

Merge Replication 101

SQL Saturday #389

April 11, 2015

Diana Dee

DrDianaDee@verizon.net

Consultant

TribalSQL Author

Adjunct Instructor, Kaplan University

Contents

- SQL Server Replication Overview
- Main Types of SQL Server Replication
- Merge Replication Overview
- Merge Replication Setup
- Merge Replication Demonstration
- Why I Love Merge Replication

SQL Server Replication Overview

- Replication enables copying and distributing data and database objects from one database to another, then synchronizing between databases to maintain consistency.
- Data can be made available closer to users or applications that query the data.
- Data changes made at multiple locations can be integrated.

Reference: SQL Server Books Online, “SQL Server Replication”

Types of SQL Server Replication

1. Snapshot replication

- Point-in-time copy
- Used to initialize **transactional** and **merge** data

2. Transactional replication

- Best for placing “read-only” data closer to the users

3. Merge replication

- Best for allowing changes to be made in any copy

Replication Paradigm

Publisher

the “original” data

Subscriber

a copy of the data

Distributor

the technology for distributing the data

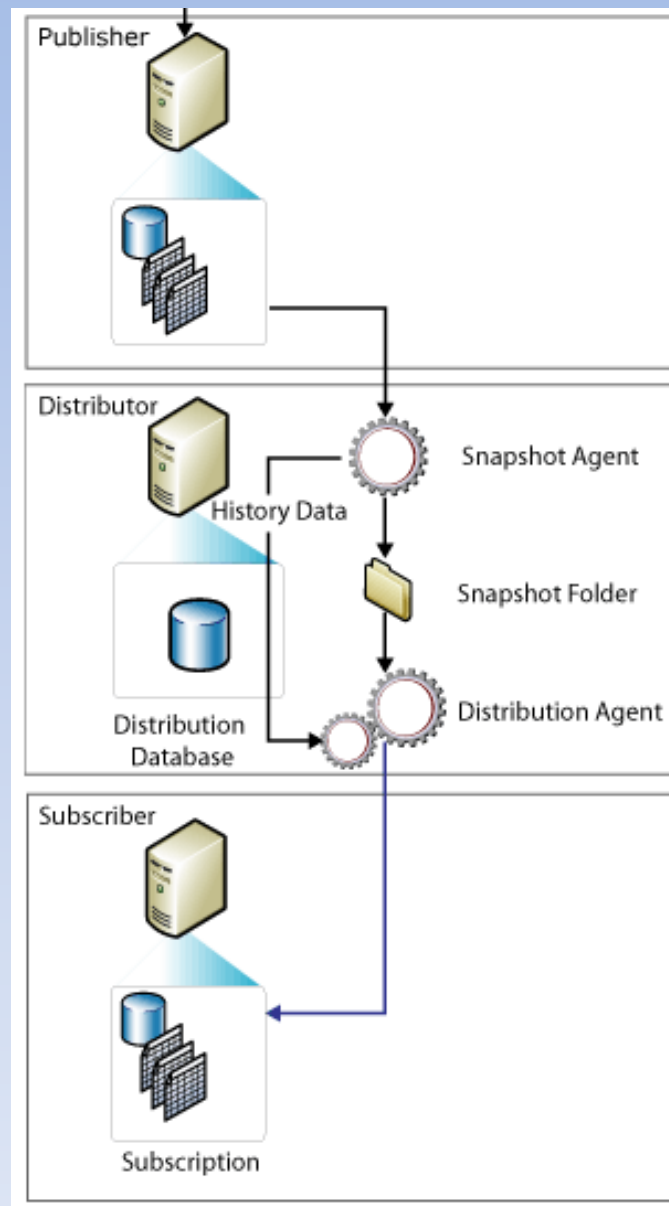
Article

an object (e.g. a table) that is replicated

Snapshot Replication

- Point-in-time copy
- Applying a snapshot to a subscriber overwrites the previous data
- Useful if data volume is small and if it is acceptable that data at the subscriber is out of date with respect to data at the publisher
- Appropriate when data changes are substantial but infrequent, as in a price list update or a purchased business list update
- Used to initialize a **transactional** or **merge** subscriber

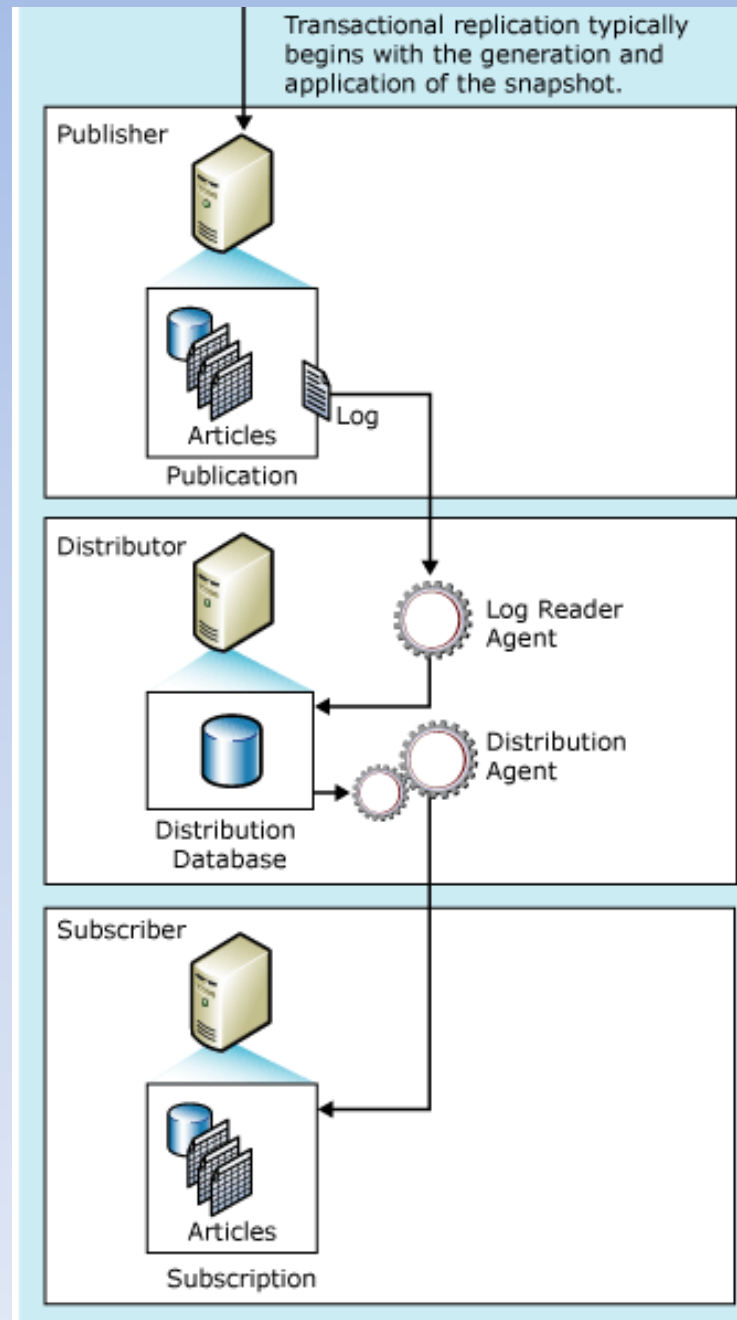
Snapshot Replication



Transactional Replication

- Use when subscriber should receive incremental data changes when they occur (very low latency)
- Use when there is very frequent insert, update, and delete activity at the publisher
- Treat subscriber as read-only. Although there are options to allow changes at the subscriber to be sent back to the publisher, those options require a good network connection or an external storage program.
- Changes are kept in and read from the transaction log

