

Chancery SMS 7.5.0 Database Split

Microsoft SQL Server replication.....	1
Transactional replication.....	2
Preparing to set up replication.....	3
Setting up replication.....	4
Quick Reference	11

Revision A, November 30, 2010, © 2009 Pearson Education, Inc. or its affiliates. All rights reserved. This edition applies to Release 7.5 of the Chancery SMS® software product and to all subsequent releases and modifications until otherwise indicated in new editions or updates.

To improve performance, large districts that have a high load on their database may want to implement a *database split*. The Chancery SMS database split is designed to distribute database processing across multiple servers, separating the *transactional activity* associated with basic operations, such as attendance, grading, scheduling, and so on, from the reporting or analytical activity associated with large or complex reports. **Note:** Districts interested in implementing a database split should first contact Pearson to determine if this is a viable solution for their district.

Microsoft SQL Server replication

Replication is the process of copying and distributing data and database objects from one database to another and then synchronizing between databases for consistency. The Chancery SMS database split uses Microsoft® SQL Server® *transactional replication* to mirror the data in a database on one server (the *Publisher*) to another database on a separate server (the *Subscriber*). Replication occurs in near real-time; the delay in updating data on the Subscriber server is insignificant.

The Microsoft SQL Server replication model is composed of the following components:

Publisher – the server and database that sends data to another database. In the Chancery SMS database split, this is the Chancery SMS database server.

Subscriber – the server and database that receives data from the Publisher. In the Chancery SMS database split, this is the Chancery SMS reporting server.

Distributor – the server that manages the flow of data and hosts the Distribution database. The Distribution database stores snapshot jobs and transactions waiting to be distributed to Subscribers. The Chancery SMS database split uses a local Distributor; that is, a server is configured to be both the Publisher and the Distributor.

Articles and Publications – articles are the basic units of replication, such as entire tables, specific columns or rows, stored procedures, view definitions, and so on. Publications are collections of one or more articles from one database grouped together to logically relate sets of data and database objects that are to be replicated together.

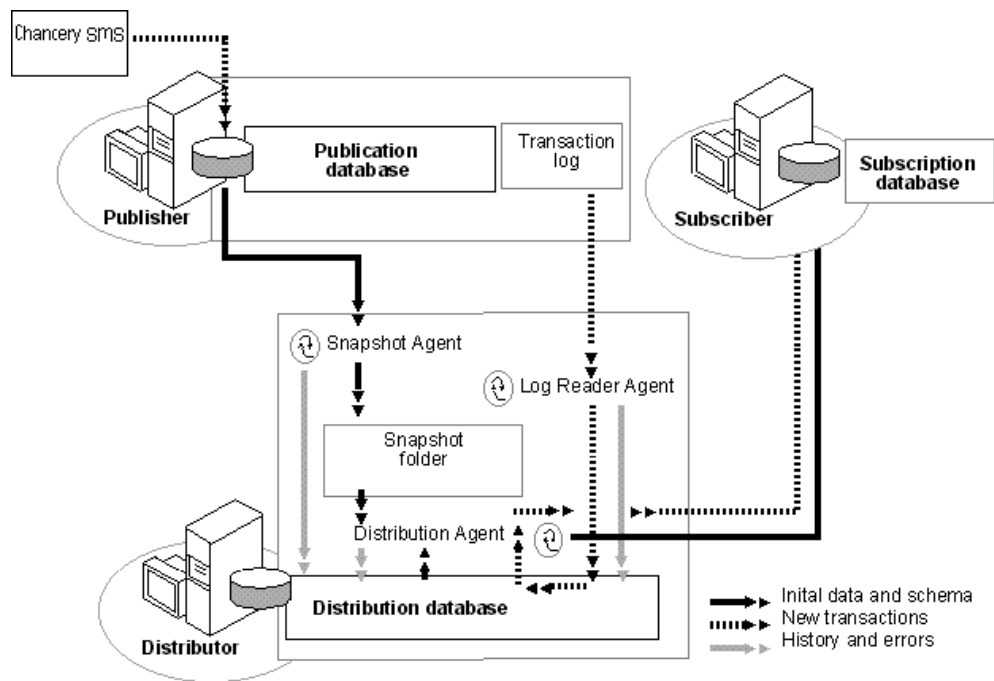
Subscriptions – requests for a copy of data or database objects to be replicated. Subscriptions define what publications the Subscriber will receive, where, and when. The Chancery SMS database split uses *push subscriptions* in which the Publisher sends transactions to the database on the subscribing server as they occur.

