TARGIT Quality Check Task

TARGIT Quality Check is a custom component for SQL Server Integration Services. The component assists in generating scripts for ensuring column and referential integrity in a relational SQL Server data warehouse database. The component works with SQL Server databases only.

Main tasks solved

Maintains referential integrity in the relational database to ensure that applications accessing the data can do so without transactional data loss, e.g. when using inner joins.

Installation

The TARGIT Quality Check component is included in the "TARGIT Custom SSIS Tasks" installation package found in the TARGIT App & Gauge Store or the Download Center on the TARGIT Portal. After installation, a restart of SQL Server Data Tools and the SSIS project is required for the task(s) to appear in the SSIS toolbox.

Tabs

1. Values: Maintenance of central lists of data values for updating missing data values
2. Empty: Insertion of dummy data rows in empty tables
3. Tables: Set up columns to be updated as well as foreign/primary key relations between tables

Values

Centralized maintenance of a replace value repository. Changing current values of the repository will permeate the entire quality check script. Any number of value definitions can be configured, e.g. if detailed values are required for specific tables. A default value definition is always available on task creation.

Empty

The Empty functionality has limited relevance on later installations. Originally designed for early versions of SQL Server Analysis Services which did not work if fact tables were empty. Selecting a table in the list will insert a dummy row in the table if the table is empty.

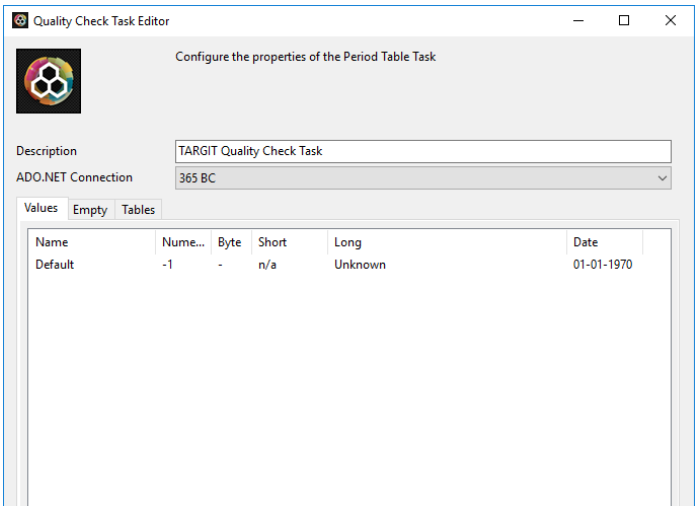
Tables

Used for selecting columns to be updated and for describing relations (foreign key->primary key) between tables. The primary advantage of using the task to describe table relations, especially in complex snowflake data model scenarios, is that the order in which referential updates are run in the script generated is automatically resolved. Therefore, the order in which relations between tables are added is not relevant. The task will handle the correct execution order for referential integrity row inserts.