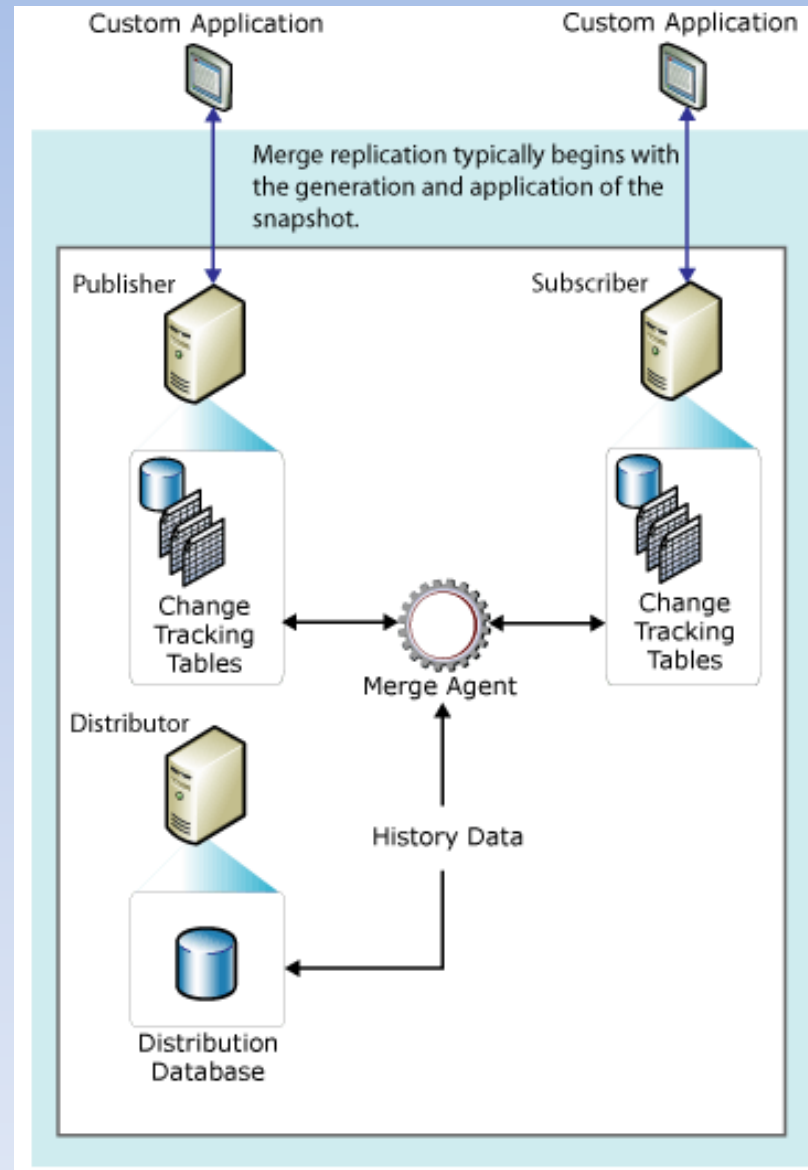
Transactional Replication



Merge Replication Overview

- Changes may be made at any copy; changes are merged to all other copies
- A copy may be offline for quite a while; when back online, data are merged
- Conflicts might occur; they can be detected and resolved
- Possible uses
 - “High availability” solution
 - Writable copies available in many locations
 - A subscriber may have a subset of the data

Merge Replication



Merge Replication Setup

- A merge subscriber is initialized using a snapshot
 - The snapshot files contain table schema, constraints and indexes, triggers, system tables, conflict tables, and table data
- If only a subset of tables are replicated, you can specify that foreign keys and/or triggers are not replicated
- Columns with the IDENTITY property (NOT FOR REPLICATION) may be assigned a value range for the publisher and subscriber(s), for each table

Merge Replication Setup, cont.

- Merge replication tracks changes by adding metadata
 - Each replicated table requires a column named “rowguid” of data type `uniqueidentifier` and default value `newid()`.
 - Many system tables are added to the database
- The Distributor is usually on the same instance as the Publisher
 - The merge agent runs on the Distributor
 - The Distributor stores the snapshot

Merge Replication Demonstration

- Merge replication setup and conflict resolution will now be demonstrated
- Only the **SalesOrderHeader (SOH)** and **SalesOrderDetails (SOD)** tables from Microsoft's **AdventureWorks2012** sample database are replicated
- The following slides show key screen images from the demonstration

Merge Replication Demonstration

Preliminary Tasks

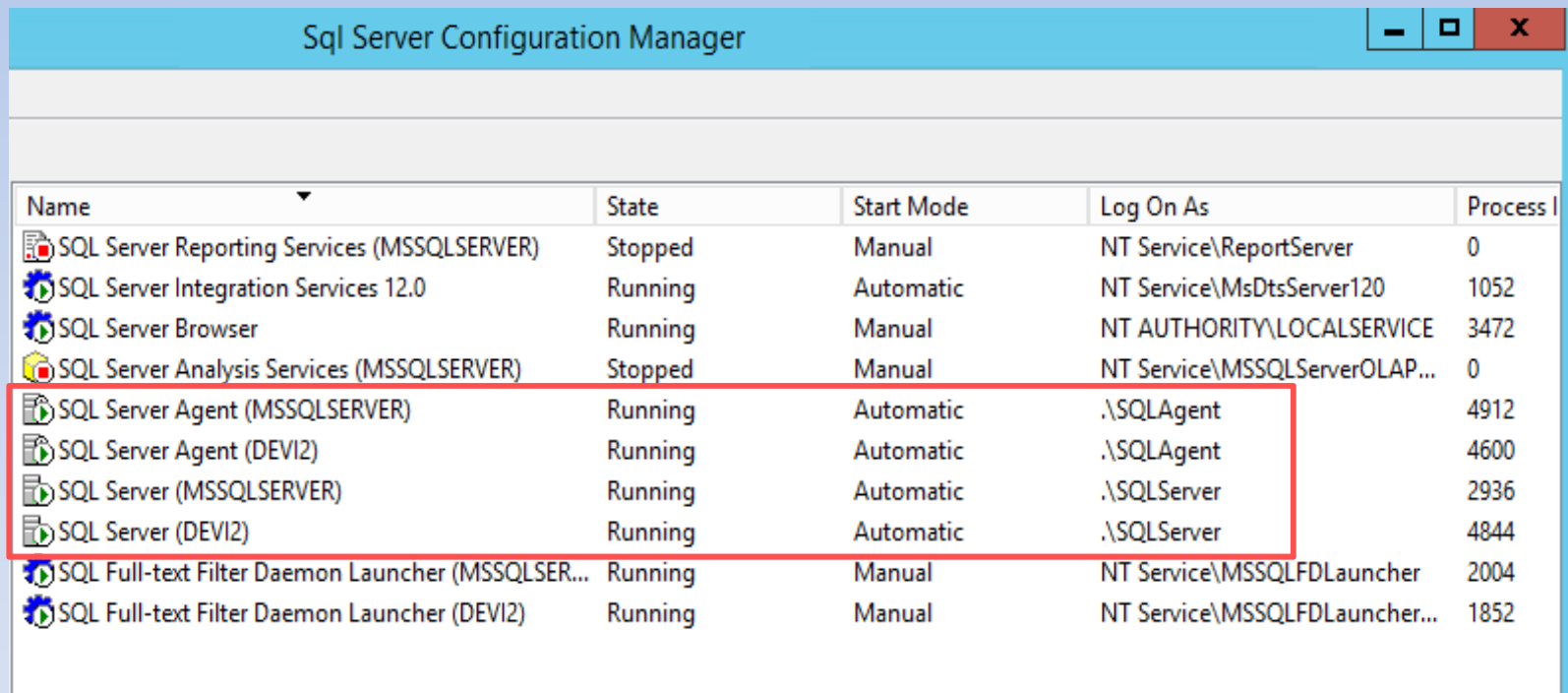
- Ensure the SQL Server and SQL Server Agent accounts are the same on all instances and have sysadmin membership on all instances