Once the replication database is set up, Chancery SMS can be configured to execute all reports against the Subscriber rather than the Publisher. Data additions or changes to the database are made on the Publisher and replicated to the Subscriber.

Warning: If you use clustering, do not use the passive node in your cluster as the Subscriber server. This configuration could negatively impact a failover for the existing production server.

Transactional replication

With transactional replication, an initial snapshot of data is sent to the Subscriber and then, when data modifications are made at the Publisher, SQL Server captures the changes and stores INSERT, UPDATE, and DELETE statements in the Distribution database. The changes are then sent to Subscribers from the Distribution database and applied in the same order.



Transactional replication agents

Transactional replication uses the following agents to carry out the tasks associated with copying and distributing data:

SQL Server Agent – hosts and schedules the agents used in replication, and provides an easy way to run replication agents. It also controls and monitors several other operations outside of replication, including monitoring the SQL Server Agent service, maintaining error logs, running jobs, and starting other processes.

Snapshot Agent – prepares schema and initial data files of published tables and stored procedures, stores the snapshot files, and records information about synchronization in the Distribution database. The Snapshot Agent typically runs under SQL Server Agent at the Distributor and can be administered using SQL Server Enterprise Manager.

Log Reader Agent – moves transactions marked for replication from the transaction log on the Publisher to the Distribution database.

Distribution Agent – moves the snapshot jobs and transactions from the Distribution database to Subscribers. The Distribution Agent can be administered using SQL Server Enterprise Manager.

Preparing to set up replication

The Chancery SMS database split requires a completely separate Subscriber server, which means that you must have another piece of hardware fully dedicated as a Subscriber. Chancery SMS supports database split in SQL Server 2005 or SQL Server 2008, and the SQL Server license for the Subscriber must be identical to the license used for the Publisher.

Before setting up transactional replication, set up the Publisher as follows:

- Make sure you are using SQL Server 2005 or 2008.
- If the Publisher is on SQL 2005 and the Subscriber is on SQL 2008 you cannot add the Subscriber to the Publisher. Upgrade the Publisher to SQL 2008 so that you have both servers on the same version.
- Make sure all report packages have been uploaded to Chancery SMS.
- Make sure the Publisher has adequate rights to access the Subscriber. SQL Server and SQL Server Agent must log into a domain account with administrative rights on the SQL servers and “Log in as a service” rights.
- Make sure you have the rights associated with members of the sysadmin server role.
- Start SQLServerAgent service and MSSQLServer service.
- Allocate adequate disk space for the Distribution database. The Distribution database must store all transactions until they can be applied to the Subscriber, including bulk transactions. Allow enough free disk space to hold a copy of the main database.
- Allocate adequate disk space for the Publisher and Subscriber databases. Since the Subscriber database is a mirror of the Publisher database, you must have enough storage space on the Subscriber server to accommodate the projected size of your Publisher database. However, since the Subscriber database uses simple recovery logging only, you do not need to allow space for transaction logs.
- Add the Subscriber server to the registered server list in Enterprise Manager on the Publisher server.
- On the Publisher server open Enterprise Manager and make sure the local server is registered by name, not as “(local)”.

Setting up replication

Setting up replication involves the following steps:

1. Stop Chancery Task Managers on production servers ([p. 4](#))
2. Run View Generator ([p. 4](#))
3. Stop Chancery Task Managers on the Publisher server ([p. 5](#))
4. Give the database owner permission to monitor SQL jobs ([p. 5](#))
5. Add identity columns and primary keys to the Publisher database ([p. 5](#))
6. Add remaining primary keys to the Publisher database ([p. 5](#))
7. Increase the field size for replication ([p. 5](#))
8. Create the database directory structure on Subscriber database ([p. 6](#))
9. Back up the Publisher database ([p. 7](#))
10. Restore the backup to the Subscriber database ([p. 7](#))
11. Set the database owner on the restored Subscriber database ([p. 7](#))
12. Remove foreign keys from the restored database tables ([p. 7](#))
13. Remove triggers from the restored database tables ([p. 8](#))
14. Set simple recovery mode on the restored database ([p. 8](#))
15. Configure the Publisher and Subscriber servers for replication ([p. 8](#))
16. Add a new publication on the Publisher server ([p. 9](#))
17. Create a Snapshot Agent for the publication ([p. 9](#))
18. Create a current list of articles to publish ([p. 9](#))
19. Add articles to the publication ([p. 9](#))
20. Activate the publication ([p. 9](#))
21. Create a subscription to the publication ([p. 9](#))
22. Set the subscription initialization ([p. 9](#))
23. Start the Snapshot Agent ([p. 10](#))
24. Wait for Snapshot Agent to complete ([p. 10](#))
25. Regenerate statistics on the Subscriber database ([p. 10](#))
26. Enable Chancery SMS to use the Subscriber server ([p. 11](#))
27. Run the view generator ([p. 11](#))

