

Microsoft SQL Server 2008

Handy tips for the busy DBA

Prepared by : Chandramohan Krishnan

Table of contents

Database Backup File Encryption in SQL Server 2008	2
Setting Up Database Mail for POP3 Account.....	3
Powershell.....	4
Installation	4
Uninstalling older PowerShell Versions	4
Getting Started.....	5
Useful Code Snippets.....	5
Positional Parameter	5
Filtering, Sorting, Formatting & Exporting	5
Examples	5
Example 1.....	5
Example 2.....	6
Example 3.....	6
DDL Triggers (Tracking DB Schema Changes)	6
Creating a DDL Trigger	6
DDL event groups.....	7
Useful performance Tuning queries	8
Useful SQL Commands	9
Collation Set Commands.....	9
Who are the Sysadmins	9
Monitoring	9
Disk space used by MSSQL Server	9
Create reporting table and SP	10
Quick way to find space used by each table in a database	10
Simple script using cursors to execute DBCC UPDATEUSAGE	10
Replication Stuff.....	12
Duplicate key errors (error 20598) on resuming replication to a restored replicate database from the primary	12
Procedure to Drop Push Subscription	13
Procedure to rebuild replication on MS SQL Server	13
Steps.....	13
Note A.....	13
Note B	14
Subscription Deactivation and Expiration.....	15

Database Backup File Encryption in SQL Server 2008

-- We can use database encryption in SQL Server 2008 to encrypt a database backup to prevent it from being install and another server, without the need to the encryption certificate and private key. This example explains how this can be done and how to restore the database.

```
-- Create the server wide 'MASTER' key
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'p@ssw0rd1'
go
--Now create a certificate for use with our database
CREATE CERTIFICATE adventureworks2008Cert
WITH SUBJECT = 'My DEK Certificate for adventureworks2008 database'
go
--set the encryption for the Northwind database by creating a database
--encryption key and password using the certificate we just created.
use adventureworks2008
go
```

```

CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE adventureworks2008Cert
go
--enable encryption on the database level.
ALTER DATABASE adventureworks2008
SET ENCRYPTION ON
go
--check you have set encryption on (should be set to '1')
SELECT name, is_encrypted FROM sys.databases
GO
--#####--
--well done the database is encrypted. Now lets test
--#####--
--We can only restore this backup to a server that holds
-- a valid certificate
BACKUP DATABASE adventureworks2008
TO DISK = 'd:\mssql_dumps\adventureworks2008.bak'
WITH INIT, STATS = 10
go

--If you are using the same server to test this you will need to drop
-- the certificate, so export it FIRST!!
use master
go
BACKUP CERTIFICATE adventureworks2008Cert
TO FILE = 'd:\mssql_dumps\adventureworks2008Cert_File.cer'
WITH PRIVATE KEY (FILE = 'd:\mssql_dumps\adventureworks2008Cert_Key.pvk',
ENCRYPTION BY PASSWORD = 'pwd1234' )
GO
--drop if on the same server
DROP CERTIFICATE adventureworks2008Cert
go
--Now try to restore the DB backup. It should fail
--with msg "Cannot find server certificate"
RESTORE DATABASE adventureworks2008
FROM DISK = 'd:\mssql_dumps\adventureworks2008.bak'
WITH REPLACE, STATS=10
go
--Import the certificate (if you are on a different SQL server,
--create a MASTER KEY first)
CREATE CERTIFICATE adventureworks2008Cert
FROM FILE = 'd:\mssql_dumps\adventureworks2008Cert_File.cer'
WITH PRIVATE KEY (FILE = 'd:\mssql_dumps\adventureworks2008Cert_Key.pvk' ,
DECRYPTION BY PASSWORD = 'pwd1234')
go
--Finally, retry the database RESTORE. This time it should work
-- Congratulations!

```

Setting Up Database Mail for POP3 Account

1. Expand **Management**, right-click **Database Mail**, and select **Configure Database Mail**.
2. Choose the **Set up Database Mail** option to set up Database Mail for the first time.
3. Move through screen to set up new account. Give it a profile name (you can have multiple emails linked to a profile).
4. Add a new SMTP account and provide mail server details, as shown below.

Specify name, description, and attributes for your SMTP account.