For this demonstration (AdventureWorks2012)

- Triggers and some Foreign Keys have been dropped from the SOH and SOD tables
- A stored procedure, Sales.AddOrder, has been created to simulate the addition of an order
- Scripts to accomplish these are available on request

Demo: Creating the Publication

Service accounts for the default (SANDY14) and named (SANDY14\DEVI2) instances have been set

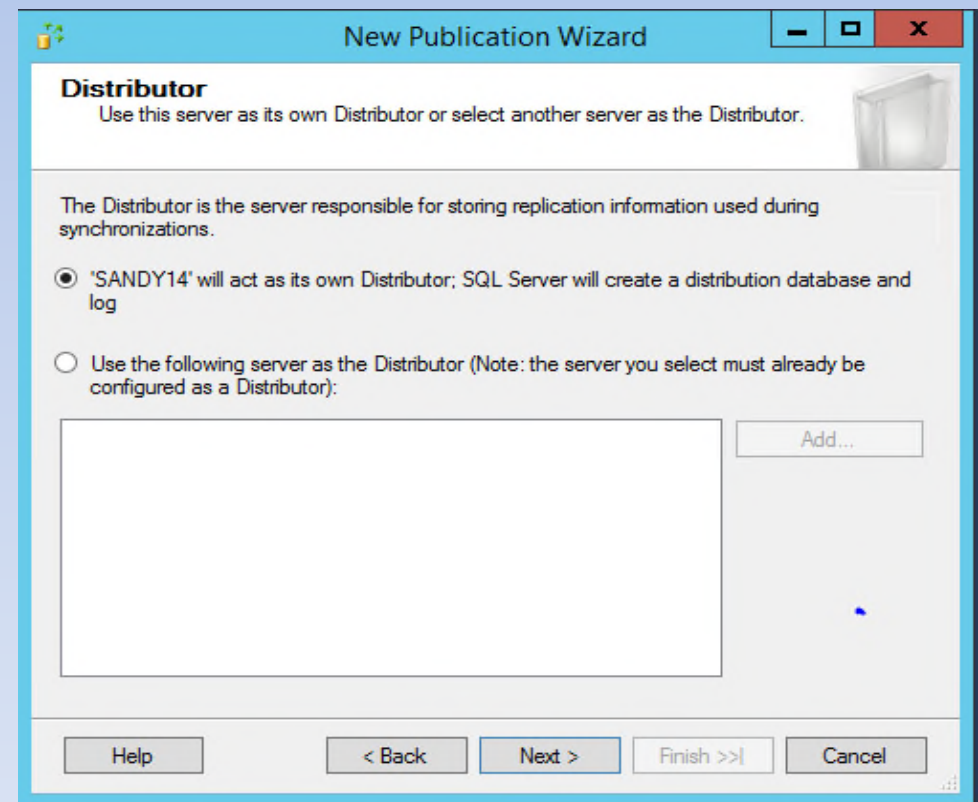
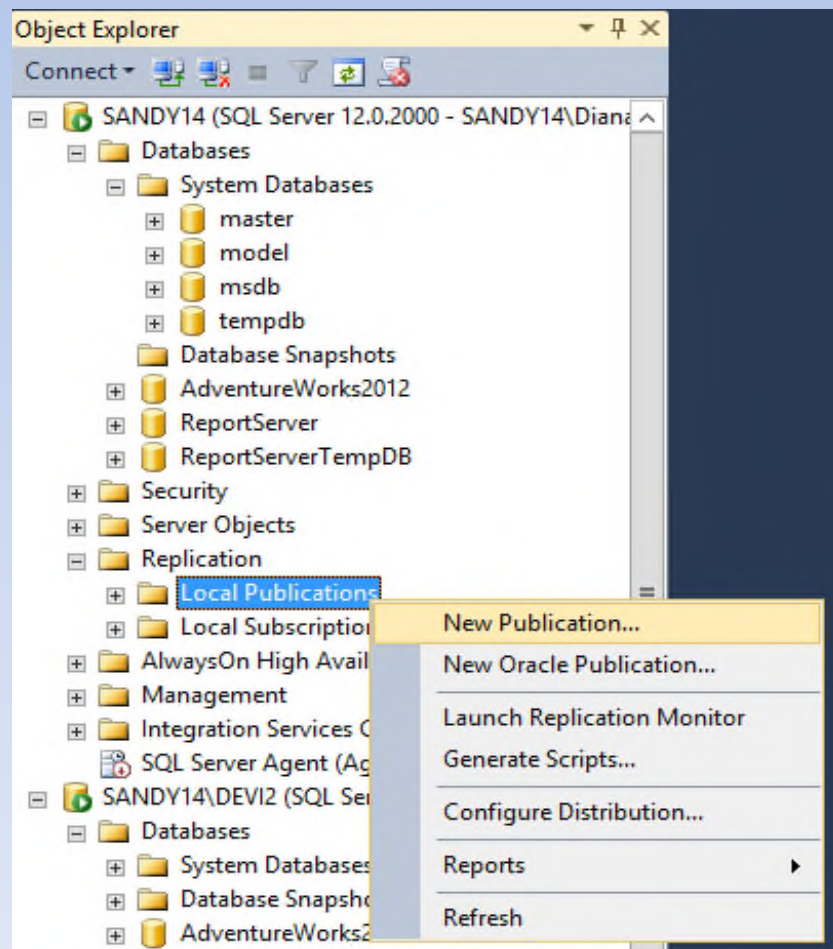


Name	State	Start Mode	Log On As	Process I
SQL Server Reporting Services (MSSQLSERVER)	Stopped	Manual	NT Service\ReportServer	0
SQL Server Integration Services 12.0	Running	Automatic	NT Service\MsDtsServer120	1052
SQL Server Browser	Running	Manual	NT AUTHORITY\LOCALSERVICE	3472
SQL Server Analysis Services (MSSQLSERVER)	Stopped	Manual	NT Service\MSSQLServerOLAP...	0
SQL Server Agent (MSSQLSERVER)	Running	Automatic	.\SQLAgent	4912
SQL Server Agent (DEV12)	Running	Automatic	.\SQLAgent	4600
SQL Server (MSSQLSERVER)	Running	Automatic	.\SQLServer	2936
SQL Server (DEV12)	Running	Automatic	.\SQLServer	4844
SQL Full-text Filter Daemon Launcher (MSSQLSER...	Running	Manual	NT Service\MSSQLFDLauncher	2004
SQL Full-text Filter Daemon Launcher (DEV12)	Running	Manual	NT Service\MSSQLFDLauncher...	1852

