

Replication

Implementing Successfully Replication - Technical Paper

Replication

Visual Dolphin is designed and developed to support remote multi-branch operations and help sharing information between the Head Office and the institution's branches.

Visual Dolphin relies for that on the replication features of Microsoft SQL Server. Replication is simply the process of sharing data between databases in different locations. That is exactly what Visual Dolphin does: it replicates data, like the customer, supplier, inventory and financial information, to make the data available to all the institution locations or entities. In other words, it means that information is all accurate and up-to-date in every location.

The following lines provide technical details on the implementation of replication in Visual Dolphin.

How to Setup Data Replication

- 1- [Backup the Publisher Database](#) or the database that we need to publish
- 2- [Install latest service pack](#) released by Microsoft and found at the following Link: www.microsoft.com/downloads
- 3- [Create a new Windows user to be used](#) later on when running the Microsoft SQL engine and SQL Server Agent. In this paper, we will define this user as REPLUSER.
- 4- [Create a new Microsoft SQL Server user](#) to be used later on when setting the replication. In this paper, we will define this user as REPLUSER.
- 5- [Register Remote Server](#)
- 6- [Checking necessary condition](#) to run the Replication
- 7- [Steps to follow when setting a publication](#)
- 8- [Steps to follow when setting a subscription](#)
- 9- [Step by step Example on how to setup a merge replication](#)
- 10- [How to Backup replicated databases](#) at publisher and subscriber
- 11- [How to install a new Visual Dolphin Release](#) on a replicated database
- 12- List of Visual Dolphin tables to replicate, with the filter condition in case of company replication

General Concepts

- a. **Replication** is the process whereby data is copied between databases on the same server or different servers connected by LANs, WANs, or the Internet.
- b. Microsoft SQL Server replication uses a **publisher**, **distributor** and **subscriber** metaphor.
- c. **Publisher** is the server or database that sends its data to another server or database.
- d. **Subscriber** is the server or database that receives data from another server or database.
- e. **Distributor** is the server that manages the flow of data through the replication system. This server contains the **distribution database**.
- f. **Publisher** contains **publication/publications**. **Publication** is a collection of one or more articles that is sent to a subscriber server or database.
- g. **Article** is the basic unit of replication and can be a table or a subset of a table.
- h. **Subscription** is the group of data that a server or database will receive.
- i. There are **push** and **pull subscriptions**:
 - **Push subscription** is when the publishing server periodically pushes transactions out to the subscribing server or database.
 - **Pull subscription** is when the subscribing server periodically connects to the **distribution database** and pulls information.
- j. The **distribution database** is a system database, which is stored on the Distributor and does not contain any user tables. This database is used to store snapshot jobs and all transactions waiting to be distributed to Subscribers.

Backup The Publisher Database or the Database that need to publish

After creating a new publication, SQL Server will automatically add inside each replicated table a new column named ROWGUID, so if any problem occurs during the replication setup and in case we need to recover this setup, we will need to restore the publisher database and restart again our replication setup.

Install last Service Pack

It is recommended to upgrade Microsoft SQL Server with the latest service pack in order to bypass any problem related to service pack or any bug found in MS SQL Server.

Create a Windows and SQL user to be used while setting the replication

It is recommended to create a new Windows User and SQL User to be used for the replication setup. If we use the 'SA' or 'administrator' accounts, we probably may need to change their passwords frequently.

In that case, passwords have to be changed at all the levels of the replication, like:

- Defining the Publisher
- Defining the Subscriber
- Subscription
- Etc...

The Localsystem account has no access to shares on the network, since it is not an authenticated network account. So, if we want to set up replication, we must change the domain account, with Windows NT/Windows 2000 administrator's rights, in Microsoft SQL Server and for SQL Server Agent services.

If Microsoft SQL Server runs on Windows NT or Windows 2000, we can create Windows NT/Windows 2000 account and include it within the local Administrators group, into Domain Users group, and set *Log in as service* permission for this account.

Register the Remote MS SQL Server

Registering the remote server is fundamental since without a registration, we cannot create any subscription with the publisher.

To be able to register a remote server, we must have a LAN, WAN or Internet connection with this server.

- Connecting to a LAN is simple: the server exists on the institution's Local Area Network. We can register directly from inside the MS SQL Enterprise manager, by just right clicking on the root and choosing the option "Register New Server" and then following the Setup wizard until we reach a successful server registration.
- The WAN and Internet connections are a little bit different since additional steps should be done prior to the registration on the server. Three cases must be taken into consideration while registering a remote server:

Case 1: Registering the subscriber server on the publisher server. It means that the publisher can see the subscriber but the subscriber cannot see the publisher.

Case 2: Registering the publisher server on the subscriber server. It means that the subscriber can see the publisher but publisher cannot see the subscriber.

Case 3: Both server publishers and subscribers can see each others.

