

Task: Create a private object of *Structure definition* type **- TelZoznam.SD.Zaznam**, with columns:

- Meno – text type
- Priezvisko – text type
- Vek – integer type
- TelCislo – text type

and private object of *Structured variable* type - **TelZoznam.SV.Zaznamy**, that uses the structure definition **TelZoznam.SD.Zaznam**.
To display the values of structured variable create the public object of picture type **TelZoznam.S.TelZoznam** containing a Browser displayer connected with **TelZoznam.SV.Zaznamy**.



4. Parameterize the configuration properties of member objects

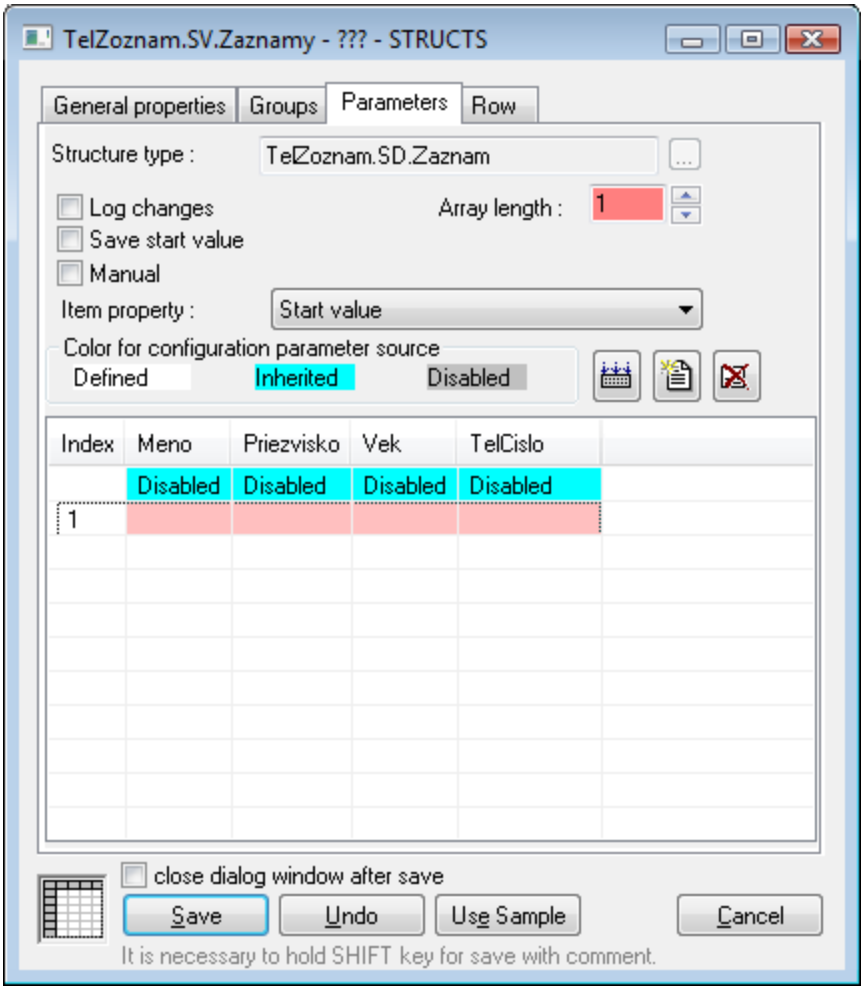
Configuration properties may be parameterized through the configuration dialog box of object.
If you want to know which configuration properties may be parameterized, right-click on any place in configuration dialog box of object and select the item **H ighlight all conf. properties** in popup menu.

If you right-click on configuration property (CP) you can choose some parameter of application module that will be assigned to it (the item **Assign AM parameter**).

Task: Parameterized CP *Array length* **TelZoznam.SV.Zaznamy** by parameter **Pocet**. Set the array length on 1 and define the values in the columns *Meno, Priezvisko, Vek* and *TelCislo* in the first row as an empty string.
Now, you may parameterize the values in the first row of structured variable.

In this way the special example of dependency among AM parameters is created. It enables to "duplicate" parameters depending on size of structured variable.

The item **Parametrize parent** in popup menu of selection window enables to parameterize the parents of several objects en bloc. Parameter that is assigned to some configuration property may be modified or removed from dialog box Configuration properties.



5. Export of application module

Application module can be exported from the popup menu opened over the object *Application module* type, the item Extended actions - **Export module**. Select a directory (e.g. C:\TelZoznam), where you want to export the application module and after that all the objects of application module will be exported. Before export the application module is checked, it prevents to export the inconsistent module.

Warning: Content of directory will be deleted before export!

6. Create an archive of application module

Utility **D2Archive** is intended for making the application module archive. The following conditions need to be fulfilled:

- Directory, to which the application object was exported, must contain an exported object of *Application module* type in its root. All others objects can be in any subdirectories.
- Name of archive (without suffix *.d2a*) must be the same as object name of *Application module* type. XML file (without suffix *.xml*) of this object must be the same, too.

The archive of application module without any licensing can be created by this command:

```
d2archive c ARCHIVE_NAME DIRECTORY_OF_MODULE
```

Note: Name of archive can be without suffix – suffix *.d2a* is added automatically.

In this task you can use the command:

```
d2archive c TelZoznam.d2a C:\TelZoznam
```

To unpack the archive use this command:

```
d2archive x ARCHIVE_NAME DIRECTORY_OF_MODULE
```

Import of application module

Import of AM is performed through the item **XML Import of application module** from menu Objects in D2000 CNF. Select the archive of application module and dialog of configuration of PAM values will open. The current configuration of parameters can be saved there.

After the configuration is finished click on button **Import** to activate the import.

Task: Import and configure the created module **TelZoznam**.
Select the file **TelZoznam.d2a**, set number of records and enter the values. After the import is successful, the picture **TelZoznam.S.TelZoznam** should display the content of structured variable.

	Meno	Priezvisko	Vek	TelCislo
1	Jozef	Mrkvička	27	0999123123
2	Peter	Novák	35	0999213213
3	Milan	Kováč	41	0999312312
4	Jozef	Mrkvička	0	0999123123
5	Jozef	Mrkvička	0	0999123123
6	Jozef	Mrkvička	0	0999123123
7	Jozef	Mrkvička	0	0999123123
8	Jozef	Mrkvička	0	0999123123

Example of application module

Following files contain the configuration of individual objects of module and complete example of application module archive **TelZoznam**.

1. Application module
2. Structure definition
3. Structured variable
4. Picture
5. Archive of application module

Related pages:

Application module