



**REPLICATOR**  
**for SQL Server Databases**

[www.singleclicksolutions.co.uk](http://www.singleclicksolutions.co.uk)

Author: George Kounis  
gkounis@singleclicksolutions.co.uk

New electronic data capture requirements frequently involve certain system constraints. These constraints are either:

- platform constraints i.e. systems are designed to run on a certain database platform; and/or
- database schema constraints i.e. if the database is structured to accept data into certain fields, you cannot add fields or change what goes where without system modifications; and/or
- business rule constraints i.e. a system makes certain assumptions about user requirements which apply at the time the system was built. For example, if you are issuing a Sales Invoice, the system knows that it has to apply sales tax to the net total because this is what it has been designed to do. If, however, additional taxes are to be collected at the point of sale, the system designer would not have known at the time to make provision for this.

Requirements, however, do change and organisations are forced to either make system modifications or find ways to meet the new requirements while keeping the existing systems. Let us look at an example:

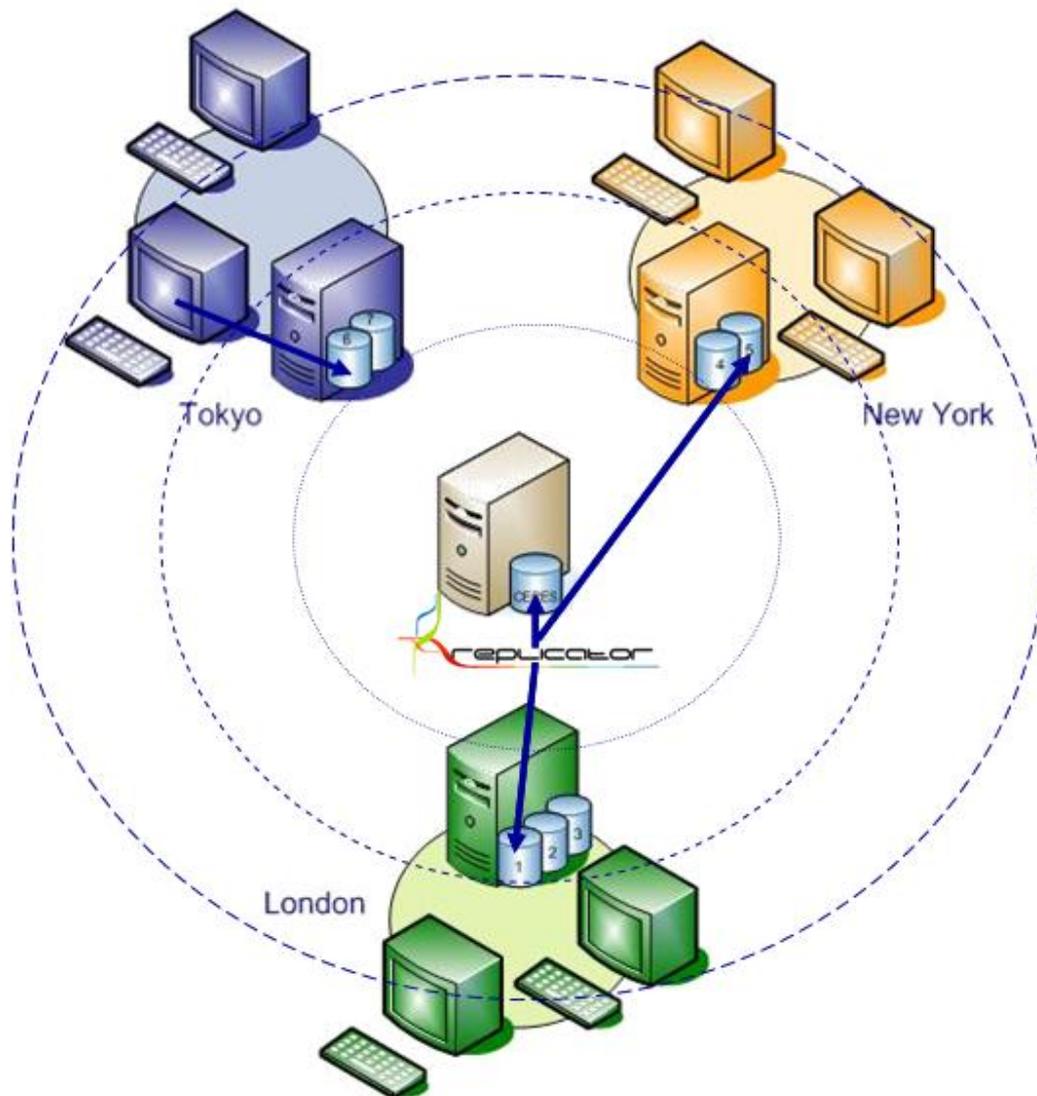
Your organisation uses an SQL Server based system and has decided to outsource their data capture to a shared service provider that uses an Oracle based system. Even if both parties happen to use an SQL Server platform, the database schema is unlikely to be the same and certainly the front-end system is not going to have the same business rules unless they both have the same system. Short of asking the Shared Service Provider to data capture directly into your system (which may defeat the economy of outsourcing) the ways available for data to be captured in one system and to be updated in another are:

1. Export the data from one and import into the other
2. Create triggers or use replication
3. Use a port listening system such as Microsoft's BizTalk Server

Depending on requirements each of these has both advantages and drawbacks. The one drawback they all have, however, is true system integration. You cannot, for example, search one system to determine if a record exists to avoid duplication and you cannot interact with a system using the information entered in another system. All the existing solutions wait for an event to be completed before they can do something with it. There can be no interaction, while the event is taking place.

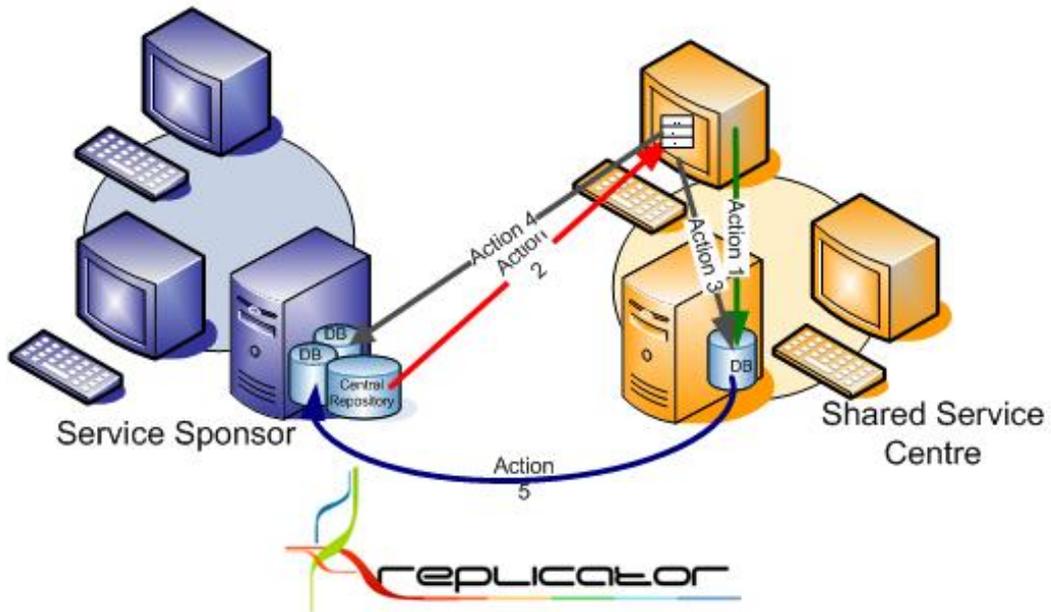
### **WhenDo Replication using the Replicator**

The recent launch of the Replicator from Single Click Solutions changes this landscape. Essentially, the Replicator is an SQL Server transaction log reader that operates as a windows service in the background and when certain operations are carried out on a SQL Server database (the WHEN part), certain actions are taken in response (the DO part).