Use cases

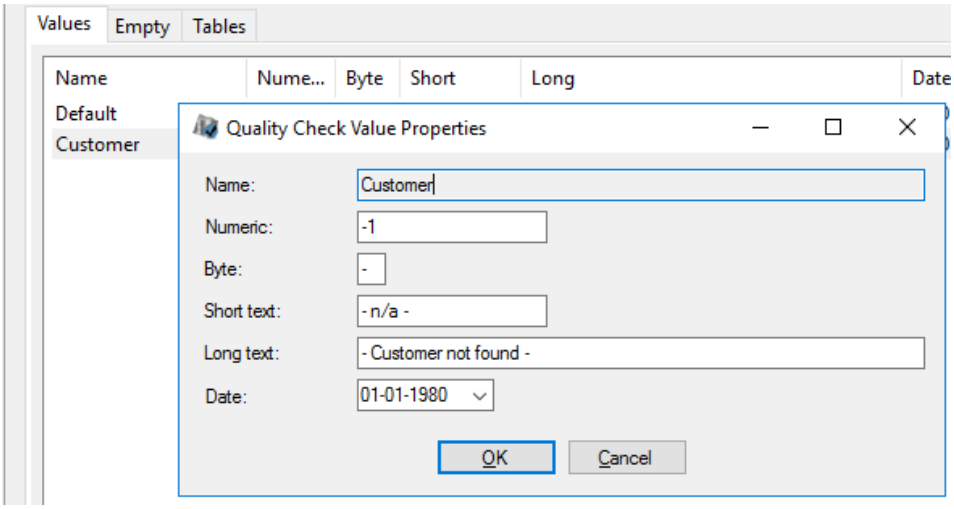
Getting started

- In the SSIS package, create an ADO.NET connection manager for the SQL Server database if not already available
- Drag the task from the SSIS Toolbox to the package Control Flow
- Double click the task and select the ADO.NET connection created. The "Invalid connection" message disappears.



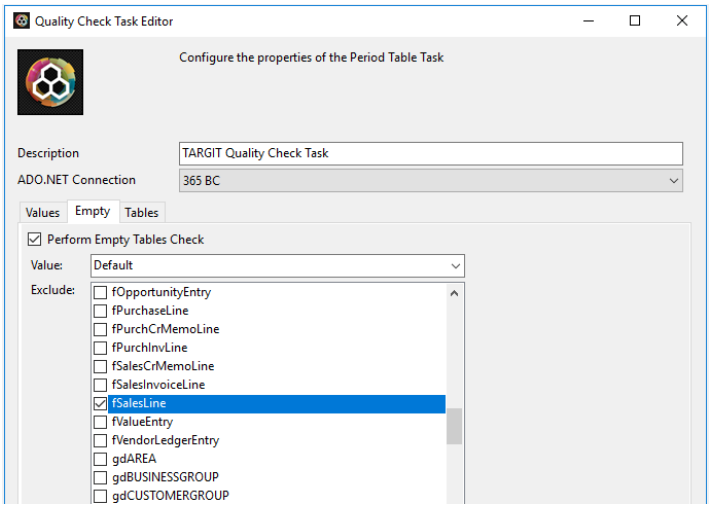
Setting up re-usable values

A default re-usable values item is available when the QC component is first added. Right click anywhere in the list box to add, duplicate, delete or change value items. Any value item has replacement values for data of main types **Numeric**, **Byte** (single character string), **Short** (string), **Long** (string) and **Date**. In the example screenshot, a set of values has been created for use in checks in customer related data.



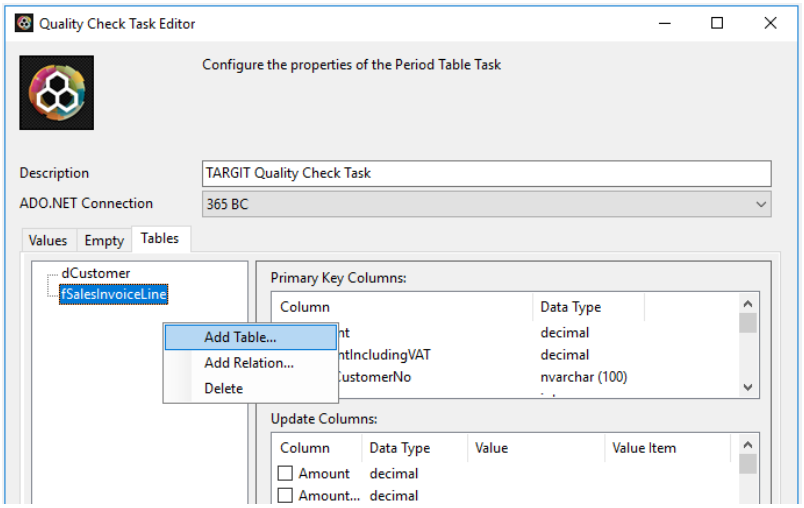
Appending dummy rows to empty tables

The use case for the empty tables check is probably rare. If the **Perform Empty Tables Check** is selected, the empty tables check uses a value item set to automatically insert a dummy row for all tables in the database. The value to insert in each column is automatically detected and applied. If a table is not to be checked for empty, it can be de-selected in the table list.