1. Stop Chancery Task Managers on production servers

You must stop all Chancery SMS Task Managers on Chancery SMS servers that are configured as web server, application server, reporting server, and scheduling server. At this time, DO NOT stop the Chancery SMS Task Manager on the Chancery SMS database server.

To stop Chancery SMS Task Managers:

- 1 On each production server, select **Start > Control Panel > Administrator Tools > Services**.
- 2 In the Services dialog, right-click any Chancery SMS Task Manager service that is currently started and click **Stop**.

2. Run View Generator

If a Chancery SMS HotFix or version update has recently been run on the production servers, verify that the View Generator has successfully run.

To check the status of View Generator:

- On each production database server, run the following script:

```
SELECT * FROM CSL_SMS_REPORT_VIEW_GEN_STS
```

 - ④ The `TIMESTAMP` should be at least as recent as the most recent database update and the field `IS_READY` should be set to 1. If either of these values is incorrect, run the View Generator.

To run the View Generator:

- On each production database, run the following script:

```
INSERT_VIEW_TASK
```

3. Stop Chancery Task Managers on the Publisher server

After View Generator successfully runs, stop all Chancery SMS Task Managers on the Publisher server.

To stop Chancery SMS Task Managers:

- 1 On the Publisher server, select **Start > Control Panel > Administrator Tools > Services**.
- 2 In the Services dialog, right-click any Chancery SMS Task Manager service that is currently started and click **Stop**.

4. Give the database owner permission to monitor SQL jobs

In the following script, "csl" represents the database username specified when the Chancery SMS database was installed.

- On the Publisher database, run the following script:

```
use master
GRANT EXECUTE ON master.dbo.xp_sqlagent_enum_jobs TO csl
```

5. Add identity columns and primary keys to the Publisher database

If any tables on the Publisher database do not have both identity columns and primary keys, you must add them. This process may take up to 5 minutes. `DAT` and other tables that are not replicated are not altered. In the following script, "CSL_SMS" represents the database name.

- On the Publisher database, run:

```
use CSL_SMS
EXEC REPLICATION_ADD_ID_AND_PK
```

6. Add remaining primary keys to the Publisher database

Add primary keys to all tables in the Publisher database that still do not have primary keys. This process may take up to 5 minutes. In the following script, "CSL_SMS" represents the database name.

- On the Publisher database, run:

```
use CSL_SMS
EXEC REPLICATION_ADD_PK
```

7. Increase the field size for replication

On the Publisher database, increase the maximum field size allowed for replication for report templates. In the following script, "CSL_SMS" represents the database name.

- On the Publisher database, run:


```
use CSL_SMS
EXEC sp_configure 'max text repl size',1000000
RECONFIGURE WITH OVERRIDE
```

8. Create the database directory structure on Subscriber database

Before you can restore the database to the Subscriber, set up file locations on the Subscriber where you can restore the database. If Chancery SMS has never been installed on the Subscriber, create the locations by setting up the same directory structure on the Subscriber as is used on the Publisher. For example, if you set up the Publisher to use one drive for data and another drive for logs, you might have the following directory structure (in this example, "SMS" represents the database name, which is derived from the physical database name):

```
F:\CSL\SMS\aspdata
F:\CSL\SMS\aspindex
F:\CSL\SMS\oddata
F:\CSL\SMS\odindex
F:\CSL\SMS\PR_Data
F:\CSL\SMS\PR_Index
F:\CSL\SMS\PR_SCG_D
F:\CSL\SMS\PR_SCG_I
G:\CSL\SMS\log
```

The directory structure may be different for each installation. If the directory structure on the Subscriber is not the same as on the Publisher (for example, the Subscriber uses different drive letters), manually adjust the file locations when you restore the database to the Subscriber.

