SO…

- You get that call
- You know the one
- At 3:30 am
- “I’m glad you are awake”
- “I think there is a problem with …replication”
- “Oh yeah, everything needs to be working by 6:00 am for the data warehouse load”
- “They said they made some changes to things yesterday. Not sure what things but things happened and things aren’t happening now”
- “Thanks, let us know every 10 or so minutes on how things are going”
- “Good Luck”

So what do you do now!!!

Help! My Replication Has Fallen and We Can't Get it Replicating again

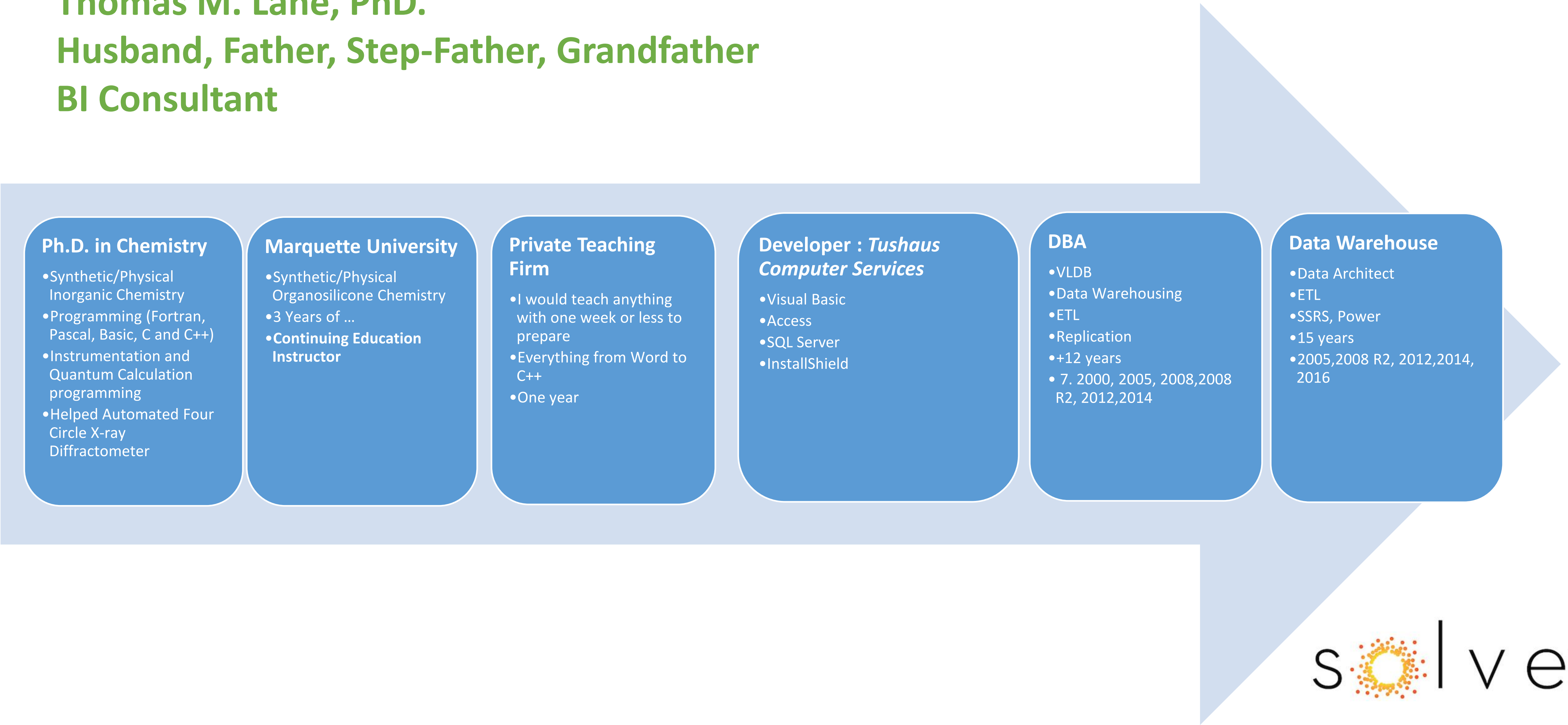
Thomas M. Lane, Ph.D.