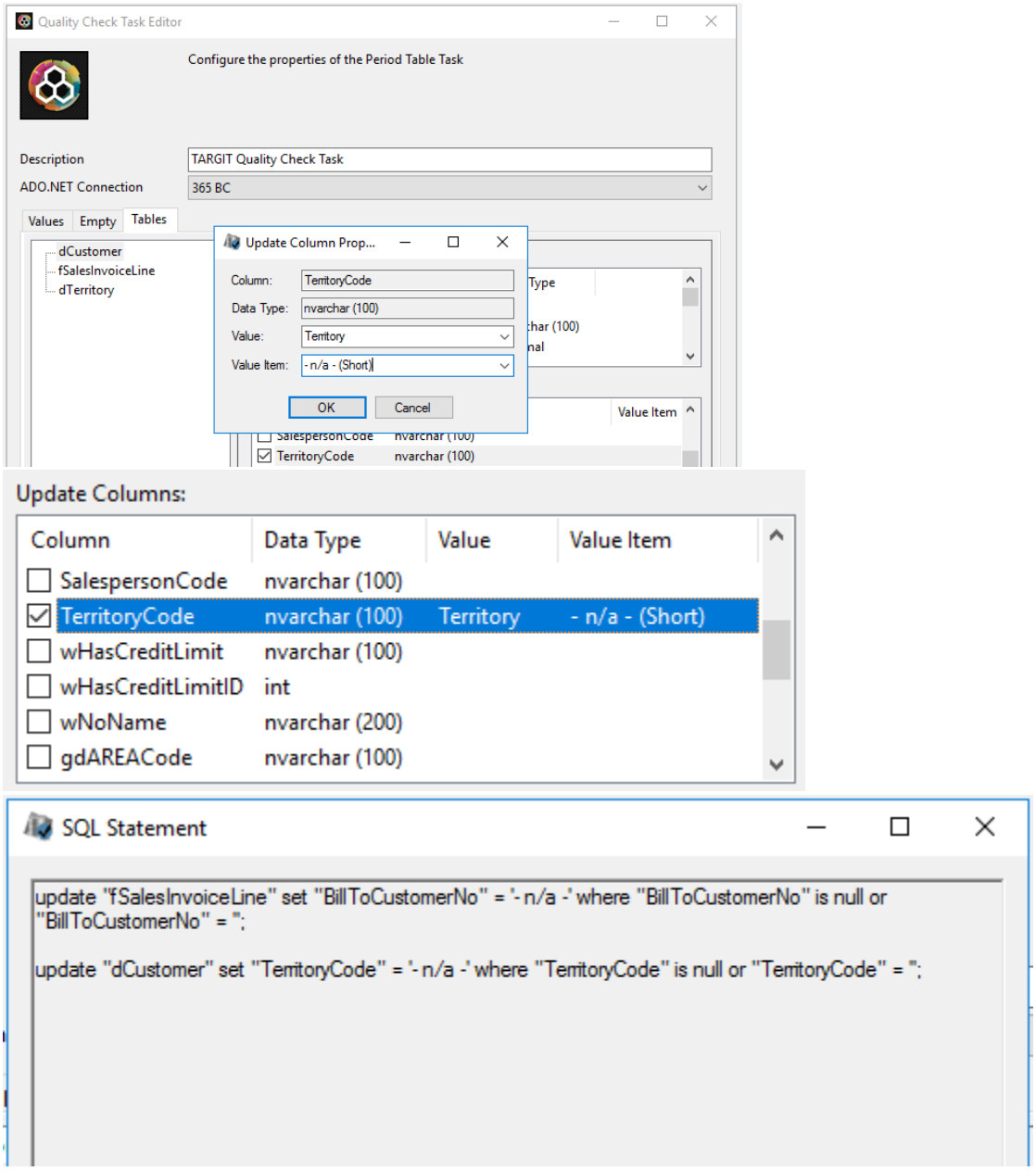


Adding a table for simple column checks

Consider Example 1. To update NULL or blank values in the data, add each table to the table tree by right clicking in the empty list on the left. No particular order is necessary.