In all cases, the registered server must have a known IP address. The registration must take place from the Client Network Utility by adding a new Alias, defining the network libraries as TCP/IP and then entering the IP address in the connection parameter under the server name.

It is important not to forget filling the Server alias with the same remote server name and removing the server IP address. Otherwise, problems will occur during the setup of the replication.

Checking necessary conditions before setting a merge replication

- The Local-System account has no access to shares on the network, since it isn't an authenticated network account. If we want to set up replication, we must change the account Microsoft SQL Server and SQL Server Agent services to an account with Windows NT/Windows 2000 administrator's rights.

If Microsoft SQL Server runs on Windows NT or Windows 2000, we can create a Windows NT/Windows 2000 account and include it within the local Administrators

group, into the Domain Users group, and set *Log in* as service permission for this account.

Windows 9x does not support Windows NT services. If Microsoft SQL Server runs on Windows 9x, it is not necessary to create an SQL account.

- Only members of the SYSADMIN server role can set up and configure replication. Without these rights, replication cannot be configured.
- SQL Server Agent service (and of course, MS SQL Server service) should be started.
- Adequate disk space should be allocated for the distribution database.
- Additional disk space has also to be reserved for the publisher and subscriber's databases. Microsoft SQL Server 7.0/2000 uses unique identifier column to identify each row during the merge replication. If the tables do not have a unique identifier column with the ROWGUIDCOL property, Microsoft SQL Server 7.0/2000 will automatically add this column to the table. Therefore, additional disk space will be required to store the data.
- It is not possible to replicate only one table with foreign key constraint. All referenced tables should be included in the publication.
- Make sure that the server being replicated to is defined as a remote server.
- Check that the Distributor, Publisher, published database and subscribers are defined.

Steps to follow when setting a publication

Configuring Distributor, Publisher, Publication database and Subscribers

From inside the Enterprise Manager and after locating the Publisher Server, select **Tools => Replication => Configure Publishing, subscribers and distribution.**

Create a new publication and choose the replication method

From inside the Enterprise Manager and after locating the Publisher server, select **Tools => Replication => Create and Manage Publications**

The **Create and Manage Publications** dialog box is displayed. Choose the database that you need to publish and click the **Create Publication** button. The **Create Publication** wizard is launched.

- Choose the database that you need to publish and click the Next button.
- Select **Merge Publication** and click the Next button
- Select the types of subscribers you expect to subscribe to this publication and click the Next button
- Check **Publish All** checkbox to publish the tables that you need to replicate from the database and click the Next button (the list of Visual Dolphin tables are displayed at the end)
- Click Next to add the uniqueidentifier columns to all tables in the published database and set **NOT FOR REPLICATION** option to the IDENTITY column of the jobs table
- Specify the **publication name** and click Next

- Data filters can be specified at this step. In case of publication per company, you can check for each table its filtering condition like, for ex.:

The filtering condition for the table **ft_c_joub** is,

```
select * from ft_c_joub where jb_cmp_seq=1
```

If we need to create a publication for the company number 1

The list of tables and their related filtering conditions can be found at the end of this document.

- If the filtering conditions are not required then Check **No**, create the publication as specified and click Next
- Click the **Finish** button to create the publication. Click the Close button

Create a New Subscription

There are two ways to create a new subscription on an existing publication:

- Pulling a subscription
- Pushing a subscription

Pull Subscriptions

Pull subscriptions are created at the Subscriber, and the Subscriber requests data and updates made at the Publisher. Pull subscriptions allow the user at the Subscriber to determine when the data changes are synchronized, which can be on demand or scheduled.

Use Pull subscriptions when:

- Administration of the subscription will take place at the Subscriber.
- The publication has a large number of Subscribers (for example, Subscribers using the Internet), and when it would be too resource-intensive to run all the agents at one site or all at the Distributor.
- Subscribers are autonomous, disconnected, and/or mobile. Subscribers will determine when they will connect to the Publisher/Distributor and synchronize changes.

Data will typically be synchronized on demand or on schedule rather than continuously. One feature of the pull subscription is that the Distribution Agent for snapshot and transactional publications and the Merge Agent for merge publications run at the Subscriber. This can result in a reduction of the amount of processing overhead on the Distributor.

However, if we need the Distribution Agent or Merge Agent to run at the Distributor, we can offload agent processing from the Subscriber. For example, this option is used if the Subscriber determines when he is connected to the network and ready to synchronize, but we need to run the agent at the Distributor for better processing power at the Distributor. *For more information, see Remote Agent Activation.*

Another feature of Pull subscriptions is that members of the **sysadmin** or **db_owner** roles at the Subscriber decide which publications are received and when. Each Subscriber can have subscriptions to multiple publications at different Publishers.

To create a subscription, a publication at the Publisher and a subscription database at the Subscriber must exist. We can create the subscription database before creating the subscription, or specify a new subscription database in the Create Pull Subscription Wizard. We can create a pull subscription to any publication that has been enabled for pull subscriptions on a registered Publisher.