Account name: DDSafe Support DBAs

Description:

Outgoing Mail Server (SMTP)

E-mail address: support@ddsafe.co.uk

Display name: Support DBAs

Reply e-mail: support@ddsafe.co.uk

Server name: mail.ddsafe.co.uk Port number: 2525

☐ This server requires a secure connection (SSL)

SMTP Authentication

☐ Windows Authentication using Database Engine service credentials

☒ Basic authentication

User name: support+ddsafe.co.uk

Password:

Confirm password:

☐ Anonymous authentication

OK Cancel Help

- To test the email account, Expand **Management**, right-click **Database Mail**, and select **Send Test E-Mail...**

Powershell

Installation

The current stable release is PowerShell v1.0. To get the greater benefits of this scripting language, install PowerShell 2.0 CTP3 (Community Technology Preview):-

- Requires .Net Framework 2.0 or higher. Go for 3.5 SP1 at least.
- Uninstall previous release of PowerShell
- If all else fails, try 2.0 CTP2

Uninstalling older PowerShell Versions

You should be able to uninstall using 'Add or Remove Programs' from the Control Panel. Make sure 'Show Updates' is selected. May need to look in the 'Windows XP – Software Updates' section.

However, this is Microsoft, so it may not work. Try the following:-

- Remove Hotfix 926139, 926140, 926141 by running:
 C:\Windows\%NtUninstallKB926139%\spuninst\spuninst.exe.
 C:\Windows\%NtUninstallKB926140%\spuninst\spuninst.exe.
 C:\Windows\%NtUninstallKB926141%\spuninst\spuninst.exe.
 You may not have all these installed.
- The Hotfix removal may break the .Net Framework installation, so reinstall this. In fact, if you get a 'System.Management.automation.dll' error (see Figure 1), you may need to install all old and current .Net Framework installations, including 2.0, 3.0 and 3.5 and their service packs. Then reinstall the latest version.



Figure 1. PowerShell install error

3. If all else fails, try 2.0 CTP2

Getting Started

Set up your user profile. First check if it exists

```
Test-path $profile ($profile is a built-in variable)
```

If false

```
New-item -path $profile -type file -force
```

You can also set up a server wide profile if you are the administrator. Just edit
%windir%\system32\windows\powershell\profile.ps1

Useful Code Snippets

Positional Parameter

Some commands allow us to omit the positional parameter (so of the default parameter). To find positional parameter of a cmdlet, execute:-

```
(Get-Help Get-Process).parameters.parameter
```

Look for the parameter with Position? = 1

Filtering, Sorting, Formatting & Exporting

Filter on processes starting with 's'. The special character '\$_' is substituted with output of get-process, row-by-row.

```
get-process |where-object {$_.Processname -like "s*"}
```

We can sort results from a cmdlet:-

```
Get-Process s*| Sort-Object CPU -desc
```

We can Format the output using these:-

```
Get-Process s*| Select-Object CPU,Id, ProcessName | Format-Table -autosize
```

```
Get-Process| Select-Object CPU,ProcessName | Sort-Object CPU -desc| Format-Table
    @{expression="ProcessName"; width=25; label="Process Name"},
    @{expression="CPU"; width=20; label="CPU Used"}
```

We can export output to CSV format

```
get-process s*|select-object CPU, Processname| Export-Csv
"c:\SCRIPTS\test.txt"
```

Examples

To be able to run scripts>

```
PS>Set-ExecutionPolicy RemoteSigned
```

Some good examples can be found at:

<http://www.simple-talk.com/sql/database-administration/why-this-sql-server-dba-is-learning-powershell/>

Create a file called "C:\AllServers.txt") to hold a list of all your SQL Servers.

Example 1

This example is a typical template for all repetitive processes against servers.

1. loops through the servers in AllServers.txt
2. makes connection
3. creates SQL commandline
4. executes the SQL
5. Formats the output into a table and returns the output.

```
foreach ($svr in get-content "C:\AllServers.txt")
```

```
{
    $con = "server=$svr;database=master;Integrated Security=sspi"
    $cmd = "SELECT '$svr' as ServerName, SERVERPROPERTY('ProductVersion') AS Version,
            SERVERPROPERTY('ProductLevel') as SP"
```

```

$da = new-object System.Data.SqlClient.SqlDataAdapter ($cmd, $con)
$dt = new-object System.Data.DataTable
$da.fill($dt) | out-null
$svr
$dt | Format-Table -autosize
}

```

Example 2

```

function RunCmd($s)
{
    $svr="$s"
    $cn = new-object System.Data.SqlClient.SqlConnection
    "server=$svr;database=master;Integrated Security=sspi"
    $cn.Open()
    $sql = $cn.CreateCommand()
    $svr
    $sql.CommandText = "dbcc freeproccache;"
    $rdr = $sql.ExecuteNonQuery();
}