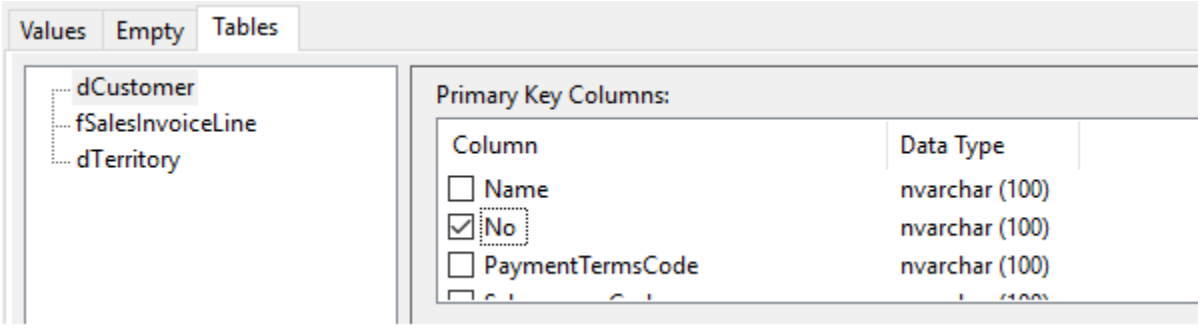
For each table, add the column to update in the Update Columns list. E.g. **TerritoryCode** for the customer table. As a column is selected in the list, the Update column properties window pops up. Select the value set and specific value to use for updating bad values in the column. At any time, the resulting update script that will be executed runtime is reviewable using the **Show SQL** button.



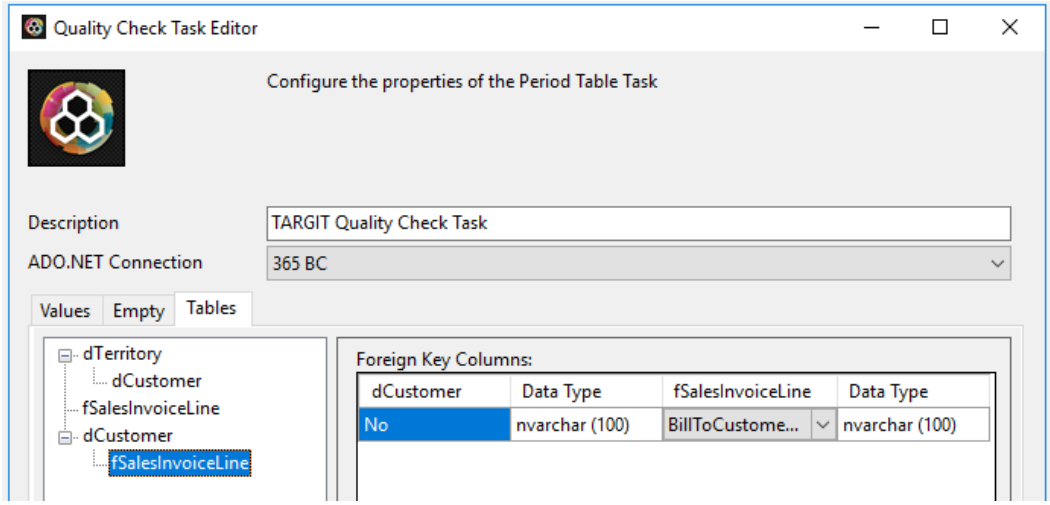
Adding tables for referential integrity checks