To create a pull subscription, we need to specify:

- The name of the subscription database,
- Whether the Snapshot Agent should create an initial snapshot and the Distribution Agent or Merge Agent should apply this snapshot at the Subscriber,
- The location of the snapshot files to apply when initializing the subscription,
- The priority of the subscription for merge,
- For snapshot replication and transactional replication, specify that the subscription will use immediate updating, queued updating, or transform published data options (these must first be enabled when creating the publication),
- Whether the Distribution Agent or Merge Agent runs continuously, on demand, or on a scheduled basis,
- Services that will be started to create the subscription.

Push subscriptions and pull subscriptions are known as named subscriptions because information about the subscription and the Subscriber is stored at the Publisher and performance information about the Subscriber is stored at the Distributor. This is in contrast to anonymous subscriptions (a different type of pull subscription) for which information about the subscription and the Subscriber is not stored.

When a pull subscription is created and a push subscription for the publication already exists for the Subscriber, an error message explains that the push subscription already exists and that any push subscriptions should be dropped before proceeding.

When creating a pull subscription and if another pull subscription to the same publication already exists, it is required to drop the existing subscription before adding the new one unless the first subscription has expired.

Push Subscription

Push subscriptions can simplify and centralize subscription administration because it eliminates the need to administer each Subscriber individually. The Distribution Agent or Merge Agent runs at the Distributor when synchronizing a push subscription.

Push subscriptions are created at the Publisher, and the replication agents propagate data and updates to a Subscriber without a request from the Subscriber. Changes can also be pushed to Subscribers on a scheduled basis.

Use Push subscriptions when:

- Data will typically be synchronized on demand or on a frequently recurring schedule.
- Publications require near real-time movement of data without polling.
- The higher processor overhead at a Publisher using a local Distributor does not affect performance
- Easier administration from a centralized location (the Distributor) is required.

The centralized Distributor will establish the schedule on which connections will be made with remote, occasionally connected Subscribers.

With push subscriptions, the Distribution Agent (for snapshot and transactional publications) or the Merge Agent (for merge publications) runs at the Distributor. However, if we need to offload agent processing from the Distributor but retain some of the benefits of easier administration, we can run the agent at the Subscriber. *For more information, see Remote Agent Activation.*

Since remote agent activation is available, the determining factors to consider when setting up subscriptions is what type we will need (push, pull or anonymous) and where the replication agent will run.

Users who are members of the **sysadmin** or **db_owner** roles at a Subscriber can set up a push subscription. However, to enable a member of the **db_owner** role to set up a push subscription, a member of the **sysadmin** role must register the Subscribers.

To create a subscription, a publication at the Publisher and a subscription database at the Subscriber should exist. The subscription database can be created before creating the subscription, or a new subscription database specified in the Create Push subscription Wizard. We can create a push subscription for any Subscriber that is enabled in the Publisher and Distributor properties.

Push and pull subscriptions are known as named subscriptions because information about the subscription and the subscriber is stored at the Publisher, and performance information about the Subscriber is stored at the Distributor. This is in contrast to anonymous subscriptions (which are a type of pull subscription) for which little or no information about the subscription and the Subscriber is stored.

When we create a push subscription, we specify:

- The name of the Subscriber,
- The name of the subscription database.,
- Whether the Distribution Agent or Merge Agent runs at the Distributor (default) or at the Subscriber using remote agent activation,
- Whether the Distribution Agent or Merge Agent runs continuously, on a scheduled basis or on demand only,

- If the Snapshot Agent should create an updated initial snapshot for the subscription and if the Distribution Agent or Merge Agent should apply that snapshot at the Subscriber,
- For snapshot or transactional publications that allow immediate updating or queued updating, the options that this subscription will use (available if advanced options are enabled in the Push subscription Wizard),
- For merge replication, the priority value for the changes made in the database to be used during conflict detection and resolution,
- For snapshot replication and transactional replication, specify that the subscription will use immediate updating, queued updating, or transform published data options (these must first be enabled when creating the publication).
- Services that will be started to create the subscription.