```

Example 3

Use a back-tick to span multiple lines

```

PS SQLSERVER:\SQL\YM-WINXP\DEFAULT\databases> invoke-sqlcmd -query " backup database
Adventureworks to
>> disk='C:\MSSQL_Data\Backup\Adventureworks_testdump_stripe1.bak', \
>> disk='C:\MSSQL_Data\Backup\Adventureworks_testdump_stripe2.bak', \
>> disk='C:\MSSQL_Data\Backup\Adventureworks_testdump_stripe3.bak', \
>> disk='C:\MSSQL_Data\Backup\Adventureworks_testdump_stripe4.bak' "
>>

```

DDL Triggers (Tracking DB Schema Changes)

DDL triggers can be used to record CREATE, DROP and ALTER statements.

Creating a DDL Trigger

To do this:-

1. Create a table to record actions.
2. Create trigger
3. Enable trigger

```

CREATE TABLE [dbo].[DatabaseLog](
    [DatabaseLogID] [int] IDENTITY(1,1) NOT NULL,
    [PostTime] [datetime] NOT NULL,
    [DatabaseUser] [sysname] NOT NULL,
    [Event] [sysname] NOT NULL,
    [Schema] [sysname] NULL,
    [Object] [sysname] NULL,
    [TSQL] [nvarchar](max) NOT NULL,
    [XmlEvent] [xml] NOT NULL,
    CONSTRAINT [PK_DatabaseLog_DatabaseLogID] PRIMARY KEY NONCLUSTERED
    (
        [DatabaseLogID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

There is a very good example of a 'catch-all' DDL trigger in the AdventureWorks databases:

```

CREATE TRIGGER [ddlDatabaseTriggerLog] ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @data XML;
    DECLARE @schema sysname;
    DECLARE @object sysname;
    DECLARE @eventType sysname;

    SET @data = EVENTDATA();
    SET @eventType = @data.value('/EVENT_INSTANCE/EventType)[1]', 'sysname');
    SET @schema = @data.value('/EVENT_INSTANCE/SchemaName)[1]', 'sysname');
    SET @object = @data.value('/EVENT_INSTANCE/ObjectName)[1]', 'sysname');

```

```

IF @object IS NOT NULL
    PRINT ' ' + @eventType + ' - ' + @schema + '.' + @object;
ELSE
    PRINT ' ' + @eventType + ' - ' + @schema;

IF @eventType IS NULL
    PRINT CONVERT(nvarchar(max), @data);

INSERT [dbo].[DatabaseLog]
(
    [PostTime],
    [DatabaseUser],
    [Event],
    [Schema],
    [Object],
    [TSQL],
    [XmlEvent]
)
VALUES
(
    GETDATE(),
    CONVERT(sysname, CURRENT_USER),
    @eventType,
    CONVERT(sysname, @schema),
    CONVERT(sysname, @object),
    @data.value('/EVENT_INSTANCE/TSQLCommand)[1]', 'nvarchar(max)'),
    @data
);

END;
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Database
trigger to audit all of the DDL changes made to the Adventureworks database.'
, @level0type=N'TRIGGER', @level0name=N'ddlDatabaseTriggerLog'
GO

```

Finally, enable the trigger