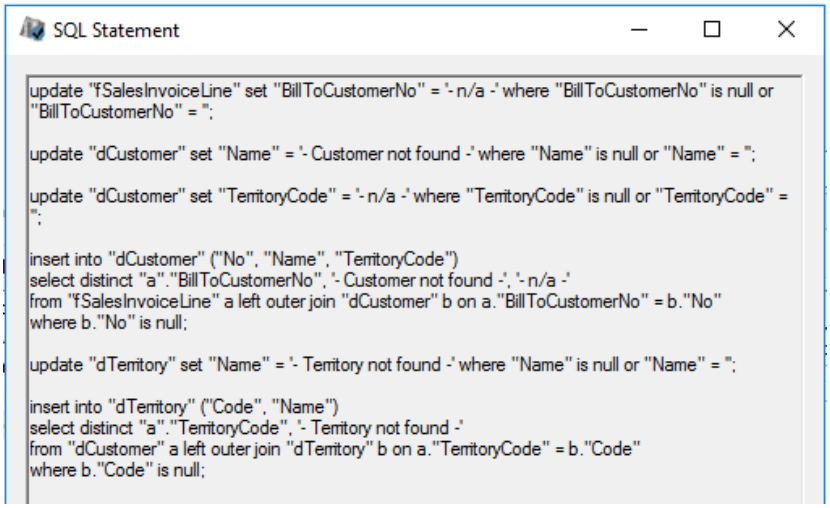
Consider Example 1. To establish full referential integrity, primary keys and relations between tables need to be defined. For each relation, dummy rows will be added to primary key (related) table to ensure that inner joins between the tables will return all transactions.

First, select the columns of the primary key of each snowflaked table using the Primary Key Columns list. Composite keys are supported.



Now, relations to the table can be defined by right clicking the table, then selecting **Add Relation**. Any number of tables can be added to check multiple tables against each table. Finally, select the column(s) of the foreign key relating to the table. Repeat for all relations in the table schema.



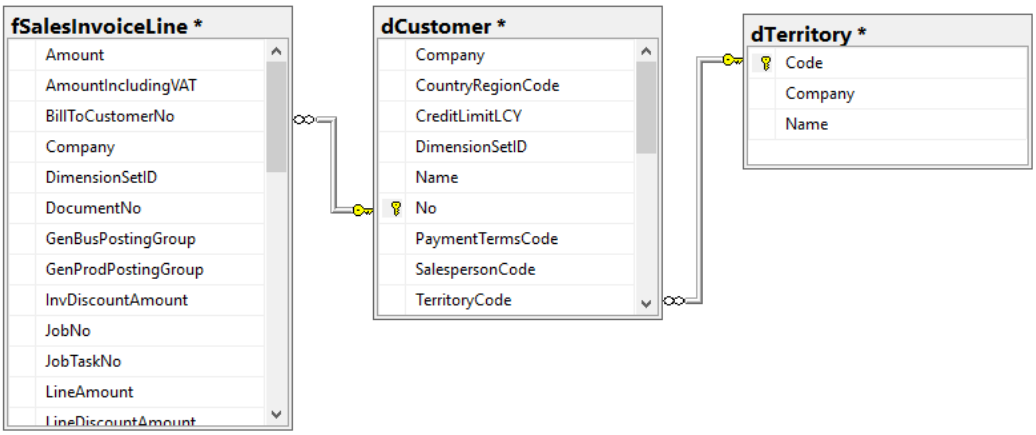


When checking the completed SQL statement, notice that the execution order is not the same as the order in which relations are added. The component will automatically resolve the order in which statements need to be executed for the insertion of rows to be made correctly.

Also, observe how the Update columns are re-used for insertion values in the dummy rows. This means that extra columns may need to be defined to get values into all columns of the added rows. Especially foreign key columns that may be used for referencing other tables of the schema.

