

About Table-to-Table Synchronization Rules

It is often the case in practical information systems that information is held in a denormalized format. For example, an employee name may be held in several tables. It is desirable (often essential) that if the name is masked in one column then the other tables in which the information is held should also be updated with an identical value. This requirement is called Table-To-Table Synchronization. There are two other types of synchronization ([Row-Internal](#) and [Table-Internal](#)) and all three have quite different purposes and functions.

Table-To-Table Synchronization rules are designed to correlate any data changes made to a table column with columns in other tables. Thus implementing a Table-To-Table Synchronization rule requires the knowledge of a join condition between the two tables. The rule will use this join condition to ensure the appropriate rows are updated correctly. Also required, is knowledge of the columns in each of the source and target tables which must be synchronized.

As an example, consider the two tables:

TABLE SOURCE
COLUMN IDNUM decimal
COLUMN NAME varchar(40)

TABLE TARGET
COLUMN IDNUM decimal
COLUMN NAME varchar(40)

Assume the NAME column must be synchronized between the two tables and the join condition is SOURCE.IDNUM=TARGET.IDNUM. If a [Substitution Rule](#) operates on the SOURCE.NAME column and a subsequent Table-To-Table Synchronization rule is applied, then each TARGET.NAME column will receive values identical to the ones in the SOURCE table where the join condition matches.

Important Note: It is important to realize that if there are IDNUM values in the TARGET table that are not present in the SOURCE table, the Table-To-Table Synchronization rule will have no effect on those rows. These columns will **not** be masked. In such cases a Substitution rule is usually applied to the columns in the target table before the synchronization takes place. This ensures that the sensitive data in the target table gets some form of masking even if the subsequent synchronization operation does not update it.

Table-To-Table Synchronization is probably the most common synchronization task and it is quite easy to achieve using the Data Masker software - just [implement](#) the specialized Table-To-Table Synchronization rule. Choose the table to use as the source and target. These two tables must be different - if they are the same then use a [Table-Internal Synchronization rule](#). Once the source and target tables have been chosen, fill in the columns to be synchronized and the join condition between the tables. The help file for the [New Table-To-Table Synchronization rule](#) form provides more details on the mechanics of the rule configuration process.

Table-To-Table Synchronization rules are created by launching the [New Table-To-Table Synchronization rule](#) form using the *New Rule* button located on the bottom of the [Rules in Set](#) tab.

SearchReplace	Enabled	Not Run		DM_CUSTOMER_NOTES	(E)
Sync. Manager	Enabled	Not Run		DM_INVOICE	Fanout: 2
Rule Group	n/a	n/a		Managed Rules	
Command	Enabled	Not Run		Cmd using Controller ...	ExitScriptOn
Command	Enabled	Not Run		Cmd using Controller ...	ExitScriptOn
Command	Enabled	Not Run		Cmd using Controller ...	ExitScriptOn
Command	Enabled	Not Run		Cmd using Controller ...	ExitScriptOn
Substitution	Enabled	Not Run		TMP_DMS_0014_888...	
FK Manager (Disable)	Enabled	Not Run		Controller Schema	Active: 1 of
Table-To-Table	Enabled	Not Run		DM_INVOICE	
Table-To-Table	Enabled	Not Run		DM_INVOICE_LINE	
FK Manager (Enable)	Enabled	Not Run		Controller Schema	Active: 1 of
Command	Enabled	Not Run		Cmd using Controller ...	ExitScriptOn
Command	Enabled	Not Run		Cmd using Controller ...	ExitScriptOn

Click on this Button to Create a New Masking Rule

File | Disable Rule | Clone Rule | Bulk Rule Gen. | New Rule

Adding a Table-To-Table Synchronization Rule