```

ENABLE TRIGGER [ddlDatabaseTriggerLog] ON DATABASE
Go

```

DDL event groups

DDL event groups are a short-cut mechanism which groups a number of 'event types' together. Note that the relationship is hierarchical.

	Server Scope	Database Scope
DDL_EVENTS		
DDL_SERVER_LEVEL_EVENTS (CREATE DATABASE, ALTER DATABASE, DROP DATABASE, ALTER_INSTANCE)	x	
DDL_LINKED_SERVER_EVENTS (CREATE_LINKED_SERVER, ALTER_LINKED_SERVER, DROP_LINKED_SERVER)	x	
DDL_LINKED_SERVER_LOGIN_EVENTS (CREATE_LINKED_SERVER_LOGIN, DROP_LINKED_SERVER_LOGIN)	x	
DDL_REMOTE_SERVER_EVENTS (CREATE_REMOTE_SERVER, ALTER_REMOTE_SERVER, DROP_REMOTE_SERVER)	x	
DDL_EXTENDED_PROCEDURE_EVENTS (CREATE_EXTENDED_PROCEDURE, DROP_EXTENDED_PROCEDURE)	x	
DDL_MESSAGE_EVENTS (CREATE_MESSAGE, ALTER_MESSAGE, DROP_MESSAGE)	x	
DDL_ENDPOINT_EVENTS (CREATE_ENDPOINT, ALTER_ENDPOINT, DROP_ENDPOINT)	x	
DDL_SERVER_SECURITY_EVENTS (ADD_ROLE_MEMBER, DROP_ROLE_MEMBER, ADD_SERVER_ROLE_MEMBER, DROP_SERVER_ROLE_MEMBER)	x	
DDL_LOGIN_EVENTS (CREATE_LOGIN, ALTER_LOGIN, DROP_LOGIN)	x	
DDL_GDR_SERVER_EVENTS (GRANT_SERVER, DENY_SERVER, REVOKE_SERVER)	x	
DDL_AUTHORIZATION_SERVER_EVENTS (ALTER_AUTHORIZATION_SERVER)	x	
DDL_DATABASE_LEVEL_EVENTS	x	x
DDL_FULLTEXT_CATALOG_EVENTS (CREATE_FULLTEXT_CATALOG, ALTER_FULLTEXT_CATALOG, DROP_FULLTEXT_CATALOG)	x	x
DDL_DEFAULT_EVENTS (CREATE_DEFAULT, DROP_DEFAULT, BIND_DEFAULT)	x	x
DDL_EXTENDED_PROPERTY_EVENTS (CREATE_EXTENDED_PROPERTY, DROP_EXTENDED_PROPERTY)	x	x
DDL_PLAN_GUIDE_EVENTS (CREATE_PLAN_GUIDE, ALTER_PLAN_GUIDE, DROP_PLAN_GUIDE)	x	x
DDL_RULE_EVENTS (CREATE_RULE, DROP_RULE, BIND_RULE)	x	x
DDL_TABLE_VIEW_EVENTS	x	x
DDL_TABLE_EVENTS (CREATE_TABLE, ALTER_TABLE, DROP_TABLE)	x	x
DDL_VIEW_EVENTS (CREATE_VIEW, ALTER_VIEW, DROP_VIEW)	x	x
DDL_INDEX_EVENTS (CREATE_INDEX, ALTER_INDEX, DROP_INDEX, CREATE_XML_INDEX, CREATE_FULLTEXT_INDEX, ALTER_FULLTEXT_INDEX, DROP_FULLTEXT_INDEX)	x	x
DDL_STATISTICS_EVENTS (CREATE_STATISTICS, UPDATE_STATISTICS, DROP_STATISTICS)	x	x
DDL_SYNONYM_EVENTS (CREATE_SYNONYM, DROP_SYNONYM)	x	x