Example 1 - Flawed relational data

- Snowflake schema. Sales Invoice table related to Customer table. Customer table related to Territory table.
- Data contains NULL values in foreign key columns, which breaks referential integrity and data consistency
- Full referential integrity needs to be established



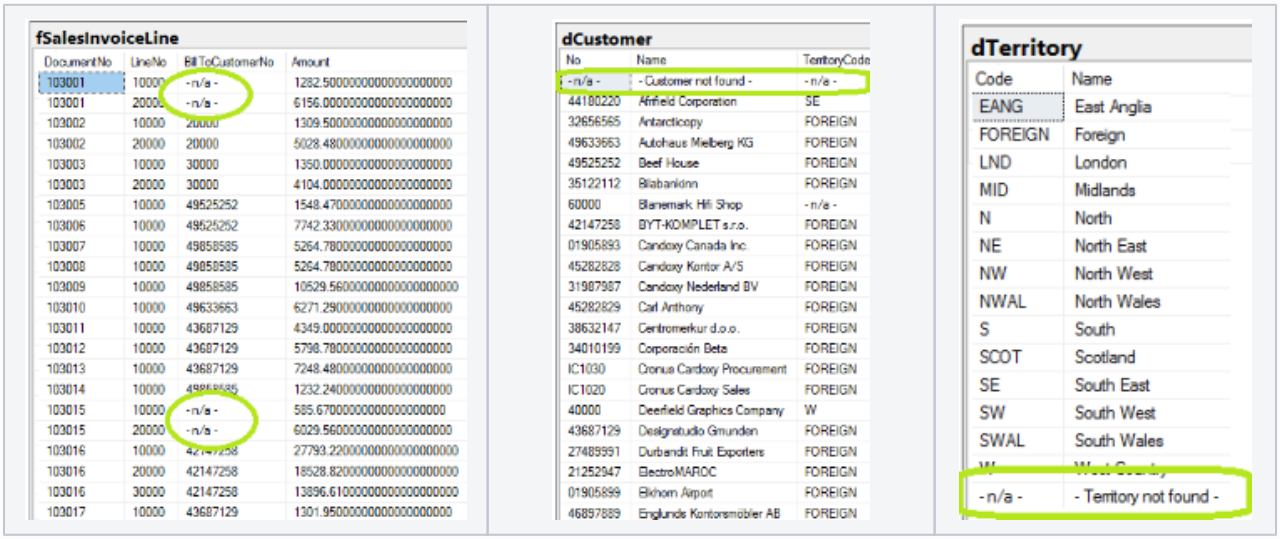
fSalesInvoiceLine

DocumentNo	LineNo	BillToCustomerNo	Amount
103001	10000	NULL	1282.500000000000000000000
103001	20000	NULL	6156.000000000000000000000
103002	10000	20000	1309.500000000000000000000
103002	20000	20000	5028.480000000000000000000
103003	10000	30000	1350.000000000000000000000
103003	20000	30000	4104.000000000000000000000
103005	10000	49525252	1548.470000000000000000000
103006	10000	49525252	7742.330000000000000000000
103007	10000	49858585	5264.780000000000000000000
103008	10000	49858585	5264.780000000000000000000
103009	10000	49858585	10529.560000000000000000000
103010	10000	49633663	6271.290000000000000000000
103011	10000	43687129	4349.000000000000000000000
103012	10000	43687129	5798.780000000000000000000
103013	10000	43687129	7248.480000000000000000000
103014	10000	49858585	1232.240000000000000000000
103015	10000	NULL	585.670000000000000000
103015	20000	NULL	6029.560000000000000000000
103016	10000	42147258	27793.220000000000000000000
103016	20000	42147258	18528.820000000000000000000
103016	30000	42147258	13896.610000000000000000000
103017	10000	43687129	1301.950000000000000000000

dCustomer

No	Name	TerritoryCode
44180220	Airfield Corporation	SE
32656565	Antarcticopy	FOREIGN
49633663	Autohaus Mielberg KG	FOREIGN
49525252	Beef House	FOREIGN
35122112	Blabankinn	FOREIGN
60000	Blanemark Hifi Shop	NULL
42147258	BYT-KOMPLET s.r.o.	FOREIGN
01905893	Candoxy Canada Inc.	FOREIGN
45282828	Candoxy Kontor A/S	FOREIGN
31987987	Candoxy Nederland BV	FOREIGN
45282829	Carl Anthony	FOREIGN
38632147	Centromerkur d.o.o.	FOREIGN
34010199	Corporación Beta	FOREIGN
IC1030	Cronus Cardoxy Procurement	FOREIGN
IC1020	Cronus Cardoxy Sales	FOREIGN
40000	Deerfield Graphics Company	W
43687129	Designstudio Gmunden	FOREIGN