doclane@gmail.com

@doclane

www.Solve100.com

Thomas M. Lane, PhD.
Husband, Father, Step-Father, Grandfather
BI Consultant



Ph.D. in Chemistry
- Synthetic/Physical Inorganic Chemistry
- Programming (Fortran, Pascal, Basic, C and C++)
- Instrumentation and Quantum Calculation programming
- Helped Automated Four Circle X-ray Diffractometer

Marquette University
- Synthetic/Physical Organosilicone Chemistry
- 3 Years of …
- **Continuing Education Instructor**

Private Teaching Firm
- I would teach anything with one week or less to prepare
- Everything from Word to C++
- One year

Developer : *Tushaus Computer Services*
- Visual Basic
- Access
- SQL Server
- InstallShield

DBA
- VLDB
- Data Warehousing
- ETL
- Replication
- +12 years
- 7. 2000, 2005, 2008,2008 R2, 2012,2014

Data Warehouse
- Data Architect
- ETL
- SSRS, Power
- 15 years
- 2005,2008 R2, 2012,2014, 2016

Before we Begin

- Only an Hour Talk so I am only to look at Transactional Replication
- In order to trouble shoot you need

to know all the parts

- Lots and lots of moving parts
  - All working together
  - To make one product with many flavors
- Data about the publications is

stored in tables in distribution database

- Data managed in the transaction logs and

MSrepl_commands

Parts of Replication

Follows a magazine/book producing terminology

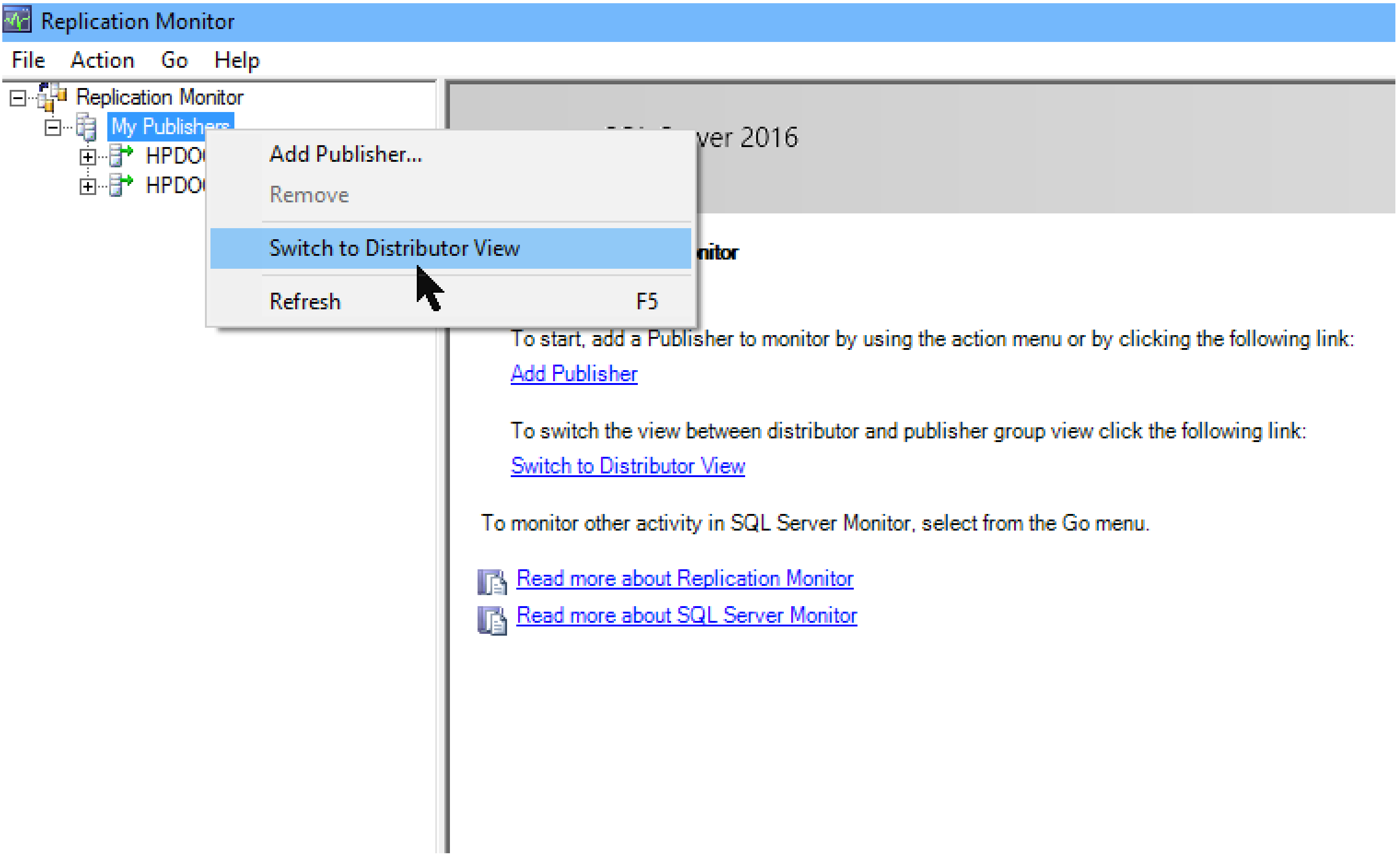
- **Articles**
  - Each article corresponds to a single SQL Server object, or a subset of an object. The objects that are replicated most often are tables, views and stored procedures.
- **The Publication**
  - A group of articles that logically belong together can be combined into a publication. The publication has options defined that apply to all the articles in that publication. The main option defined by a publication is the type of replication that is to be used.
- **The Publisher**
  - The publisher monitors all articles for changes, and makes information about those changes available to the distributor