To check the directory structure on the Publisher:

- 1 On the Publisher, open Enterprise Manager.
- 2 In Enterprise Manager, right-click the Chancery SMS database and select **Properties**.
- 3 Click the **Data Files** tab.
- 4 From the **Locations** column, record each unique drive\folder combination.
- 5 Click the **Transaction Log** tab.
- 6 From the **Locations** column, record each unique drive\folder combination.
- 7 Click **Cancel**.

To recreate the directory structure on the Subscriber:

- On the Subscriber, enter DOS commands to recreate the directory structure. For example, to recreate the directory structure in the example above, enter the following DOS commands:

```
F:
cd\
md csl
cd csl
md sms
cd sms
md aspindex
md aspdata
md odindex
md oddata
md PR_Data
```

```

md PR_Index
md PR_SCG_D
md PR_SCG_I
G:
cd\
md csl
cd csl
md sms
cd sms
md log

```

9. Back up the Publisher database

If the database file structure is identical on the Publisher and Subscriber, use the following script to back up the Publisher database. “CSL_SMS” represents the database name, “CSL_SMS_DA” represents the DA database name, and “s-vadb3” represents the Subscriber server name. If the database file structure on the Subscriber is different (for example, it uses different drives or folders), manually restore the database and adjust the file structure.

- On the Publisher database, run:


```

Backup Database CSL_SMS
To Disk='\\s-vadb3\f$\csl\sms_0810.bak'
With Init

Backup Database CSL_SMS_DA
To Disk='\\s-vadb3\f$\csl\SMS_0804_DA.bak'
With Init

```

10. Restore the backup to the Subscriber database

In the following script, “CSL_SMS” represents the database name and “CSL_SMS_DA” represents the DA database name.

- On the Subscriber server, run:


```

use master
Restore Database CSL_SMS
From Disk='f:\csl\sms_0810.bak'
With Replace

Restore Database CSL_SMS_DA
From Disk='f:\csl\SMS_0804_DA.bak'
With Replace

```

11. Set the database owner on the restored Subscriber database

In the following script, “CSL_SMS” represents the database name.

- On the Subscriber database, run:


```

use CSL_SMS
GO
sp_changedbowner 'csl'

```

12. Remove foreign keys from the restored database tables

This process may take up to 10 minutes. In the following script, “CSL_SMS” represents the database name.

- On the Subscriber database, run the following script:

```
USE CSL_SMS
EXEC REPLICATION_DROP_FOREIGN_KEYS
```

13. Remove triggers from the restored database tables

This process may take up to 10 minutes. In the following script, “CSL_SMS” represents the database name.

- On the Subscriber database, run:

```
USE CSL_SMS
EXEC REPLICATION_DROP_ALL_TRIGGERS
```

14. Set simple recovery mode on the restored database

In the following script, “CSL_SMS” represents the database name.

- On the Subscriber database, run:

```
Alter database CSL_SMS
set recovery simple
```

15. Configure the Publisher and Subscriber servers for replication

If you want to create the Distribution database on the same server as the Publisher, use the following script to configure the servers for replication.

In this script, “CSL_SMS” represents the database name and “S-VADB3” represents the Subscriber server name. The snapshot folder, server data directory, and server log directory must exist on the Publisher before you run this script.

- On the Publisher server, run:

```
use csl_sms
EXEC REPLICATION_INITIALIZE_PUBLISHER
'CSL_SMS' ,
'sa' , -- sa password for Publisher server
'S-VADB3' , -- Replace with the name of the
-- Subscriber server
'sa' , -- sa password for Subscriber
-- server
'sa' , -- Distributor_admin password
-- (system created account)
'H:\SQL_Temp' , -- Snapshot folder
-- Must already exist
'F:\DistributionData' , -- Distribution database data
-- location -
-- Optional - defaults to SQL
-- server data directory
-- Must already exist
'G:\DistributionLogs' -- Distribution database log
-- location -
-- Optional - defaults to SQL
-- server log directory
-- Must already exist
```


16. Add a new publication on the Publisher server

In the following script, “CSL_SMS” represents the database name.

- On the Publisher server, run:
Use CSL_SMS
EXEC REPLICATION_ADD_PUBLICATION

17. Create a Snapshot Agent for the publication

In the following script, “CSL_SMS” represents the database name.

- On the Publisher server, run:
Use CSL_SMS
EXEC REPLICATION_ADD_PUBLICATION_SNAPSHOT_AGENT

18. Create a current list of articles to publish

In the following script, “CSL_SMS” represents the database name.