Example 2 - Fixed relational data



fSalesInvoiceLine

DocumentNo	LineNo	BillToCustomerNo	Amount
103001	10000	- n/a -	1282.50000000000000000000
103001	20000	- n/a -	6156.00000000000000000000
103002	10000	20000	1309.50000000000000000000
103002	20000	20000	5028.48000000000000000000
103003	10000	30000	1350.00000000000000000000
103003	20000	30000	4104.00000000000000000000
103005	10000	49525252	1548.47000000000000000000
103006	10000	49525252	7742.33000000000000000000
103007	10000	49858585	5264.78000000000000000000
103008	10000	49858585	5264.78000000000000000000
103009	10000	49858585	10529.56000000000000000000
103010	10000	49633663	6271.29000000000000000000
103011	10000	43687129	4349.00000000000000000000
103012	10000	43687129	5798.78000000000000000000
103013	10000	43687129	7248.48000000000000000000
103014	10000	49858585	1232.24000000000000000000
103015	10000	- n/a -	585.67000000000000000000
103015	20000	- n/a -	6029.56000000000000000000
103016	10000	42147258	27793.22000000000000000000
103016	20000	42147258	18528.82000000000000000000
103016	30000	42147258	13896.61000000000000000000
103017	10000	43687129	1301.95000000000000000000

dCustomer

No	Name	TerritoryCode
- n/a -	- Customer not found -	- n/a -
44180220	Armeld Corporation	SE
32656565	Antarcticopy	FOREIGN
49633663	Autohaus Mielberg KG	FOREIGN
49525252	Beef House	FOREIGN
35122112	Blabankinn	FOREIGN
60000	Blanemark Hifi Shop	- n/a -
42147258	BYT-KOMPLET s.r.o.	FOREIGN
01905893	Candoxy Canada Inc.	FOREIGN
45282828	Candoxy Kontor A/S	FOREIGN
31987987	Candoxy Nederland BV	FOREIGN
45282829	Carl Anthony	FOREIGN
38632147	Centromerkur d.o.o.	FOREIGN
34010199	Corporación Beta	FOREIGN
IC1030	Cronus Candoxy Procurement	FOREIGN
IC1020	Cronus Candoxy Sales	FOREIGN
40000	Deerfield Graphics Company	W
43687129	Designstudio Gmunden	FOREIGN
27489991	Durbandit Fruit Exporters	FOREIGN
21252947	ElectroMARIOC	FOREIGN
01905899	Elkhorn Airport	FOREIGN
46897889	Englunds Kontorsmöbler AB	FOREIGN

dTerritory

Code	Name
EANG	East Anglia
FOREIGN	Foreign
LND	London
MID	Midlands
N	North
NE	North East
NW	North West
NWAL	North Wales
S	South
SCOT	Scotland
SE	South East
SW	South West
SWAL	South Wales
- n/a -	- Territory not found -

- All TARGIT custom components connect to the SQL Server database using an ADO.NET connection
- In later Data Tools (Visual Studio) versions, check the TargetServerVersion property for the SSIS project if the TARGIT components do not show up in the SSIS Toolbox
- Updating column data in large tables can be time consuming. Consider which columns are relevant for updating instead of just adding them all
- If the TARGIT components are not in the SSIS Toolbox after installation, a trouble shooting article is available (see Related articles)
- The TARGIT custom components are available for SQL Server versions beginning at 2005
- The components are free to use, but only officially supported for active TARGIT customers

Related articles

- TARGIT Custom Tasks - Problem solving
- TARGIT Period Table Task
- TARGIT Quality Check Task
- TARGIT Quality Check Transformation
- TARGIT Enumeration Task