Parts of Replication

- **The Distributor**
  - SQL Server instance keeps track of all subscribers and all published changes and makes sure that each subscriber gets notified of each change. Most of the changes are tracked in a distribution database. The distributor can be a separate SQL Server instance, but often the distribution service runs on the same machine as the publisher.
- **Subscription**
  - A subscription defines which server (subscriber) is to receive the updates published in a publication.
  - Link between one publication and one subscriber.
  - Push subscriptions and pull subscriptions.
- **Subscriber**
  - The subscriber is the SQL Server instance that receives all the published information through subscriptions.

Lets fix that pro

- Where do we begin
  - Calm your nerves
    - A shot of Tequila
    - Or Two
    - Check Linkedin
  - Open SSMS
  - Connect to your publisher
    - You can change view
  - Find bad publication
    - If you don’t see an error check the subscribers
    - View details
    - Click on the most recent error message



To start, add a Publisher to monitor by using the action menu or by clicking the following link:

Add Publisher

To switch the view between distributor and publisher group view click the following link:

Switch to Distributor View

To monitor other activity in SQL Server Monitor, select from the Go menu.

Read more about Replication Monitor

Read more about SQL Server Monitor

Lets see what is wrong with the Subscription

- Get the transaction sequence number from error message
  - 0x00000049000003880000400000000
- Run FindProblemReplicationCode.sql
- Latency
  - Tokens vs Canary Table
- Subscriber data changes

Why did you let this happen!!!

- Why would someone call YOU to tell you there is an issue with replication?
    - This should never happen!
    - Get notification
        - Tracer Token
        - Canary
            - Add a table to publication
            - Only one record
            - Update record every
        - Which is best?
    - In transactional replication subscriber is updatable.
        - Can you make it read-only?
        - From some?

Canary

- Add a table named dbo.Canary_PubName to each publication
- dbo.Canary_PubName has a single row with a datetime column in it
- A SQL Server Agent job on the publisher updates the datetime to the current timestamp every minute
- A SQL Server Agent job on the subscriber checks dbo.Canary_PubName every minute and alerts if the difference between the current time and the timestamp is greater than N minutes

https://www.brentozar.com/archive/2014/07/monitoring-sql-server-transactional-replication/

Why did you let this happen!!!