- On the Publisher server, run:
Use CSL_SMS
EXEC REPLICATION_CREATE_PUBLISH_LIST

19. Add articles to the publication

After you create the articles in step 18, you must add them to the publication. This process may take up to 5 minutes. In the following script, “CSL_SMS” represents the database name.

- On the Publisher database, run:
Use CSL_SMS
EXEC REPLICATION_ADD_ARTICLES_TO_PUBLICATION

20. Activate the publication

In the following script, “CSL_SMS” represents the database name.

- On the Publisher database, run:
Use CSL_SMS
EXEC REPLICATION_SET_PUBLICATION_ACTIVE

21. Create a subscription to the publication

In the following script, “S-VADB3” represents the Subscriber server and “CSL_SMS” represents the Subscriber database name. **Warning:** The Subscriber server name must be in UPPER CASE.

This process may take up to 5 minutes.

- On the Publisher database, run:
EXEC REPLICATION_CREATE_SUBSCRIPTION

'S-VADB3', -- Subscriber server name
'CSL_SMS' -- Subscriber database name

22. Set the subscription initialization

Set the subscription to initialize as soon as the snapshot is complete.

- On the Publisher database, run:
use CSL_SMS
EXEC REPLICATION_SET_SUBSCRIPTION_INITIALIZATION

23. Start the Snapshot Agent

- On the Publisher database, run:

```
use CSL_SMS
EXEC REPLICATION_START_SNAPSHOT_AGENT
```

24. Wait for Snapshot Agent to complete

This procedure can take 1 hour or more depending on the size of the database.

- On the Publisher database, run:

```
use CSL_SMS
EXEC REPLICATION_WAITFOR_SNAPSHOT_AGENT 120
```

When this procedure completes, the snapshot will have been transferred to the Subscriber. Verify that the replication components have been created and that the system has been replicated and is ready for use.

To verify the replication components:

- 1 On the Publisher, expand **Replication > Local Publications**. Then expand the node for the publication created in step 16. Add a new publication on the Publisher server (p. 9).
- 2 Verify that a subscription is created with the format [*SubscribingServer*].
- 3 Right-click the subscription and select **Launch Replication Monitor**.
 - ④ On the Replication Monitor, the status should show "Ok".
- 4 Click **Subscription Watchlist**.
- 5 Right-click the subscription and select **View Details**.
- 6 Click the **Distributor to Subscriber History** tab.
 - ④ When you receive an Action message that states, "Delivered Snapshot from <Publisher>...", the Subscriber is ready for use.
 - ④ To monitor the replication status, you can click the **Undistributed Commands** tab. Before replication is complete, the number of undistributed commands will be numerous. When the snapshot is processed and the Subscriber is ready, the undistributed commands will drop to zero.

25. Regenerate statistics on the Subscriber database

Once the snapshot is applied, regenerate all statistics on the Subscriber database. This process may take up to 15 minutes.

- On the Subscriber database, run:

```
DECLARE @table_name varchar(1000),
        @schema_name varchar (255),
        @sql nvarchar(4000)
declare c1 cursor for
select s.name,t.name from sys.tables t
join sys.schemas s on s.schema_id = t.schema_id
open c1
fetch next from c1 into @schema_name, @table_name
while @@Fetch_Status = 0
begin
    Select @sql = 'UPDATE STATISTICS '+ @schema_name + '.'
+ @table_name +' WITH FULLSCAN'
    --print @sql
    exec sp_executesql @sql
    fetch next from c1 into @schema_name, @table_name
```

```
end
close c1
deallocate c1
```

26. Enable Chancery SMS to use the Subscriber server

To enable Chancery SMS to use the Subscriber server to run reports, complete the following procedure on each web server, application server, reporting server, and scheduling server. You can use this procedure to turn *split server mode* on or off at any time.

To enable or disable split server mode on Chancery SMS:

- 1 Open the following file:
C:\inetpub\WWWRoot\ChancerySMS\SqlDataProvider.Config
- 2 Make the following changes to the file:
 - a In <reportServerConnectionString> change the server name to the name of the Subscriber server.
 - b To enable split server mode, set <enableReplication> to "true".
OR, to disable split server mode, set <enable Replication> to "false".
- 3 Execute IISReset.
- 4 Stop and restart the Task Manager service.

27. Run the view generator

Run the View Generator again to create correct views on the Subscriber server.

- On the Publisher database, run:
EXEC INSERT_VIEW_TASK

Quick Reference

You can use the following script as a quick way to enter the scripts in Query Analyzer.