Demo: Creating the Publication

Right-click **Local Publications**
and then select **New Publication**

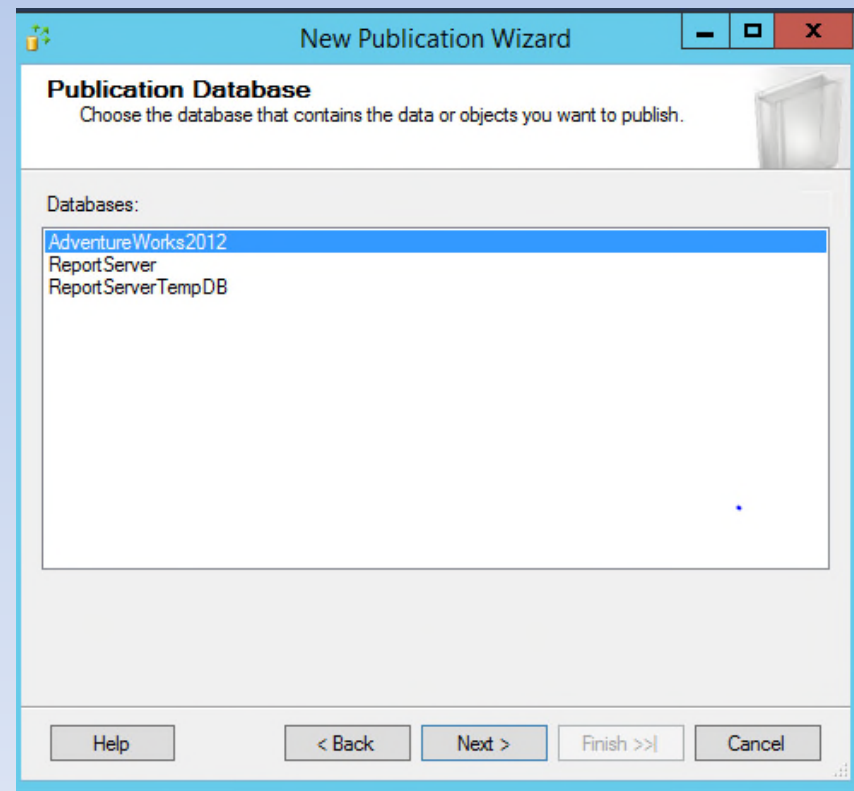
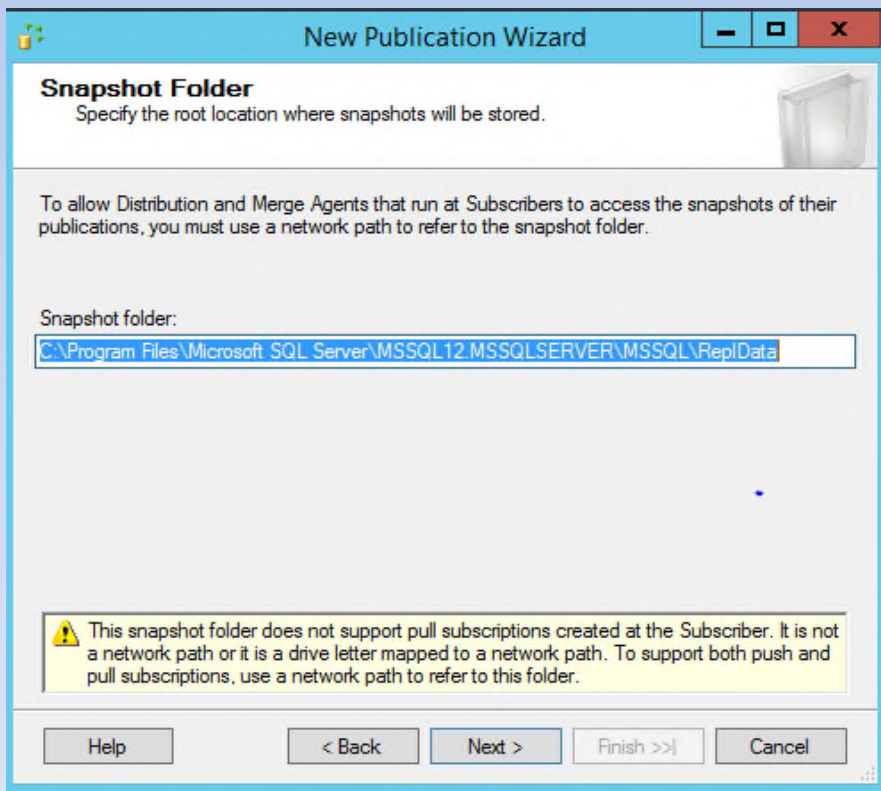
In the **New Publication Wizard**
the publisher acts as its distributor



Demo: Creating the Publication

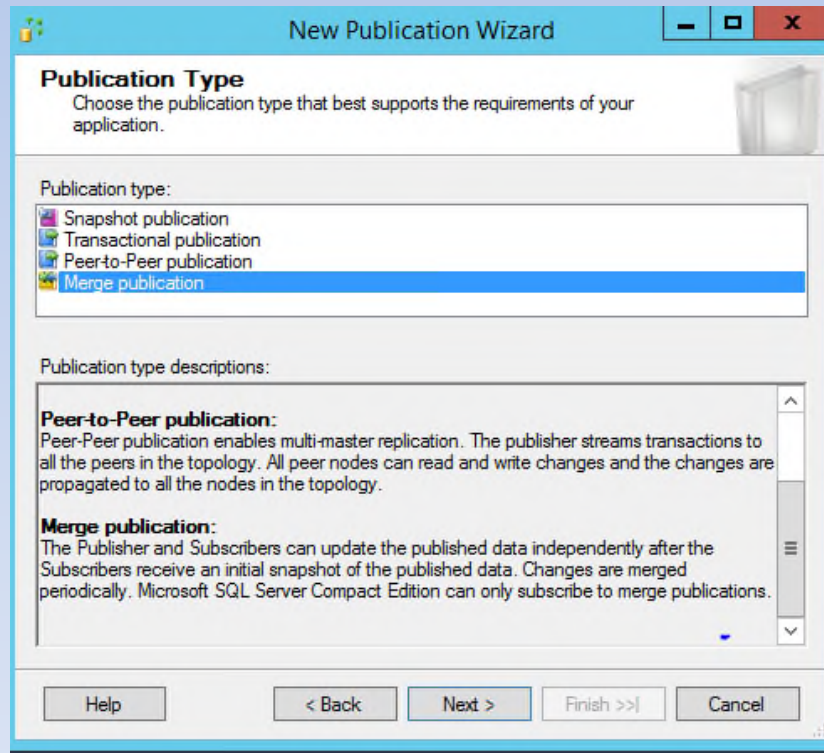
The snapshot folder, repldata, must be changed to a share if you need a *pull* subscriber.

The publication database is chosen.