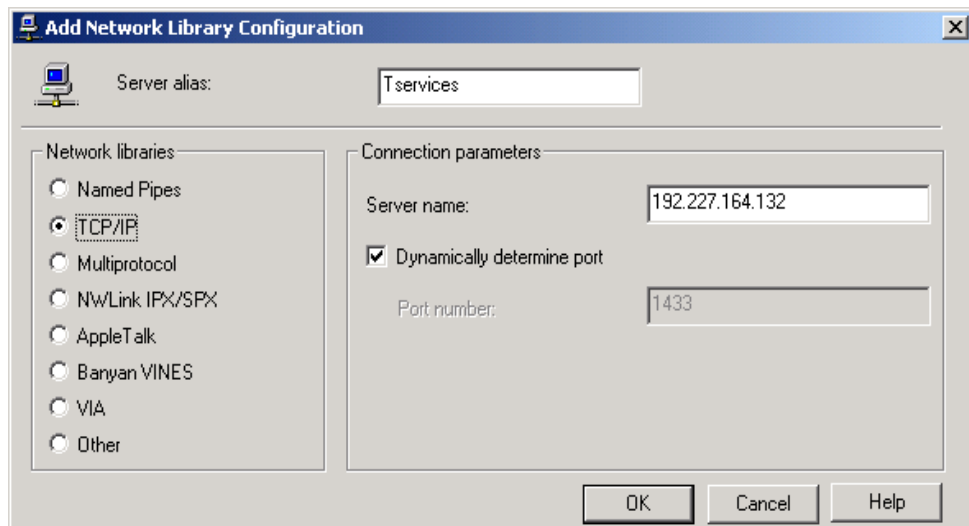
Step by step example

Setup

In this example, we will use only one server to replicate data: Publisher, Subscriber and Distributor databases will reside on the same machine. We will use Merge replication with pull subscription.

For setting up the Merge replication, the standard setup from the SQL Server Enterprise Manager can be used. First of all, the new remote server in which the replication will happen should be registered. The steps are as follows:

- Run the Client Network Utility Program: Start → All Programs → Microsoft SQL Server → Client Network Utility.



- A new screen will open. Click on Alias Tab and then Press the Add Button to define the new Remote Microsoft SQL Server.

- Run after the SQL Enterprise Manager Program:

Start → All Programs → Microsoft SQL Server → Enterprise Manager → Click on SQL Server group → Right + click and choose New SQL Server Registration.

A new SQL Server registration wizard screen will open. It requires a new remote SQL Server registration.

- In the SQL Server registration wizard:

Press Next → Choose the SQL Server that you need to register from the list of available servers and press >>, and then Press Next. In the select an Authentication Mode, choose the option to connect using the SQL Server Login information that was assigned by the system administrator (SQL Server Authentication), and then Press Next.

Put The Login Name that corresponds to the username created from Microsoft SQL Server for replication purpose, and then Press Next. In the Select SQL Server Group screen, the option Add the SQL server(s) to an existing SQL Server Group must be selected and then Press the Next button.

Finally, press the button Finish to end the registration procedure. If any error occurs at validation, the reason of failure should be checked. It could be the IP address, the protocol, connection with the remote server, incorrect username or wrong password etc...

Configuration

We will replicate data from the ADMDUBAI database into ADMDBCMP1 database, by using a Merge replication with the filter by column option in order to replicate from the Main database to ADMDBCMP1 only the data of one company, knowing that ADMDUBAI database contains data for more than two companies.

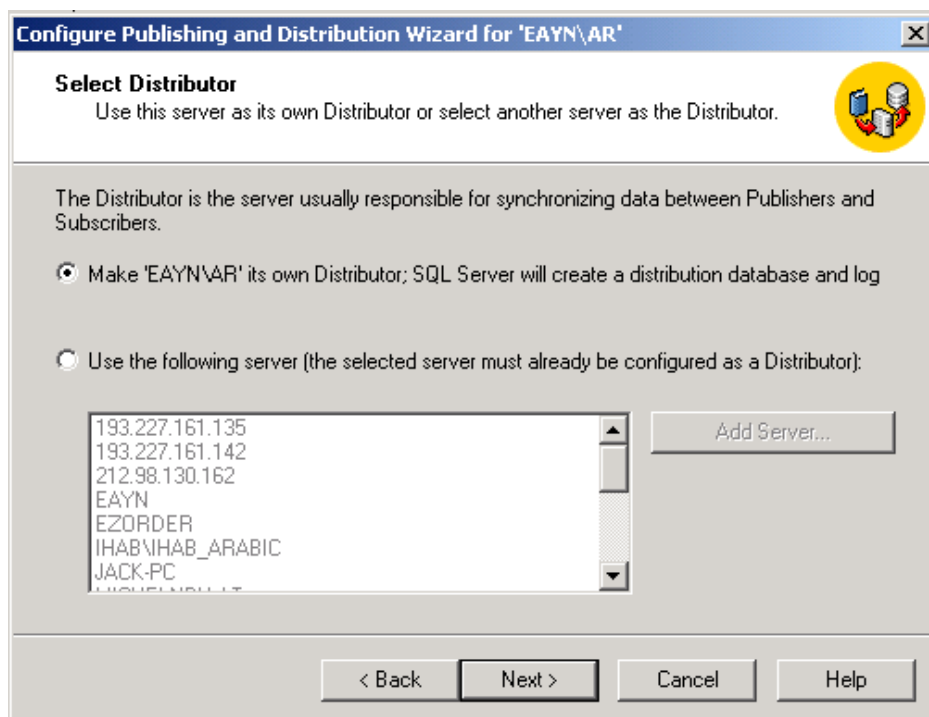
- Defining Distributor, Publisher and subscribers:

Select

Tools =>

Replication =>

Configure Publishing, Subscribers, and Distribution...