**Figure 1 – WhenDo Replication**

Apart from the ability to take specified actions in one system in response to certain operations taking place in another system (WhenDo Replication) the REPLICATOR has the ability to trigger additional systems into action in real-time and use data from one to take action in another



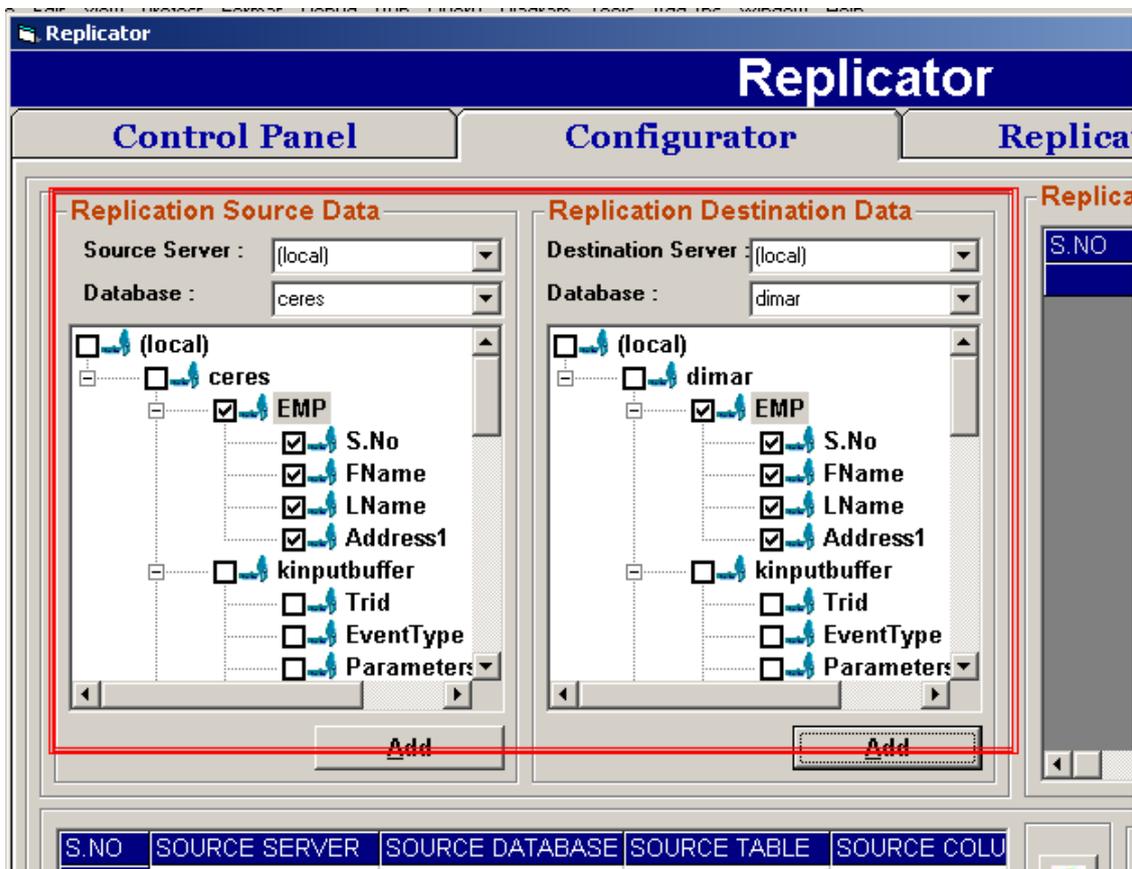
**Figure 2 – Complete System Integration**

Replicator- Configuration form:

**1: Configuring source and Destination details:**

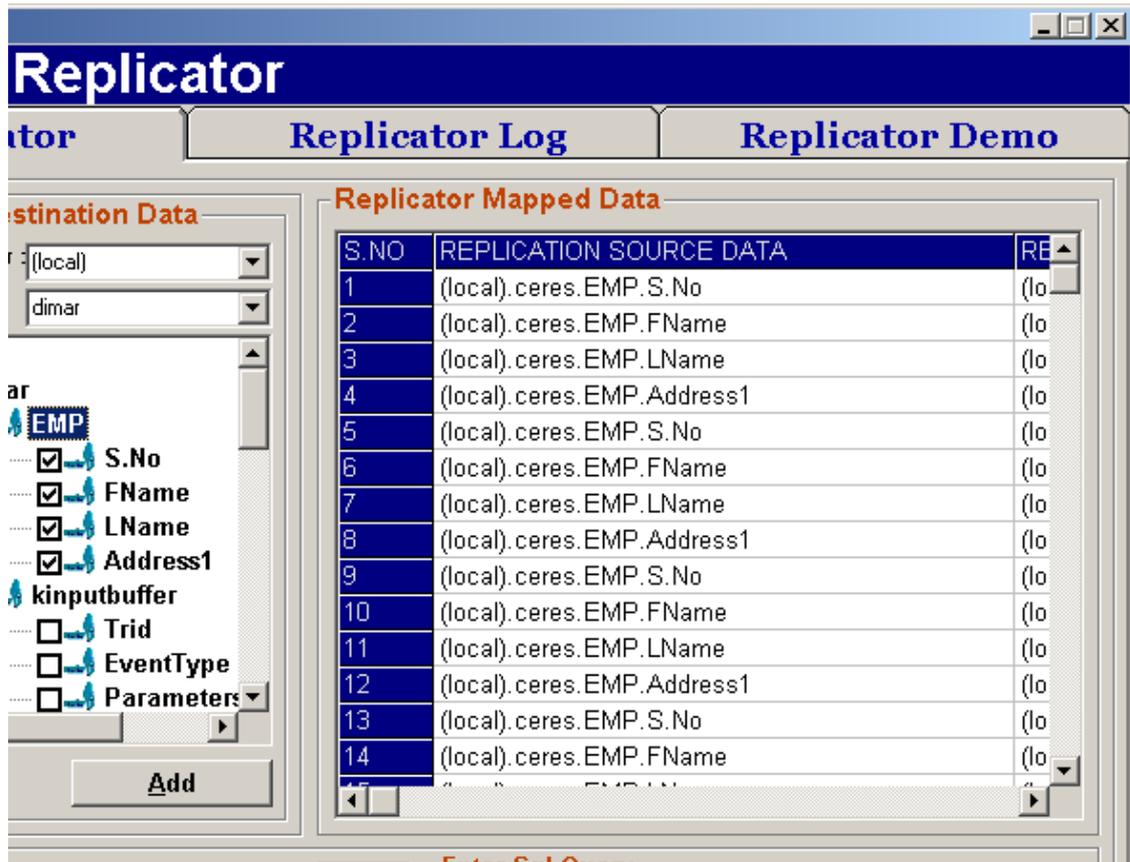
**Please select the source server and source destination, then you can see list of all tables available within that database in tree view. Now you please select the table name whichever you want to replicate data.**

Similarly, you please select destination server and destination database. And select the destination table from the other tree view. After that please click add button under source data and destination data to add the details into mapping grid.



In mapping grid, you can adjust source and destination details by using “Move Up” and “Move Down” buttons (buttons are in green color). In sql query text area you can add sql query corresponding to the mapping details.

After providing all the details you please click on “Save Mapping” details. After saving the details you can see the mapped details on right top grid “Replicator Mapped Details”.