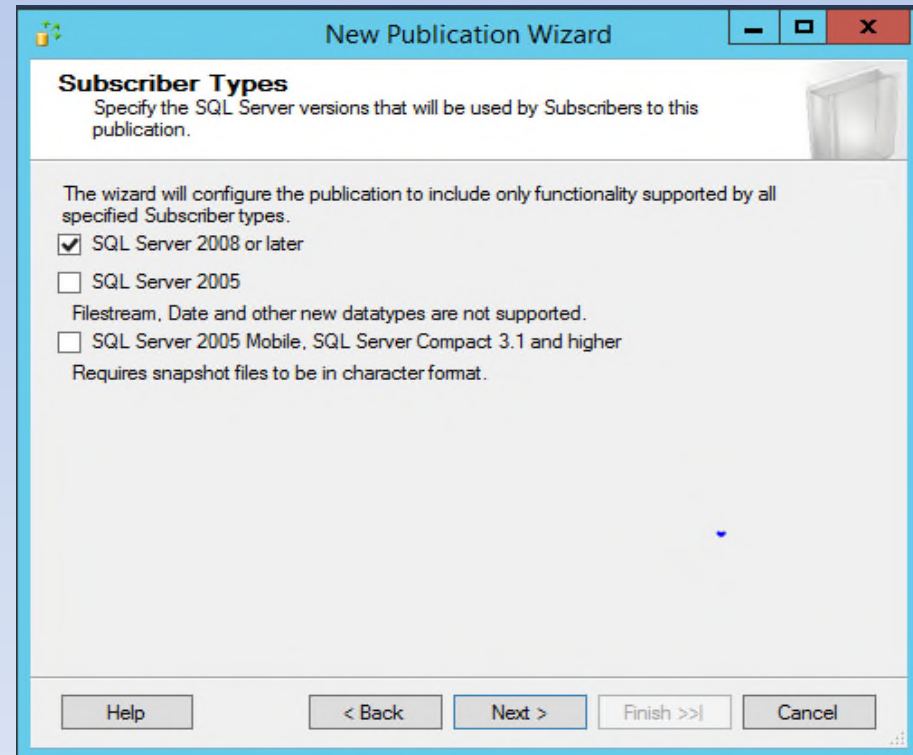


Demo: Creating the Publication

The type of replication, *merge*, is chosen.

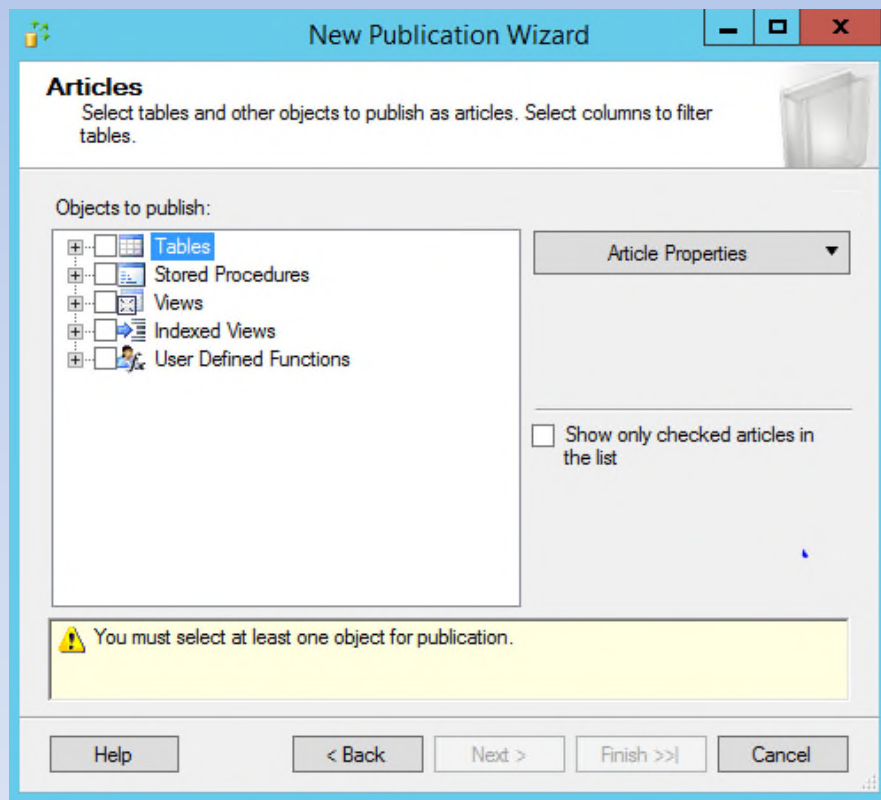


Subscriber versions are chosen. The publisher version must be equal to or later than the version of the subscriber.

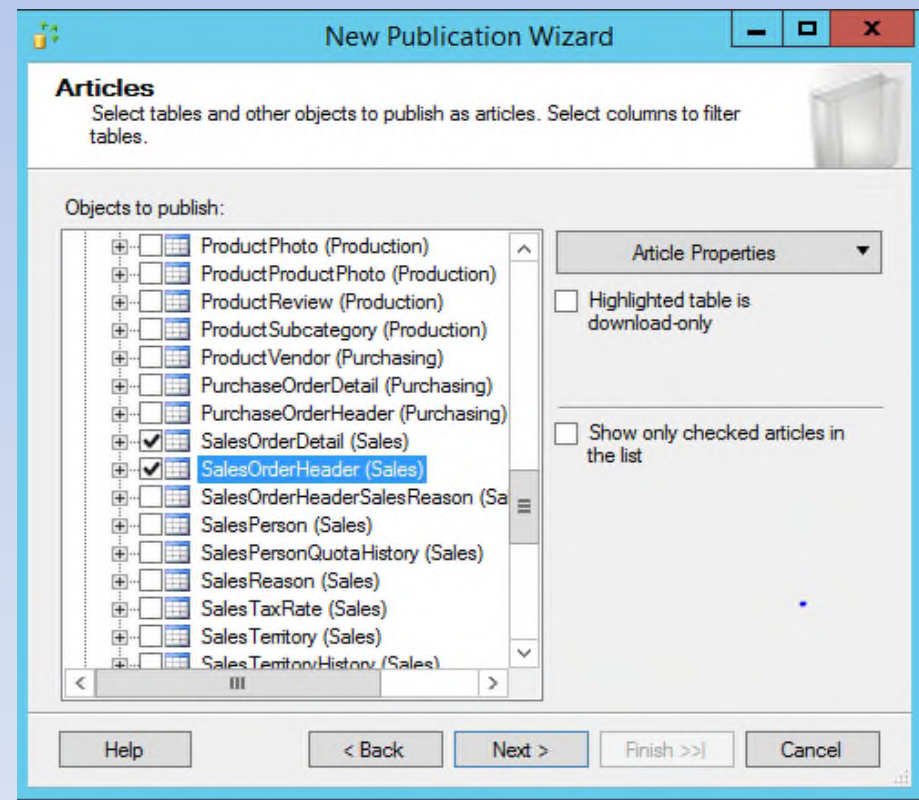


Demo: Creating the Publication

The tables to replicate are chosen.