- Configure SQL Server Agent service to start automatically when the computer is started. Check Yes, configure the SQL Server Agent service to start automatically and click the Next button
- Specify snapshot folder using a network path and click the Next button.
- Customize the publishing and distribution settings, or choose the default settings. Check *No, use the following default settings* and click the Next button,
- Click the Finish button
- Microsoft SQL Server created the distribution database, enabled publishing, and installed the distributor. Once completed, click OK.
- CHIGRIK\SQL2000 was installed as Distributor. The replication monitor has been added to the console tree on CHIGRIK\SQL2000 server. Close the screen.

Create a New Publication

- Start creating publications and articles:
 - Select Tools =>
 - Replication =>
 - Create and Manage Publications
- From the Create and Manage Publications dialog box, choose ADMDUBAI database and click the Create Publication button
- The Create Publication wizard is launched. Click the Next button.
- Choose the ADMDUBAI database and click the Next button
- Select Merge publication and click the Next button
- Select the types of Subscribers that you expect to subscribe to this publication and click the Next button
- Check Publish All checkbox to publish all tables from the ADMDUBAI database and then de-select all the tables that are not for replication (The list of tables to replicate according to the version 5.0.xx Inventory and Version 2.4.xx Accounting can be found in the appendix), and then click the Next button.
- Click the Next button to add uniqueidentifier columns in all tables in the ADMDUBAI database and set NOT FOR REPLICATION option to the IDENTITY column of the jobs table
- Specify the name of the publication 'ADMDUBAICMP1' as the publication name and click the Next button
- Specify data filters on this step, and for the purpose of this example we will use a data filter on the tables that contain the company code in order to replicate data for a specific company only. Check *Yes, create the publication as specified* and click the Next button
- A new screen displays all the replicated tables with an option to specify the data filter on each table, if required. Press the button with the three dots (...) and add inside the select statement the *where* condition with the filtering criteria. (If replication is required for a specific company and not for all the companies, a list of tables to replicate with their related filters can be found in the appendix)
- Click the Finish button to create the publication
- The 'ADMDUBAICMP1' publication was created. Click the Close button.

Create a new Subscription

- Backup the Publisher database (after creating the subscription and restore it at subscriber)
- To create a new subscription, Click the Pull New Subscription button
- Select Enterprise Manager → Tools → Replication → Pull new Subscription.
- This will launch the Pull Subscription wizard. Click the Next button.
- In the Look for Publication, keep the option '*Look at publications from registered server*' selected and press Next
- In the Choose publication window, select the publication server, expand it and select the publication called 'ADMDUBAICMP1' and press next
- In the Specify Synchronization server login, choose the option '*Use SQL Server authentication*' and then put the user and password that were created specifically for replication, and then click the Next button.
- In the Choose destination database screen, select the subscriber database and then press Next
- In the Allow anonymous subscriptions, choose the option '*No, this is a named subscription*' and then press Next
- Choose the option '*No, the subscriber has already the schema and data*' and press Next. Since we already have a backup of the publisher database, we need to initialize the schema and data again from publisher to subscriber
- Choose the way of replicating data according to a certain schedule, on demand or online replication and then Press Next.
- The set subscription priority option must be equal to the following: '*Use the Publisher as a Proxy for the subscriber when resolving conflicts*' and then press Next twice and then Press the Finish button.
- Click the Close button to close the Create and Manage Publication dialog window.
- The last step of setting up Merge replication should be the creation a current script of the replication settings. It can be useful to help recover the replication settings in the event of server failure.

- | |
|---|
| <ul style="list-style-type: none">- After setting the Publication a script that Disable all Foreign Keys from being replicated should be executed. Otherwise, many conflicts will be faced while synchronizing the data.- After setting the Publication, a script that Disable the triggers that update the user who created the record or the last user who updated the record from being replicated should also be executed. Otherwise, many conflicts may occur while synchronizing the data.- After setting a subscription, a script to update the starting sequences at the subscriber to prevent data duplication should also be executed.- Restrictions: All Families of items, types of items, batches, must be created and managed only from Back office. User access should not be given to these tables from the subscriber site. |
|---|

Backup and restore strategies

The backup and restore strategies are different for each replication types. In these lines, the backup and restore strategies for the Merge replication is described.

Since Merge replication is more complicated than Snapshot or Transactional replication, and usually used when an update of data on both publisher and subscribers is required, backup and restore strategies should be planned more carefully.

There are four main strategies for backing up and restoring Merge replication:

- Backup Publisher, master and model databases.
- Backup Publisher, Distributor, master and model databases.
- Backup Publisher, Subscriber(s), master and model databases.
- Backup Publisher, Distributor, Subscriber(s), master and model databases.

- 1) Backing up Publisher, Master and Model databases is the simplest strategy. This strategy has its advantages and disadvantages.

The main advantage is that it requires the least amount of storage resources and do not require coordinating the backup with the backup of any other servers.