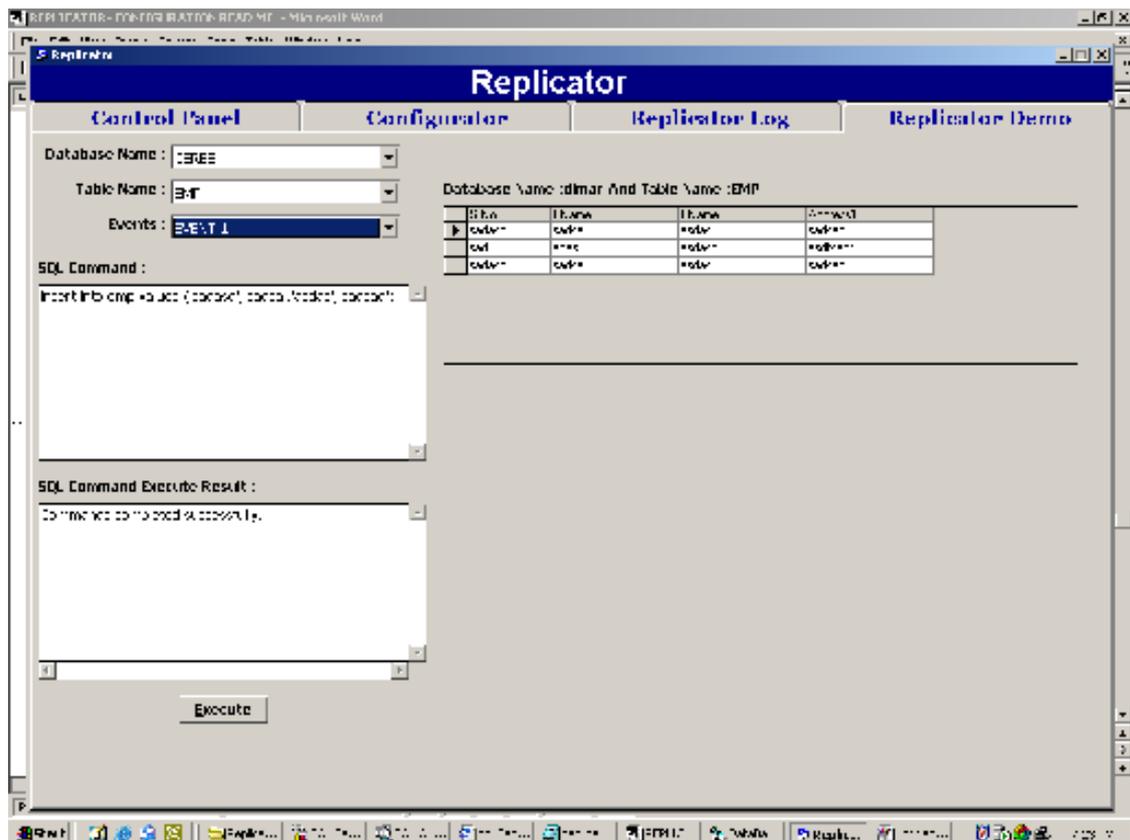


## Replicator Demo:

To apply changes to the Replicator destination tables, Please select on the database and Source table. Depending upon your table selection all the events corresponding to that particular table will be displayed in events combo.