DDL_FUNCTION_EVENTS (CREATE_FUNCTION, ALTER_FUNCTION, DROP_FUNCTION)	x	x
DDL_PROCEDURE_EVENTS (CREATE_PROCEDURE, ALTER_PROCEDURE, DROP_PROCEDURE)	x	x
DDL_TRIGGER_EVENTS (CREATE_TRIGGER, ALTER_TRIGGER, DROP_TRIGGER)	x	x
DDL_EVENT_NOTIFICATION_EVENTS (CREATE_EVENT_NOTIFICATION, DROP_EVENT_NOTIFICATION)	x	x
DDL_ASSEMBLY_EVENTS (CREATE_ASSEMBLY, ALTER_ASSEMBLY, DROP_ASSEMBLY)	x	x
DDL_TYPE_EVENTS (CREATE_TYPE, DROP_TYPE)	x	x
DDL_DATABASE_SECURITY_EVENTS	x	x
DDL_CERTIFICATE_EVENTS (CREATE_CERTIFICATE, ALTER_CERTIFICATE, DROP_CERTIFICATE)	x	x
DDL_USER_EVENTS (CREATE_USER, ALTER_USER, DROP_USER)	x	x
DDL_ROLE_EVENTS (CREATE_ROLE, ALTER_ROLE, DROP_ROLE)	x	x
DDL_APPLICATION_ROLE_EVENTS (CREATE_APPROLE, ALTER_APPROLE, DROP_APPROLE)	x	x
DDL_SCHEMA_EVENTS (CREATE_SCHEMA, ALTER_SCHEMA, DROP_SCHEMA)	x	x
DDL_GDR_DATABASE_EVENTS (GRANT_DATABASE, DENY_DATABASE, REVOKE_DATABASE)	x	x
DDL_AUTHORIZATION_DATABASE_EVENTS (ALTER_AUTHORIZATION_DATABASE)	x	x
DDL_SSB_EVENTS	x	x
DDL_MESSAGE_TYPE_EVENTS (CREATE_MSGTYPE, ALTER_MSGTYPE, DROP_MSGTYPE)	x	x
DDL_CONTRACT_EVENTS (CREATE_CONTRACT, DROP_CONTRACT)	x	x
DDL_QUEUE_EVENTS (CREATE_QUEUE, ALTER_QUEUE, DROP_QUEUE)	x	x
DDL_SERVICE_EVENTS (CREATE_SERVICE, ALTER_SERVICE, DROP_SERVICE)	x	x
DDL_ROUTE_EVENTS (CREATE_ROUTE, ALTER_ROUTE, DROP_ROUTE)	x	x
DDL_REMOTE_SERVICE_BINDING_EVENTS (CREATE_REMOTE_SERVICE_BINDING, ALTER_REMOTE_SERVICE_BINDING, DROP_REMOTE_SERVICE_BINDING)	x	x
DDL_XML_SCHEMA_COLLECTION_EVENTS (CREATE_XML_SCHEMA_COLLECTION, ALTER_XML_SCHEMA_COLLECTION, DROP_XML_SCHEMA_COLLECTION)	x	x
DDL_PARTITION_EVENTS	x	x
DDL_PARTITION_FUNCTION_EVENTS (CREATE_PARTITION_FUNCTION, ALTER_PARTITION_FUNCTION, DROP_PARTITION_FUNCTION)	x	x
DDL_PARTITION_SCHEME_EVENTS (CREATE_PARTITION_SCHEME, ALTER_PARTITION_SCHEME, DROP_PARTITION_SCHEME)	x	x

Useful performance Tuning queries

There is a very good article on P&T entitled “[Troubleshooting Performance Problems in SQL Server 2005](#)”

shows long running transactions