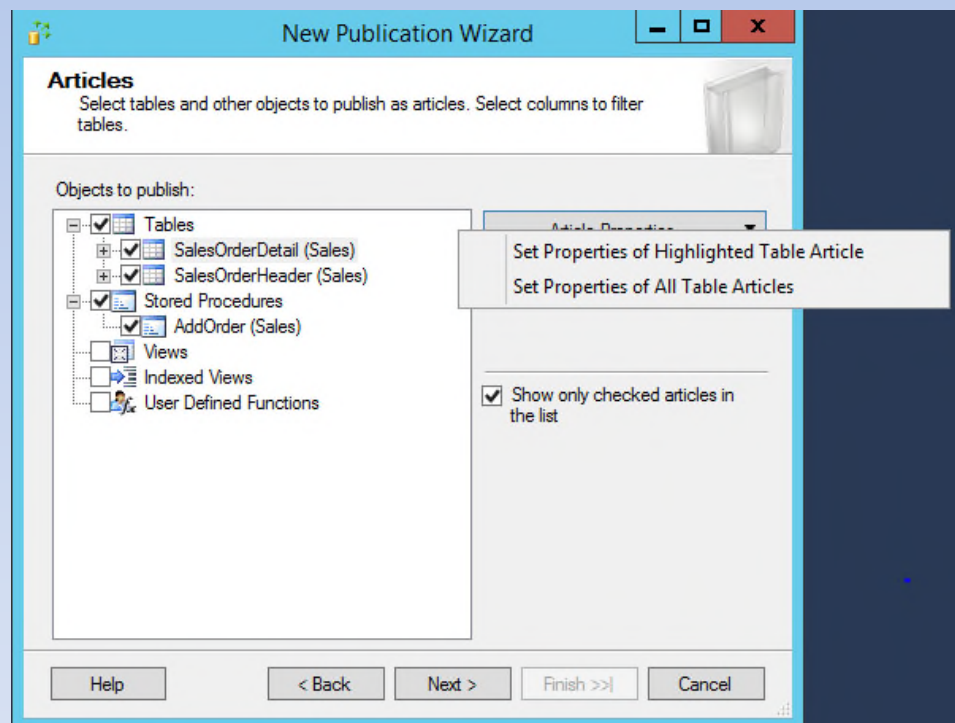
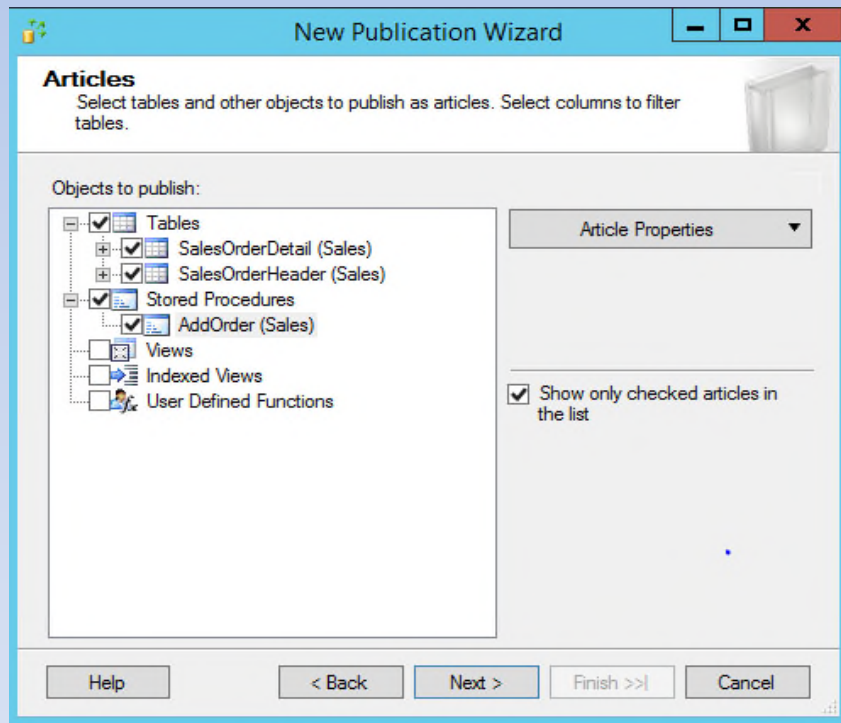
(Usually all tables would be replicated.)



Demo: Creating the Publication

Also replicated is a stored procedure, Sales.AddOrder.

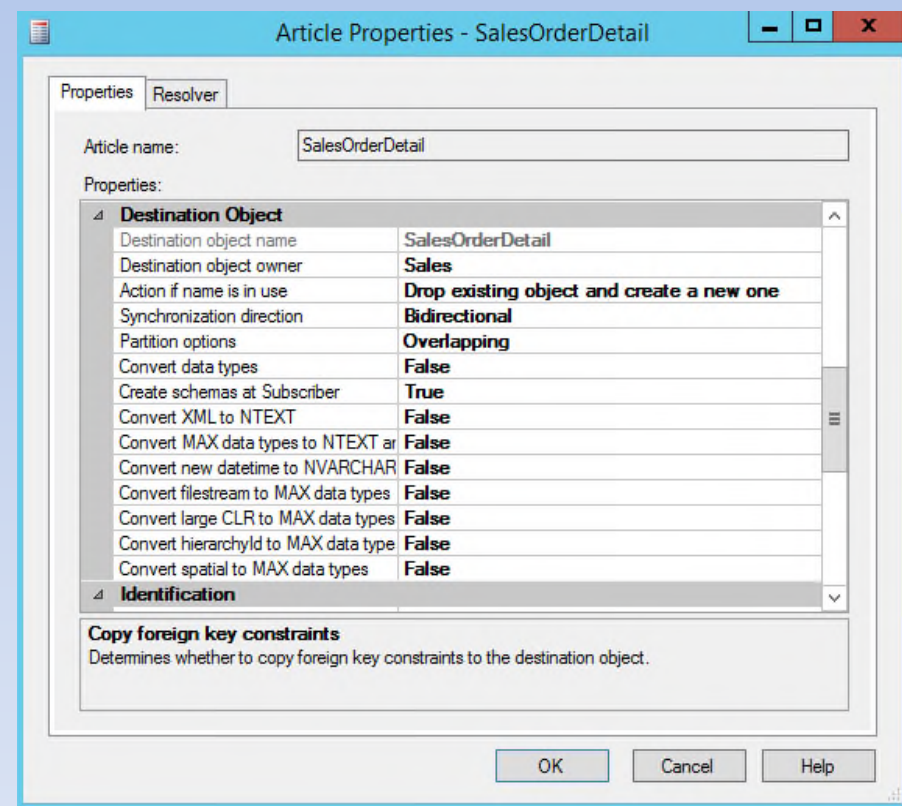
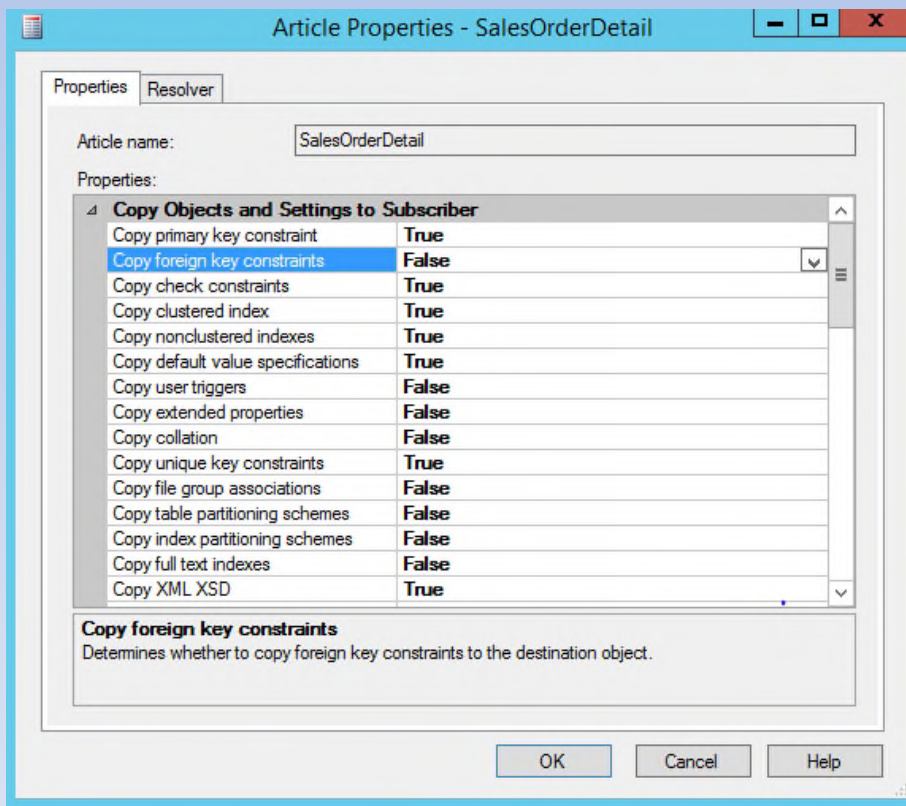
Click **Article Properties** to set properties of an article



Demo: Creating the Publication

In this case, in SOD we chose not to replicate F.K.s or Triggers.

