

Comparing objects

D2000 CfgSynchro - comparing objects

In order to synchronize configuration databases it is necessary for every object to correctly determine its image (counterpart) in the other database. For this purpose, so called unique identifier (UID - 64-character-long string), that is a part of each object's configuration and its value is generated automatically by the D2000 system when the object is created and it never changes. As for objects of [System](#) type, the UID is the same as the name of the object itself. This ensures that that if any additional system objects are added to the system in various configurations, they will be considered correct.

What stands above implies, if in two different configuration databases objects of the same name and type are created, then when synchronizing, these will be considered to be totally different objects, which will not permit the synchronization to take place, because of their UIDs being different (in certain cases it is possible to suppress such behaviour by synchronizing the UIDs on user's request - this is referred to as **Name conflict**).

Configuration database A			Configuration database B	
Name	UID		Name	UID
U.New	98.....			
			U.New	99.....
U.Old	D4.....	↔	U.Old	D4.....
U.Old1	F8.....	↔	U.Old2	F8.....

The picture shows two configuration databases that consist of objects of [User variable](#) type - *U.New*, *U.Old*, *U.Old1* and *U.Old2*. So as to prevent ambiguities it is necessary to not only refer to an object by its name but to include the name of the database the object belong to , as well, which will form the prefix of the original object's name (e.g. *A#U.New* denotes an object called *U.New* from the configuration database A).

In accordance to the definition, **UID** is generated for every object when it is created. Therefore we can claim objects *A#U.Old* and *B#U.Old2*, as well as objects *A#U.Old1* and *B#U.Old2* to be identical (created as copies). Objects *A#U.New* and *B#U.New* are different and due to which e.g. when copying object *A#U.New* to the configuration of B, a quite new object is created. In order to preserve changes within a configuration, in addition to being copied, the object will have to be renamed, too.

The first step of synchronizing databases is opening them. During this period, the necessary part of the configuration is automatically loaded. If there are objects without set UID, they will receive one automatically. Such a situation may occur if the database is opened by the process [D2000 CfgSynchro](#) for the first time (after upgrading the version of D2000 system). Subsequently, pairs of objects are formed based on matching UIDs and their [mutual state](#) is evaluated because of possible change of their configuration.