- 1 Copy the script to the clipboard.
- 2 Paste the script into Query Analyzer.
- 3 Replace the following variables with appropriate values:
S-VADB3 = the Subscriber server name
CSL_SMS = the reporting database name
CSL_SMS_DA = the DA database name
sa = The Publisher server password, Subscriber server password, or Distributor admin password, as appropriate.
- 4 Execute the scripts one command at a time.

Scripts

```
-- Stop all Chancery Task Managers connected to the Publisher
-- On Publisher database:
use master
GRANT EXECUTE ON master.dbo.xp_sqlagent_enum_jobs TO csl
```

```

-- On Publisher database:
use CSL_SMS
EXEC REPLICATION_ADD_ID_AND_PK

-- On Publisher database:
use CSL_SMS
EXEC REPLICATION_ADD_PK

-- On Publisher database:
use CSL_SMS
EXEC sp_configure 'max text repl size',1000000
RECONFIGURE WITH OVERRIDE

-- Create directory structure on Subscriber

-- On Publisher database:
Backup Database CSL_SMS
To Disk='\\s-vadb3\f$\csl\sms_0810.bak'
With Init

Backup Database CSL_SMS_DA
To Disk='\\s-vadb3\f$\csl\SMS_0804_DA.bak'
With Init --

-- On Subscriber server:
use master
Restore Database CSL_SMS
From Disk='f:\csl\sms_0810.bak'
With Replace

Restore Database CSL_SMS_DA
From Disk='f:\csl\SMS_0804_DA.bak'
With Replace

-- On the Subscriber database:
use CSL_SMS
GO
sp_changedbowner 'csl'

-- On the Subscriber database:
USE CSL_SMS
EXEC REPLICATION_DROP_FOREIGN_KEYS

-- On the Subscriber database:
USE CSL_SMS
EXEC REPLICATION_DROP_ALL_TRIGGERS

-- On the Subscriber database:
Alter database CSL_SMS
set recovery simple

-- On Publisher server:
use csl_sms
EXEC REPLICATION_INITIALIZE_PUBLISHER
'CSL_SMS',
'sa', -- sa password for Publisher server
'S-VADB3', -- Replace with the name of the
Subscriber server
'sa', -- sa password for Subscriber

```

```

-- server
'sa', -- Distributor_admin password
-- (system created account)
'H:\SQL_Temp', -- Snapshot folder
-- Must already exist
'F:\DistributionData', -- Distribution database data
-- location -
-- Optional - defaults to SQL
-- server data directory
-- Must already exist
'G:\DistributionLogs' -- Distribution database log
-- location -
-- Optional - defaults to SQL
-- server log directory
-- Must already exist

Use CSL_SMS
EXEC REPLICATION_ADD_PUBLICATION

Use CSL_SMS
EXEC REPLICATION_ADD_PUBLICATION_SNAPSHOT_AGENT

Use CSL_SMS
EXEC REPLICATION_CREATE_PUBLISH_LIST

Use CSL_SMS
EXEC REPLICATION_ADD_ARTICLES_TO_PUBLICATION

Use CSL_SMS
EXEC REPLICATION_SET_PUBLICATION_ACTIVE

Use CSL_SMS
EXEC REPLICATION_CREATE_SUBSCRIPTION
'S-VADB3,'-- Subscriber server name
'CSL_SMS -- Subscriber database name

Use CSL_SMS
EXEC REPLICATION_SET_SUBSCRIPTION_INITIALIZATION

Use CSL_SMS
EXEC REPLICATION_START_SNAPSHOT_AGENT
-- Wait for snapshot to finish applying

Use CSL_SMS
EXEC REPLICATION_WAITFOR_SNAPSHOT_AGENT 120
-- On the Subscriber database:
DECLARE @table_name varchar(1000),
        @schema_name varchar (255),
        @sql nvarchar(4000)
declare c1 cursor for
select s.name,t.name from sys.tables t
join sys.schemas s on s.schema_id = t.schema_id
open c1
fetch next from c1 into @schema_name, @table_name
while @@Fetch_Status = 0
begin
    Select @sql = 'UPDATE STATISTICS '+ @schema_name + '.'
+ @table_name + ' WITH FULLSCAN'
    --print @sql
    exec sp_executesql @sql

```

```
        fetch next from c1 into @schema_name, @table_name
    end
    close c1
    deallocate c1
```