- Run GenerateTriggers.sql
  - Leave access for replication user and a sql login
    - Password and user account should be hard to remember and perhaps warn the user.
  - Make sure they know what they are doing

Best Practices

- Script out everything and keep good copies
  - Make sure to add passwords N'password' in script
  - Test them
- Make replication user
  - Minimum required limited access to everything in replication
  - Password does not change
  - User can't change password
  - Domain account
  - May require cred and proxy setup
- Monitor
  - Latency
  - Errors

Best Practices

- Limit changes to publications
  - These will not require re-initialization of snapshot (invasive)
    - ALTER TABLE
    - ALTER VIEW
    - ALTER PROCEDURE
    - ALTER FUNCTION
    - ALTER TRIGGER
  - Consider leaving old column by adding defaults
- Limit the number of row and columns replicated
- Remember each process run an application that uses resources
  - Important when considering Pull or Push subscriptions
- Indexes and Keys are not replicated
  - Use on either side to improve performance

Best Practices

- Consider optimizing the Agent Profiles
- Name all table constraints

General Considerations

- Could not find stored procedure 'sp_MSins_<table_name>‘ run:
  - sp_scriptPublicationcustomprocs ‘name of pub’
- Explicitly adding, dropping, or altering indexes is not supported. Indexes created implicitly for constraints (such as a primary key constraint) are supported.
- Altering or dropping identity columns that are managed by replication is not supported. It is recommended that constraints be explicitly named.
- If a table is published for replication, it is not possible to alter a column in that table to a data type of XML if a publication snapshot has already been generated. To alter the column, you must first remove replication.

General Considerations

- Schema changes should be made only at the Publisher
- Changes propagated to a republishing Subscriber are by default propagated to its Subscribers.
- If the schema change references objects or constraints existing on the Publisher but not on the Subscriber, the schema change will succeed on the Publisher but will fail on the Subscriber.
- All objects on the Subscriber that are referenced when adding a foreign key must have the same name and owner as the corresponding object on the Publisher.

Lets create a Distributor

- Best practice is to have a separate distributor from publisher and subscribers
  - 3 nodes minimum
- Use Wizard but script and save file
  - Always change passwords in replication script files and resave
  - Null replace with N'*password*'
- Check Job list before
- Right click replication (HPDOCLAPTOP\SUBSCRIBER)
  - This snapshot folder does not support **pull** subscriptions created at the Subscriber. It is not a network path or it is a drive letter mapped to a network path. To support both push and pull subscriptions, use a network path to refer to this folder.
- Now look at the Jobs
- 6 New Jobs

Thank you for your time

- Thomas Lane
  - @doclane
  - doclane@gmail.com
- Police: Canary in a Coal Mine
- Monty Python: Monty Python and the Holy Grail

Lets create a Distributor

- Agent history clean up: distribution
  - EXEC dbo.sp_MShistory_cleanup @history_retention = 48
  - By default the SQL Replication Agent History cleanup job runs every 10 minutes and deletes any historical data older the 48 hours.
  - Change? Replication runs every 2 hours, connections bad, etc.?
- Distribution clean up: distribution
  - EXEC dbo.sp_MSdistribution_cleanup @min_distretention = 0, @max_distretention = 72
  - The purpose of this code is to clean up any commands in the MSRepl_Commands table which are more than 72 hours old.
  - NOTE FOR PUBLICATION: If the immediate_sync column is set to 1 then all commands are being held for 72 hours. However, if it is set to 0 then commands are removed from MSRepl_commands the moment that they have been passed to all subscribers… not held for the full 72 hours.
  - exec dbo.sp_helppublication @publication = '‘
  - sp_changepublication @publication = ‘Test', @property = 'immediate_sync', @value = 'false'

Lets create a Distributor

- Agent history clean up: distribution
  - EXEC sys.sp_expired_subscription_cleanup
  - Checks the status of all the subscriptions of every publication and drops those that have expired.
  - Change?
- Reinitialize subscriptions having data validation failures
  - exec sys.sp_MSreinit_failed_subscriptions @failure_level = 1
  - This agent will check that the subscription is valid
- Replication agents checkup
  - sys.sp_replication_agent_checkup @heartbeat_interval = 10
  - this agent will identify agents that are not logging. It runs every 10 minutes.
- Replication monitoring refresher for distribution
  - exec dbo.sp_replmonitorrefreshjob
  - This agent supports the replication monitor.