The main disadvantage of this strategy is that we may need to setup replication from the beginning in the event of a Publisher or Distributor failure. With this strategy, we should backup the publication database after changing existing publications, or after the new publications were added, or whenever changes are made to a replicated objects schema (for example, adding or dropping a column).

- 2) Backing up Publisher, Distributor, master and model databases is a rarely used strategy, than the first one, because in most cases, it is not necessary to restore a distribution database when restoring a publication database backup for merge publications. Usually distribution database does not store any data used in change tracking and it does not provide temporary storage of merge replication changes.

The main disadvantage of this strategy is that the Publisher and Distributor's databases should be backed up simultaneously (or as closely as possible). It also requires more computing and storage resources than the first way.

- 3) Backing up Publisher, Subscriber(s), master and model databases is used when some changes were made on Subscriber(s) and these changes are required to be synchronized with the publication database.

- 4) Backup Publisher, Distributor, Subscriber(s), master and model databases is the most complex backup strategy.

Its main advantage is that in the event of a Publisher, Distributor or Subscriber(s) failure, we can quickly restore the failed database without setting up replication from the beginning.

The disadvantage of this strategy is that the Publisher and Distributor's databases should be backed up simultaneously (or as closely as possible), and this strategy requires lot of computing and storage resources.

For each strategy, the **msdb** and master databases on the Publisher, Distributor and Subscriber should be backed up. **msdb** database is used by SQL Server Agent for scheduling alerts and jobs. The master database is the main system database that contains entries per subscriber, every login account, system configuration settings ...

It is strongly recommended that a current script of the replication settings is created and kept. It can be useful to help recover the replication settings in the event of server failure.

How to install a new Visual Dolphin Release on a replicated database

Preparation Phase

The preparation phase may be initiated few days in advance.

- Prepare the releases, newsletters, list of changes in the database, OCX ...
- Install the latest releases in a test environment
- Test the release on backup data within the same environment. (Ideally perform the test on a simulation of a locally replicated data).
- In case of a large number of sites, organize the deployment per site and make sure that all tasks to be executed are listed.
- Install the EXE at the sites: Download remotely the systems (all modules), reports, special printouts and external triggers to be executed on a spare directory at the subscribers.
- Make a safe copy of the current release (exe) and reports on a backup directory.
- Advise the subscribers to keep their servers running and log the users out of the Visual Dolphin programs at the time of installation.

Deployment of the New Release

Depending on the system usage by the users, it is preferable that the procedure is executed during the night or on a non-working day.

- Make sure that all sites are accessible from the Publisher server (Head Office).
- Replicate all remote databases.
- Make remotely a local backup of all the sites to be replicated (by naming the backups: Database + name of the site + YYMMDD)
- Run a special Backup in the Head Office (data & programs).
- Stop replication.(make sure before stopping the replication that it is not working)
- Make a copy of the old Visual Dolphin release (EXE) in a folder in all the sites.
- Install new release, EXE, Reports and client's specials on back office.
- Run the new release (EXE) on back office and control that the replication updates the fields

This operation may take 5 minutes to 2 hours depending on the number of subscriptions. The execution respects the following order: Finance, Inventory, Retail then remaining modules, if any.

- Make sure that all fields are updated and created: Check at the publisher if stored procedures, triggers, indexes, views, user sequences tables, and general parameters, are being updated
- Full test the programs.
 - Check the users security table
 - Create a new invoice & print it.
 - Create retail invoice & print, if applicable
 - Create retail receipt & print, if applicable
 - Create JV & print.
 - Make the close Shift, if applicable
 - Test of menus with ocx
- Prepare a new backup for each site at the Head Office, named Database + name of the site + YYMMDD
- Start Replication on site
- Check changes of data structure if replicated to all subscribers.
- Run the diagnostic program to compare the data structure with few sites.
- Install the EXE & Report special to all Sites.
- Check at the subscriber if stored procedures, triggers, indexes, Views, user sequences tables, and general parameters, are being updated (to match with back office)

- Make full test to all sites:
 - Check the users security table
 - Create a new invoice & print it.
 - Create retail invoice & print, if applicable
 - Create retail receipt & print, if applicable
 - Create JV & print.
 - Make the close Shift, if applicable
 - Test of menus with ocx

- If the Retail module is installed, notify all users to make open shift and set the cash balance