```
select * from sys.dm_exec_query_stats order by total_worker_time
```

shows syntax of long running transaction


```
select * from sys.dm_exec_sql_text(<sql_handle>)
```

Top 50 Users of CPU

```
select top 50
    sum(qs.total_worker_time) as total_cpu_time,
    sum(qs.execution_count) as total_execution_count,
    count(*) as number_of_statements,
    qs.plan_handle
from
    sys.dm_exec_query_stats qs
group by qs.plan_handle
order by sum(qs.total_worker_time) desc
```

Useful SQL Commands

Find the collation of a database

```
Select databasepropertyex('db_name', 'collation')
```

Collation Set Commands

```
--Find the collation of the current server
select SERVERPROPERTY ('collation')
--Find the collation of the current database
select convert(sysname,DatabasePropertyEx(db_name(),'Collation'))
--Find collation of all columns in a table
select name, collation from syscolumns where
[id]=object_id('MyTable')
--Find all collation available in SQL Server
select * from ::fn_helpcollations()

--other good tests
select SERVERPROPERTY ('SqlCharSet')
select SERVERPROPERTY (' SqlCharSetName ')
select SERVERPROPERTY (' SqlSortOrder ')
select SERVERPROPERTY('ProductLevel')
```

Who are the Sysadmins

```
SELECT      rol.name, mem.name, mem.type_desc
FROM      sys.server_role_members AS srm
INNER JOIN sys.server_principals AS rol ON rol.principal_id = srm.role_principal_id
INNER JOIN sys.server_principals AS mem ON mem.principal_id = srm.member_principal_id
WHERE rol.name = 'sysadmin'
```

Monitoring

Disk space used by MSSQL Server

1. create table in 'reporting database' & sp__spacegrowth in msdb database. Set to run once a day.
2. To see results execute sp__spacegrowth_db.

Create reporting table and SP