Other article properties (that we did not change).



Demo: Creating the Publication

SOD identity ranges and Column-level tracking were set.

Not replicating F.K.s and triggers, column-level tracking, and identity ranges were set for SOH also.

Article Properties - SalesOrderDetail

Properties Resolver

Article name: SalesOrderDetail

Properties:

Description	
Source object name	SalesOrderDetail
Source object owner	Sales
Identity Range Management	
Automatically manage identity ranges	Automatic
Publisher range size	1000
Subscriber range size	1000
Range threshold percentage	80
Next range starting value	
Maximum identity value	2147483647
Identity increment	1
Merging Changes	
Tracking level	Column-level tracking
Verify INSERT permission	False
Verify UPDATE permission	False
Verify DELETE permission	False
Multicolumn UPDATE	True

Tracking level
Determines whether to treat changes to the same row or to the same column as a conflict.

OK Cancel Help

Article Properties - SalesOrderHeader

Properties Resolver

Article name: SalesOrderHeader

Properties:

Source object name	SalesOrderHeader
Source object owner	Sales
Identity Range Management	
Automatically manage identity ranges	Automatic
Publisher range size	500
Subscriber range size	500
Range threshold percentage	80
Next range starting value	
Maximum identity value	2147483647
Identity increment	1
Merging Changes	
Tracking level	Column-level tracking
Verify INSERT permission	False
Verify UPDATE permission	False
Verify DELETE permission	False
Multicolumn UPDATE	True

Tracking level
Determines whether to treat changes to the same row or to the same column as a conflict.

OK Cancel Help

Demo: Creating the Publication

No article properties for the stored procedure was changed

The Wizard warns of an issue when replicating a stored proc

The 'Article Properties - AddOrder' dialog box shows the configuration for a new article. The 'Article name' is 'AddOrder'. Under 'Properties', the 'Copy Objects and Settings to Subscriber' section is expanded, showing 'Copy extended properties' and 'Copy permissions' both set to 'False'. The 'Destination Object' section is also expanded, showing 'Destination object name' as 'AddOrder', 'Destination object owner' as 'Sales', 'Action if name is in use' as 'Drop existing object and create a new one', and 'Create schemas at Subscriber' as 'True'. The 'Identification' section is expanded, with 'Description' selected, showing 'Source object name' as 'AddOrder' and 'Source object owner' as 'Sales'. A 'Description' field at the bottom contains the text 'The description of the article.'.

Copy Objects and Settings to Subscriber	
Copy extended properties	False
Copy permissions	False

Destination Object	
Destination object name	AddOrder
Destination object owner	Sales
Action if name is in use	Drop existing object and create a new one
Create schemas at Subscriber	True

Identification	
Description	
Source object name	AddOrder
Source object owner	Sales

Description
The description of the article.

OK Cancel Help

The 'New Publication Wizard' shows a warning under 'Article Issues'. The message states: 'The following issues may require changes to your application to ensure it continues to operate as expected.' Below this, a list of issues is shown, with 'Objects referenced by stored procedures are required.' selected. A description follows, explaining that SQL Server requires all objects referenced by published stored procedures to be available at the Subscriber. It lists the following stored procedures published in this publication: '[Sales].[AddOrder]'.

Article Issues
The following issues may require changes to your application to ensure it continues to operate as expected.

Issues:

- Objects referenced by stored procedures are required.

Description:

SQL Server requires that all objects referenced by published stored procedures, such as tables and user-defined functions, be available at the Subscriber. If the referenced objects are not published as articles in this publication, they must be created at the Subscriber manually.

The following stored procedures are published in this publication:

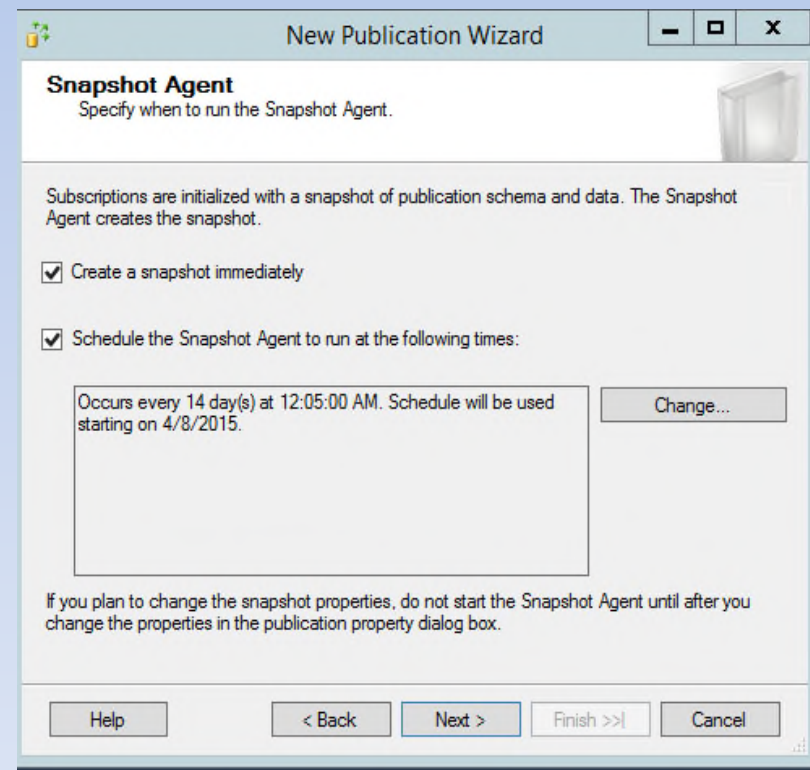
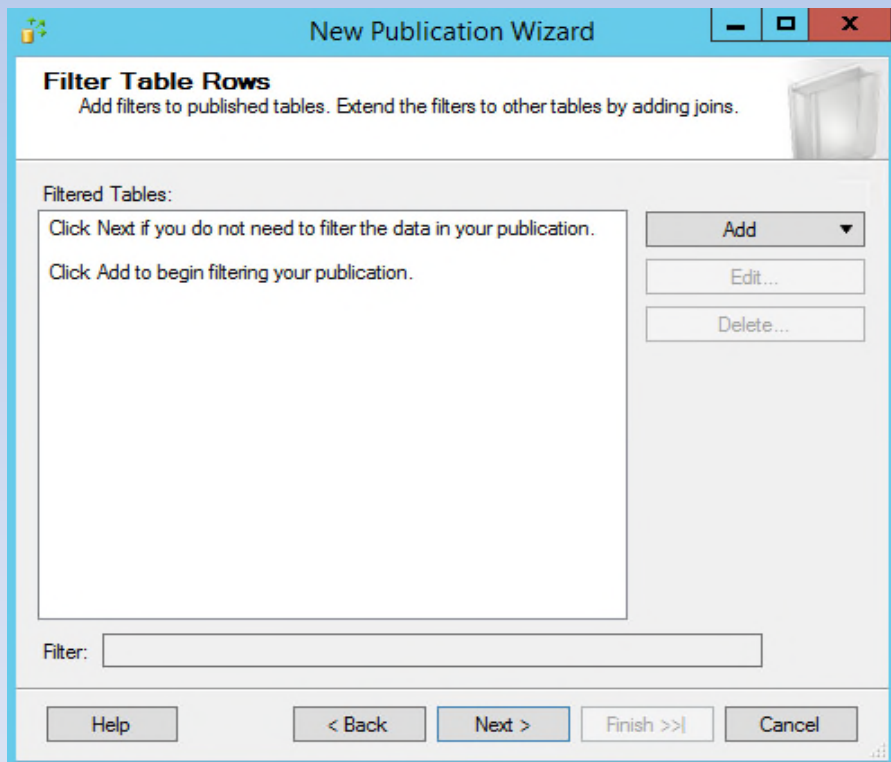
[Sales].[AddOrder]

Help < Back Next > Finish >> Cancel

Demo: Creating the Publication

Table rows may be filtered; we did not.