Lets create a Distributor

- Creates distribution database
  - [dbo].[MSpublications]
    - Each publication
  - [dbo].[MSarticles]
    - Each Article
  - [dbo].[MSsubscriptions]
    - Each article in each subscription

Lets create a Publication

- New Publication
  - Setup a new Domain account with access to the snapshot location

    Login on Publication and distributor SQL Server

    ```
    USE [master]
    GO
    CREATE LOGIN [HPDOCLAPTOP\ReplicationUser] FROM WINDOWS WITH DEFAULT_DATABASE=[master]
    GO
    USE [distribution]
    GO
    CREATE USER [HPDOCLAPTOP\ReplicationUser] FOR LOGIN [HPDOCLAPTOP\ReplicationUser]
    GO
    USE [distribution]
    GO
    ALTER ROLE [db_owner] ADD MEMBER [HPDOCLAPTOP\ReplicationUser]
    GO
    ```
  - May need to create a credential
  - CREATE CREDENTIAL [HPDOCLAPTOP\ReplicationUser] WITH IDENTITY = N'HPDOCLAPTOP\ReplicationUser', SECRET = N'D1gestiveJuices‘
  - Setup SQL Server Agent Replication proxy

Lets create a Publication

- Two new Jobs created
  - Runs separate application not part of SQL Server Engine
  - Logreader Agent
    - {Servername}\{Instance}-Databasename-{1}
    - LogRead.exe
      - Extracts relevant committed transactions from the publisher's log, repackages them and sends them to the distributor in the correct sequence.
    - Publisher [HPDOCLAPTOP\SQL16] -PublisherDB [ReplTransDB] -Distributor [HPDOCLAPTOP\SQL16] -DistributorSecurityMode 1 -Continuous
  - Snapshot Agent
    - {Servername}\{Instance}-Databasename-{Pubname}-{1}
    - -Publisher [HPDOCLAPTOP\SQL16] -PublisherDB [ReplTransDB] -Distributor [HPDOCLAPTOP\SQL16] -Publication [SQL16_ReplTransPub] -DistributorSecurityMode 1
  - Check out Replication Monitor
  - Check out Task manager

Lets create a Subscription

- Create database on Subscriber server
- Add replication user
- Right click on the publication, click New Subscription
- If issues exists
  - Confirm replication user password
  - Make sure to change the package owner to SA
  - Change the “run agent” step to execute as the proxy account
  - Make sure replication user has access to snapshot folder (Full rights)
  - Make sure replication user has access to C:\Program Files\Microsoft SQL Server\1x0\COM (exec)
- Distribution Agent
  - Distrib.exe takes the snapshots, and log entries from the agents we’ve described, and dispatches them to the subscribers.
  - Check Task Manager

Lets create a Subscription

- Snapshot needs to be run at least once since the oldest transaction aged out
- If distribution database gets large change the two jobs to clean up more often
- Look are replication Monitor
- Look at agent status
- Tracers – can be done programmatically
- Question: How good is this tracer? Can you depend on it?
- Check out Warning
  - Configure alerts
  - Will need operator set up

Lets see what is wrong with the Subscription

- Run GenerateTriggers.sql
  - A access hole for one sql server login
  - Make sure they know what they are doing

debugging

```
sp_helptext 'sp_MSdel_dboF4201'
{CALL [sp_MSdel_dboF4201] (789211)}

CALL [sp_MSdel_dboF4201] (778870)
SET IDENTITY_INSERT [dbo].[F4211] ON
INSERT INTO [dbo].[F4211]
([SDKCOO]
           ,[SDDOCO]
           ,[SDDCTO]
           ,[SDLNID]
 ,[F4211_ID]
 )
 VALUES('xxx',0,'xx',0,789211)
SET IDENTITY_INSERT [dbo].[F4211] OFF
select * from [dbo].[F4211] where [F4211_ID] = 789211


select *
from msrepl_commands (nolock)
where command_id = <no in message>
```