```

use reporting
go
drop TABLE [dbo].[DatabaseFileSizes]
go
CREATE TABLE [dbo].[DatabaseFileSizes](
    [SampleDateTime] datetime,
    [DatabaseName] [nvarchar](128) ,
    [DatabaseFileName] [sysname] ,
    [fileid] [smallint] NULL,
    [drive] [nvarchar](1),
    [filename] [nvarchar](260) ,
    [filegroup] [nvarchar](128) ,
    [size_KB] bigint ,
    [maxsize] [nvarchar](128) ,
    [growth] [nvarchar](30) ,
    [usage] [nvarchar](30)
) ON [PRIMARY]
go
use msdb
go
drop Procedure sp__spacegrowth
go
Create Procedure sp__spacegrowth
as
INSERT INTO reporting..DatabaseFileSizes
select  getdate(),
        databasename=db_name(dbid), name, fileid, drive=left(filename, 1), filename,
        filegroup=filegroup_name(groupid),
        'size'= convert(bigint, size) * 8,
        'maxsize'=(case maxsize when -1 then N'Unlimited'
                    else
                        convert(nvarchar(15), convert(bigint, maxsize) * 8) + N' KB'
                    end),
        'growth'=(case status & 0x10000
                    when 0x100000 then
                        convert(nvarchar(15), growth) + N'%'
                    else
                        convert(nvarchar(15), convert(bigint, growth) * 8) + N' KB'
                    end),
        'usage'=(case status & 0x40 when 0x40 then 'log only' else 'data only' end)
from master..sysaltfiles
order by dbid
go
create procedure sp__spacegrowth_db @dbname varchar(40), @days int
as
select * from reporting..DatabaseFileSizes
where Databasename=@dbname
and SampleDateTime < DATEADD(day, @days, getdate())
order by fileid
go

```

Quick way to find space used by each table in a database

```
sp_msforeachtable 'sp_spaceused "?"'
```

Simple script using cursors to execute DBCC UPDATEUSAGE

```

declare @dbname sysname

--Cursor to get the list of database names
declare cur cursor for
    select name from sysdatabases

OPEN cur
FETCH NEXT FROM cur INTO @dbname /*get the first DB name */

--Start loop
WHILE @@FETCH_STATUS = 0
BEGIN
    print '#####'
    print 'Correct inaccuracies in the sysindexes table for: ' + @dbname
    print '#####'
    DBCC UPDATEUSAGE (@dbname) /* Do the real work */
    FETCH NEXT FROM cur INTO @dbname
END
--Finally close the cursor to stop memory leaks
CLOSE Cur

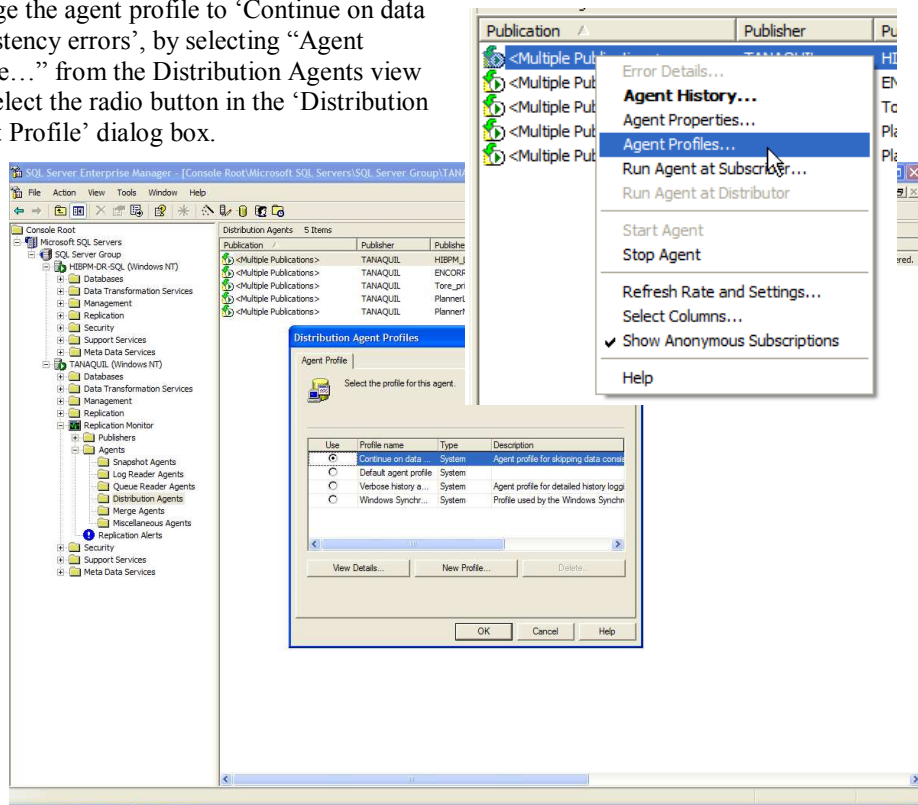
```


Replication Stuff

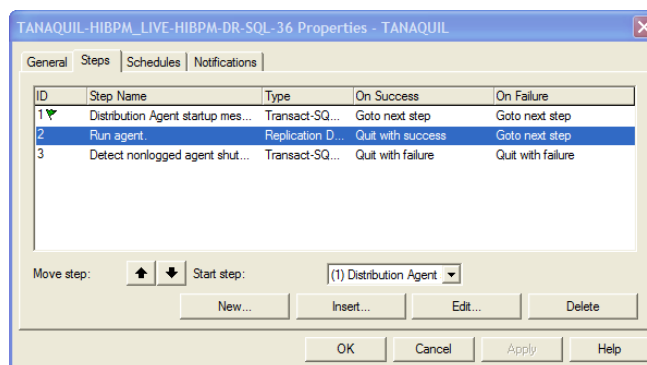
Duplicate key errors (error 20598) on resuming replication to a restored replicate database from the primary

To fix this in Sybase we could set 'auto correction' on. In Microsoft SQL Server we need to either

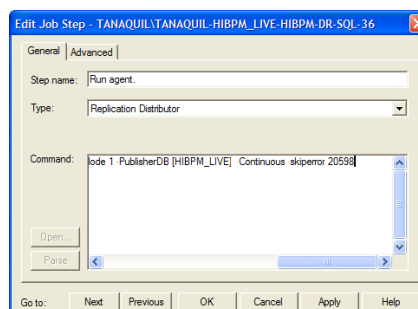
1. Change the agent profile to 'Continue on data consistency errors', by selecting "Agent Profile..." from the Distribution Agents view and select the radio button in the 'Distribution Agent Profile' dialog box.



2. Change the agent by adding the -skiperror switch to the command line. Select the 'Agent Properties...' from the menu and in the properties dialog and 'Steps' tab, double-click on the 'Run agent'



Now scroll to the far right of the Command line and add
-skiperror <error number>



Restart distribution agent.

Procedure to Drop Push Subscription

On the primary SQL server,