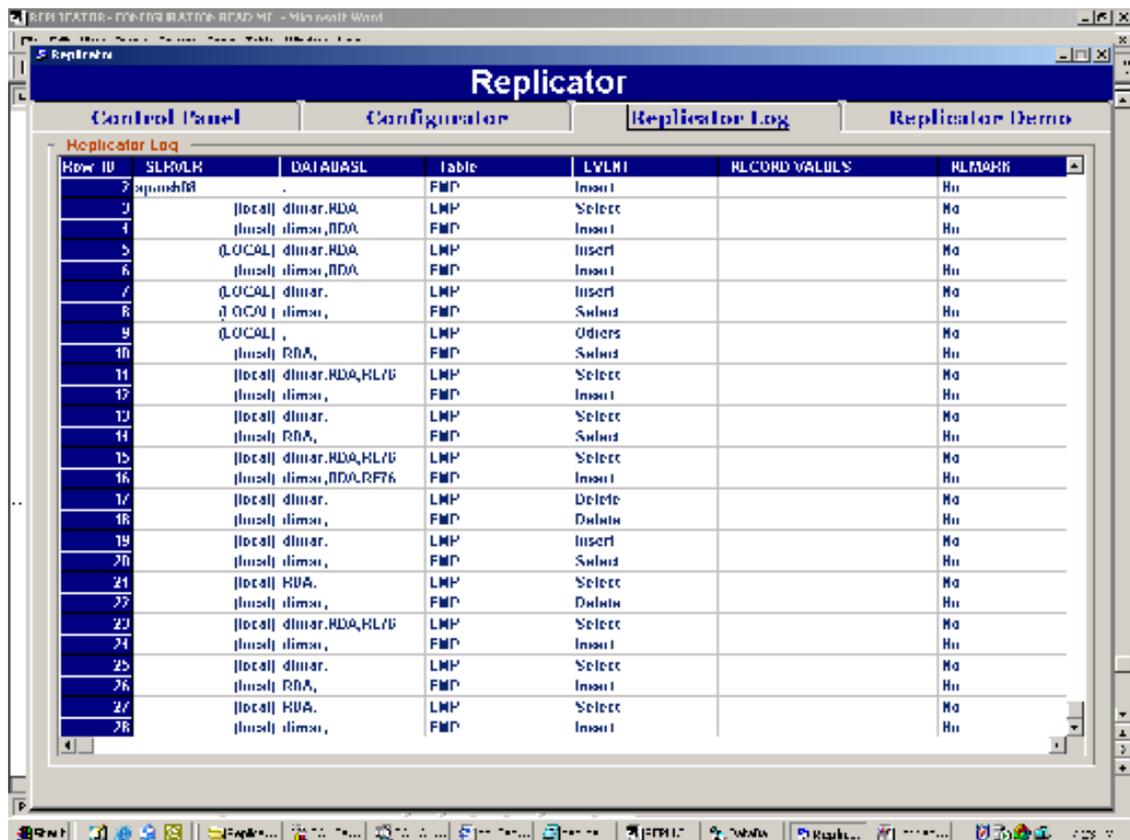
To execute a particular event please select the event from the combo, so that the data will be replicated to the corresponding destination tables.

Even you can enter any other sql query in sql command text area and can click execute button.



## Replicator Log:

Whenever an event occurs, the event details will be saved in Replicator Log.



Row ID	SERVER	DATABASE	Table	LEVEL	RECORD VALUES	REMARK
1	spunsh08	.	FMP	Insert		Ho
2	[local] dmar.RDA		LMP	Select		Ho
3	[msdt] dmsi.RDA		FMP	Insert		Ho
4	[LOCAL] dmar.RDA		LMP	Insert		Ho
5	[msdt] dmsi.RDA		FMP	Insert		Ho
6	[LOCAL] dmar.		LMP	Insert		Ho
7	[LOCAL] dmsi.		FMP	Select		Ho
8	[LOCAL] .		LMP	Others		Ho
9	[msdt] RDA,		FMP	Select		Ho
10	[local] dmar.RDA,HL/V		LMP	Select		Ho
11	[msdt] dmsi.		FMP	Insert		Ho
12	[local] dmar.		LMP	Select		Ho
13	[msdt] RDA,		FMP	Select		Ho
14	[local] dmar.RDA,HL/V		LMP	Select		Ho
15	[msdt] dmsi.RDA,RF76		FMP	Insert		Ho
16	[local] dmar.		LMP	Delete		Ho
17	[msdt] dmsi.		FMP	Delete		Ho
18	[local] dmar.		LMP	Insert		Ho
19	[msdt] dmsi.		FMP	Select		Ho
20	[local] RDA,		LMP	Select		Ho
21	[msdt] dmsi.		FMP	Delete		Ho
22	[local] dmar.RDA,HL/V		LMP	Select		Ho
23	[msdt] dmsi.		FMP	Insert		Ho
24	[local] dmar.		LMP	Select		Ho
25	[msdt] RDA,		FMP	Insert		Ho
26	[local] RDA,		LMP	Select		Ho
27	[msdt] dmsi.		FMP	Insert		Ho
28	[msdt] dmsi.		FMP	Insert		Ho