Appendix: List of tables to replicate and their Filtering Conditions

Table Name	SubsetFilter
fm_actype	NULL
fm_c_auxb	auxb_cmp_seq=1
fm_c_batch	bat_cmp_seq=1 (*)
fm_c_branch	br_cmp_seq=1
fm_c_budget	bd_cmp_seq=1
fm_c_budgeta	bda_cmp_seq=1
fm_c_budgetb	bdb_cmp_seq=1
fm_c_chartc	ch_c_cmp_seq=1
fm_c_costc	c_cmp_seq=1
fm_c_costd	d_cmp_seq=1
fm_c_diag_param	dp_cmp_seq=1
fm_c_diffexch	df_cmp_seq=1
fm_c_fiscal	fis_cmp_seq=1
fm_c_param	cp_cmp_seq=1
fm_c_ppaya	ppa_cmp_seq=1
fm_c_ppaya_notes	pn_cmp_seq=1
fm_c_ppayb	ppb_cmp_seq=1
fm_c_ppayc	ppc_cmp_seq=1
fm_c_ppayd	ppd_cmp_seq=1
fm_c_prdbudg	pbd_cmp_seq=1
fm_c_project	prj_cmp_seq=1
fm_c_security	sec_cmp_seq=1
fm_c_shipment	sh_cmp_seq=1
fm_c_user_param	up_cmp_seq=1
fm_c_user_privilege	usp_cmp_seq=1
fm_chart	NULL
fm_eqchart	NULL
fm_language	NULL
fm_sdcura	NULL
fm_sdcurb	NULL
ft_c_joua	ja_cmp_Seq=1
ft_c_joub	jb_cmp_seq=1
ft_c_jouc	jc_cmp_seq=1
ft_c_joud	jd_cmp_Seq=1
ft_c_schedule	sch_cmp_Seq=1
im_accnb	ac_cmp_seq=1
im_brand	br_cmp_seq=1
im_buyer	buy_cmp_seq=1
im_buyercat	buc_cmp_seq=1
im_c_custnotes	custn_cmp_seq=1
Im_c_fiscal	lfis_cmp_seq=1
im_c_gparam	gpc_cmp_seq=1
im_c_Maintgparam	gpmaint_cmp_seq=1
im_c_retail_group_param	ret_grp_cmp_seq=2
im_c_retail_mode	mod_cmp_seq=2
im_c_retail_param	par_cmp_seq=2
Im_c_security	sec_cmp_seq=1
im_c_transfer_a	ta_cmp_seq=1
im_c_transfer_b	tb_cmp_seq=1
im_c_user_param	up_cmp_seq=1

(*) if we need to replicate it, we should remove the replication trigger on update, because if we keep it, the batch sequence will be always overwritten, otherwise every time we add a new batch we should then tell all our remote sites)

Table Name	SubsetFilter
im_c_user_privilege	usp_cmp_seq=1
im_catalog_a	cta_cmp_seq=1
im_catalog_b	ctb_cmp_seq=1
im_CatgLevel	lv_cmp_seq=1
im_custcat	cc_cmp_seq=1
im_custom	cu_cmp_seq=1
im_customer	cust_cmp_seq=1
im_customerudf	cmp_seq=1
im_discount	disc_cmp_seq=1
im_division	di_cmp_seq=1
im_family	fa_cmp_seq=1
im_invtype	invt_cmp_seq=1
im_item_budget	ibud_cmp_seq=1
im_item_cost	icost_cmp_seq=1
im_item_kit	ik_cmp_seq=1
im_item_main	im_cmp_seq=1
im_itea	ia_cmp_seq=1
im_iteb	ib_cmp_seq=1
im_itemc_lot	icl_cmp_seq=1
im_itemSup	is_cmpseq=1
im_Maintgparam	NULL
im_points	ps_cmp_seq=1
im_prcode	pc_cmp_seq=1
Im_PriceA	pra_cmp_seq=1
Im_PriceB	prb_cmp_seq=1
Im_PriceC	prc_cmp_seq=1
Im_PriceC_temp	NULL
im_prlst	NULL
im_Prodtype	prodt_cmp_seq=1
im_promcode	prmc_cmp_seq=1
im_promlst	NULL
im_pswar	pw_cmp_seq=1
im_reason	re_cmp_seq=1
im_retail_shortcut	sht_cmp_seq=2
im_routera	roa_cmp_seq=1
im_routerb	rob_cmp_seq=1
im_sal_catalog	sc_cmp_seq=1
im_salcat	sac_cmp_seq=1
im_salesman	sal_cmp_seq=1
im_salsmroutera	sra_cmp_seq=1
im_salsmrouterb	srb_cmp_seq=1
im_salsmrouterc	src_cmp_seq=1
im_seasonality	se_cmp_seq=1
im_supcat	suc_cmp_seq=1
im_supplier	sup_cmp_seq=1
im_type	ty_cmp_seq=1
im_typecatg	tc_cmp_seq=1
im_unit	un_cmp_seq=1
im_uom	uo_cmp_seq=1
im_vat	va_cmp_seq=1
im_warehouse	wa_cmp_seq=1
lt_c_adjcost_a	adj_cmp_seq=1
lt_c_adjcost_b	adjb_adj_cmp_seq=1
it_c_compa	cpa_cmp_seq=1
it_c_compb	cpb_cmp_seq=1
it_c_prod_a	prda_cmp_seq=1
it_c_prod_b	prdb_cmp_seq=1
it_c_prod_charges	prdc_cmp_seq=1