```
use hibpm_live
go
exec sp_dropsubscription @publication = N'hibpm_live', @article = N'all',
@subscriber = N'all', @destination_db = N'all'
go
--Now remove any tran log markers
EXEC sp_repldone @xactid = NULL, @xact_segno = NULL, @numtrans = 0, @time = 0,
@reset = 1
go
--truncate the log
backup log HIBPM_LIVE with truncate_only
go
--Shrink the log file
use HIBPM_LIVE
go
DBCC SHRINKFILE (AWD_LFT_DEV_Log,250)
go
```

Procedure to rebuild replication on MS SQL Server

Steps

1. Create the publication (if one does not already exist) from the primary or distributor
 - Transactional replication
 - Select servers compatibility for 2000 and 7
 - Publish All tables
 - Everything else is default settings
2. Drop then Create a push subscription from this publication
 - Select the destination server and database
 - Select "No the subscriber already has schema and data"
 - Everything else is default values
3. Stop synchronising the subscription
4. Backup and Restore the database from the primary to replicate
5. Start synchronising the subscription by stopping and restarting the Log Reader
 - If you get errors about missing stored procs go to Note A
 - If you get errors about duplicate rows go to Note B

Note A

This error will be reported by the distribution agent and if you click on the agent it will show you the exact error message,

- Take a look at this and note which stored proc it is missing.
- Run select * from sysarticles from the primary db and note the artid which corresponds to the missing object.
- Run the following procs on the primary db:
 1. exec sp_scriptinsproc 'artid'
 2. exec sp_scriptdelproc 'artid'
 3. exec sp_scriptmappedupproc 'artid'
- Take the output of these 3 procedures and apply them to the replicant db. If you have problems getting very long definitions out of the database, try using 'osql' from the command line. E.g. *osql -E -STANAQUIL -w2048 -i qry.sql -o qry.out*
- Restart the distribution agent and it should be fine if not you might get another missing proc in which case repeat the above steps.

Another way to do this is:-

--Fix missing custom procs.

--get the publication name

-- This stored procedure is executed at the Publisher on the publication database. However, this **can truncate the output** for tables with lots of columns.

sp_helppublication

go

-- extract all replication SP's

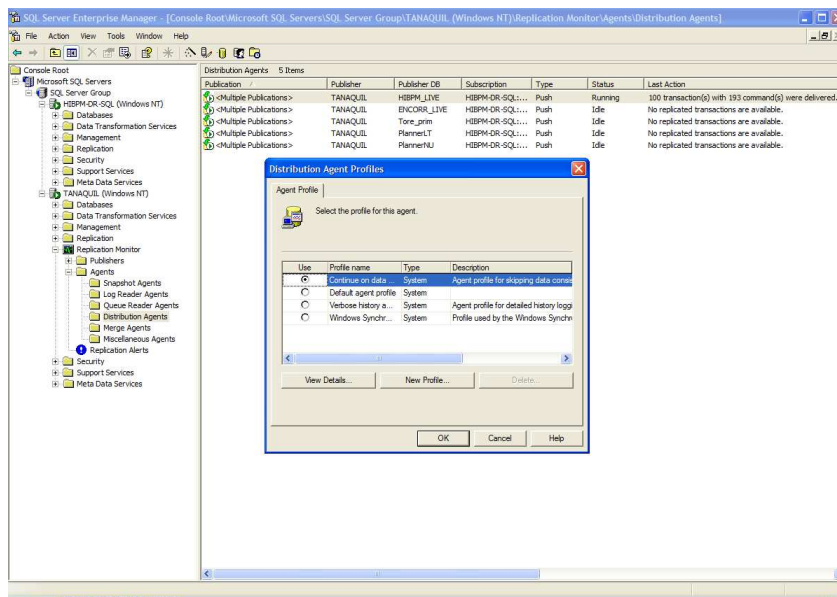
sp_scriptpublicationcustomprocs 'HIBPM_LIVE'

-- now just search output for missing SP and put on subscriber (target) server

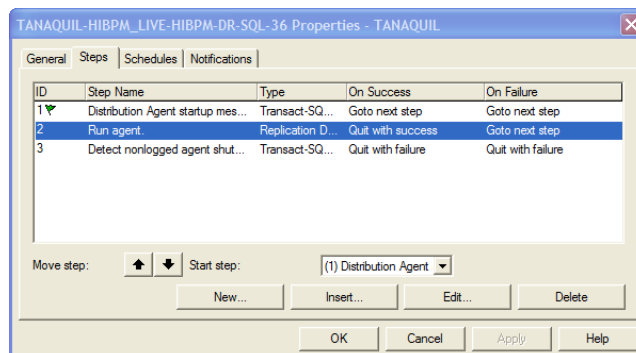
Note B

To fix these duplicate errors (error 20598), or indeed any similar errors you do the following:

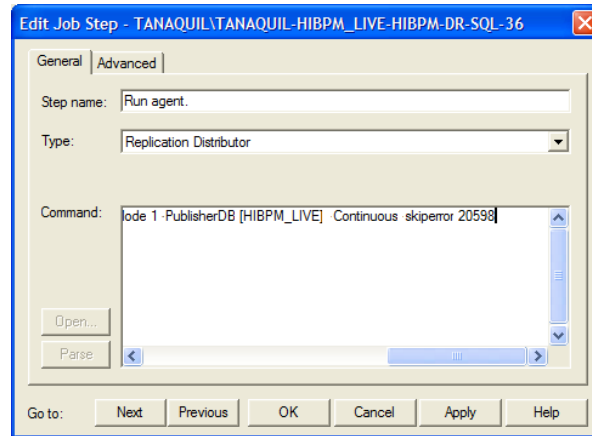
1. Change the agent profile to 'Continue on data consistency errors', by selecting "Agent Profile..." from the Distribution Agents view and select the radio button in the 'Distribution Agent Profile' dialog box.



2. Additionally you may want to change the agent by adding the --skiperror switch to the command line. Select the 'Agent Properties...' from the menu and in the properties dialog and 'Steps' tab, double-click on the 'Run agent'



Now scroll to the far right of the Command line and add
--skiperror <error number>



3. Restart the distribution agent and you will see it reporting skipped errors.
4. When it stops reporting this and instead just reports that it is delivering transactions you can stop the distribution agent, again right click the distribution agent and put it back to Default agent profile and also take out the skiperror from the command line if you put it in and restart the agent.

Subscription Deactivation and Expiration

Refer to http://doc.ddart.net/mssql/sql2000/html/replsql/replmon_92ge.htm for full details.

When a subscription is not synchronized, the subscription will be marked deactivated by the Distribution Cleanup Agent that runs on the Distributor.

To stop this from happening, disable the Expired Subscription Cleanup Agent (see screenshot below)