Table Name	SubsetFilter
it_c_purchase_a	pra_cmp_seq=1
it_c_purchase_b	prb_cmp_seq=1
it_c_purchase_specs	prs_cmp_seq=1
it_c_request_a	reqa_cmp_seq=1
it_c_request_b	reqb_cmp_seq=1
it_c_retail_a	rea_cmp_seq=2
it_c_retail_b	reb_cmp_seq=2
it_c_retail_cashStatus	cas_cmp_seq=2
it_c_retail_cashStatus_detail	casd_cmp_seq=2
it_c_retail_rate	rat_cmp_seq=2
it_c_retail_reca	reca_cmp_seq=2
it_c_retail_recb	recb_cmp_seq=2
it_c_retail_recc	recc_cmp_seq=2
it_c_retail_serial	res_cmp_seq=2
it_c_schedule	sch_cmp_seq=1
it_packa	tra_cmp_seq=1
it_packb	trb_cmp_seq=1
it_trans_a	tra_cmp_seq=1
it_trans_add	tad_cmp_seq=1
it_trans_b	trb_cmp_seq=1
it_trans_charges	trc_cmp_seq=1
it_trans_disc	trd_cmp_seq=1
it_trans_lot	trl_cmp_seq=1
it_transfer	NULL
it_transnotes	trn_cmp_seq=1
menus	NULL
rp_activity	NULL
rp_other_act	NULL
rpaux	NULL
rpaux1	NULL
rpauxAddress	NULL
rpauxdays	NULL
rpauxgroup	NULL
rpauxname	NULL
rpauxnotes	NULL
rpauxphone	NULL
rpauxtype	NULL
rpauxudf	NULL
rpcity	NULL
rpcountry	NULL
rpreion	NULL
sd_logfile	NULL
sd_oleobject	NULL
sd_oleobject_m	NULL
sd_udf	NULL
sd_udf_m	NULL
sdcomp	cmp_seq=1
sdoleobject	NULL

New Tables created in version 5 and that need to be replicated

Financial

Table Name	SubsetFilter
fm_c_saparam	Sp_cmp_seq=1
fm_c_ppaya_notes	Pn_cmp_seq=1
fm_c_param_mc	cm_cmp_seq=1
fm_c_saparam	Sp_cmp_seq=1
sd_udf_m	Null
sd_udf	Null
rpauxudf	Null
rpauxgroup	NULL
sd_oleobject_m	NULL
sdoleobject	NULL

Inventory

Table Name	SubsetFilter
lm_catglevel	Lv_cmp_seq=1
lm_discount	disc_cmp_seq=1
lm_catalog_a	cta_cmp_seq=1
lm_catalog_b	ctb_cmp_seq=1
lm_seasonality	Se_cmp_seq=1
lm_typeCatg	Tc_cmp_seq=1

Remark definition tables (exist only if the tables are installed)

Table Name	SubsetFilter
Sd_remarka	rma_cmp_seq=1
Sd_remarkb	rmb_cmp_seq=1

Third Parties Tables

Table Name	SubsetFilter
IM_sal_catalog	sc_cmp_seq=1
IM_routera	roa_cmp_seq=1
IM_routerb	rob_cmp_seq=1
IM_salsmroutera	sra_cmp_seq=1
IM_salsmrouterb	srb_cmp_seq=1
IM_salsmrouterc	src_cmp_seq=1
IM_customerudf	cmp_seq=1
IM_c_custnotes	custn_cmp_seq=1

Inventory Transaction Tables

Table Name	SubsetFilter
lt_trans_remark	trr_cmp_seq=1

Procurement tables (exist only if the procurement module is installed)

Table Name	SubsetFilter
IM_c_procsecurity	sec_cmp_seq=1
It_c_procRFQ_A	rfqa_cmpseq=1
It_c_procRFQ_B	rfqb_cmpseq=1
It_c_procRFQ_SUP	rfqs_cmpseq=1
IT_c_procrs_a	prsa_cmpseq=1
IT_c_procrs_b	prsb_cmpseq=1

Warehouse Delivery tables (exist only if the module is installed)

Table Name	SubsetFilter
IM_c_truck	tk_cmp_seq=1
It_c_ddista	da_cmp_seq=1
It_c_ddistb	db_cmp_seq=1



Lebanon	Phone	961-1-399855
	Fax	961-1-380420
	PO.Box	166607 – 1100 2140 Ashrafieh Beirut – Lebanon
United Arab Emirates	Phone	971-4-3383318
	Fax	971-4-3383319
	PO.Box	35046 Al Romoul Dubai – U.A.E.
Saudi Arabia		
	Riyadh	Phone 966-1-4730784
		Fax 966-1-4730784
	Jeddah	Phone 966-2-6608824
		Fax 966-2-6603835
		PO.Box 5586 Riyadh 11432 - Kingdom Of Saudi Arabia
	Email	sdcg@softwaredesign.com.lb
	Website:	http://www.sd-lb.com