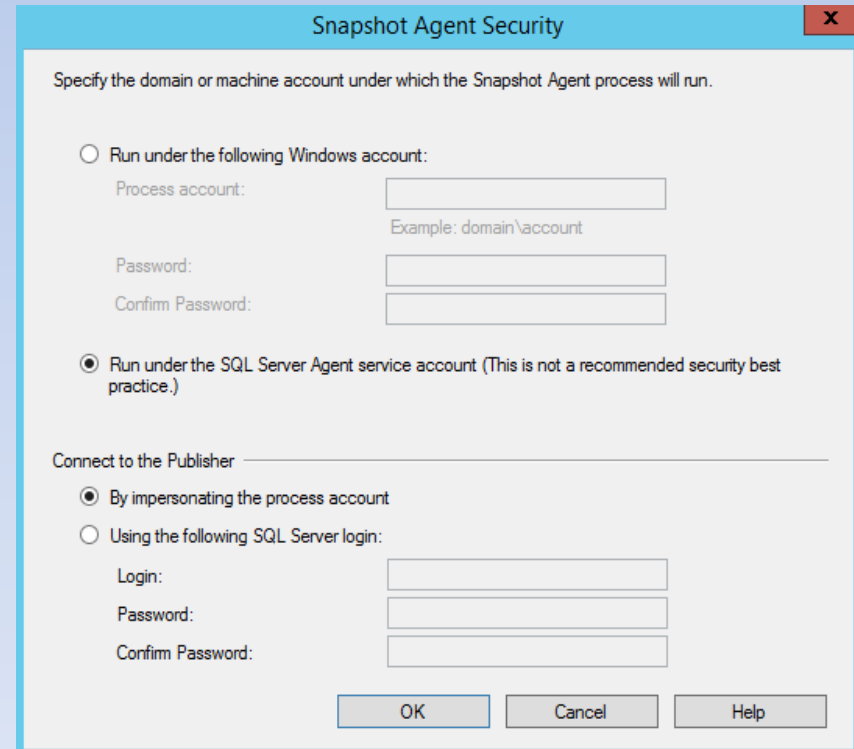
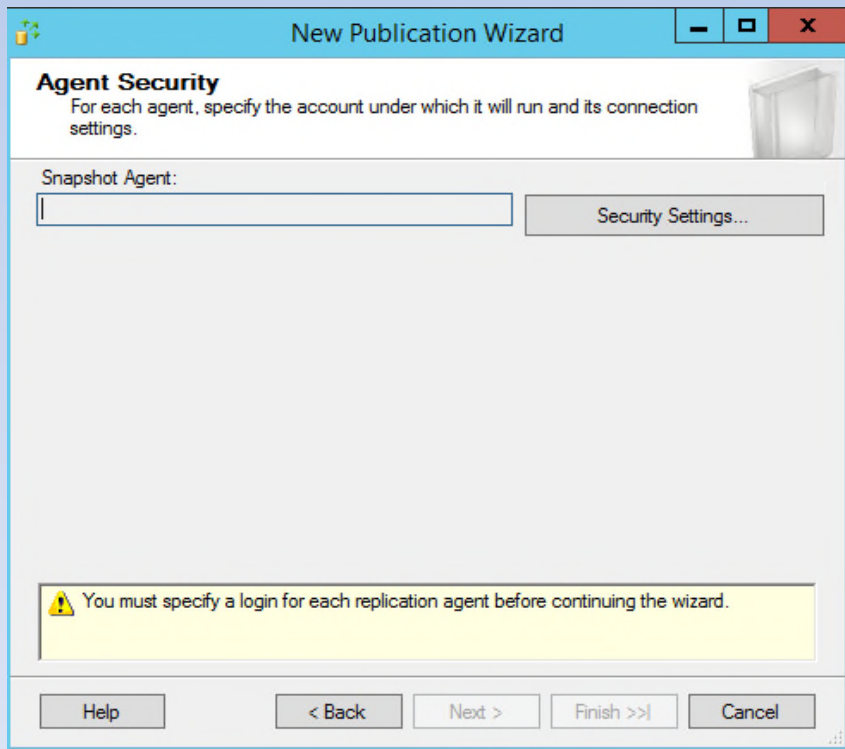
We chose to generate the snapshot immediately and to make the expiration interval 14 days.



Demo: Creating the Publication

The Snapshot Agent account must be specified. Click the **Security Settings** button.

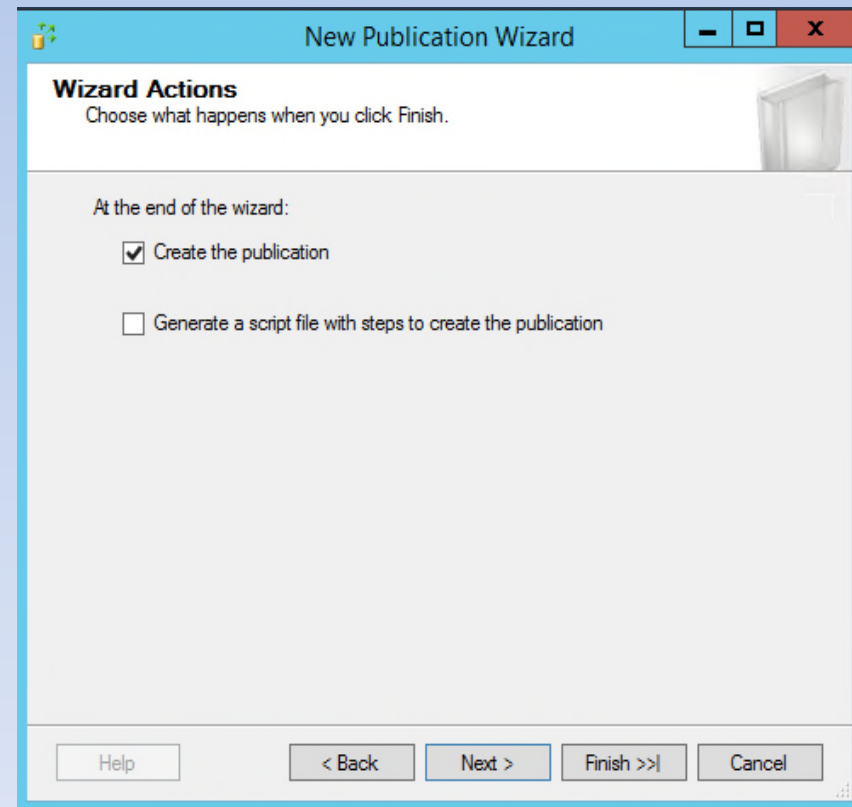
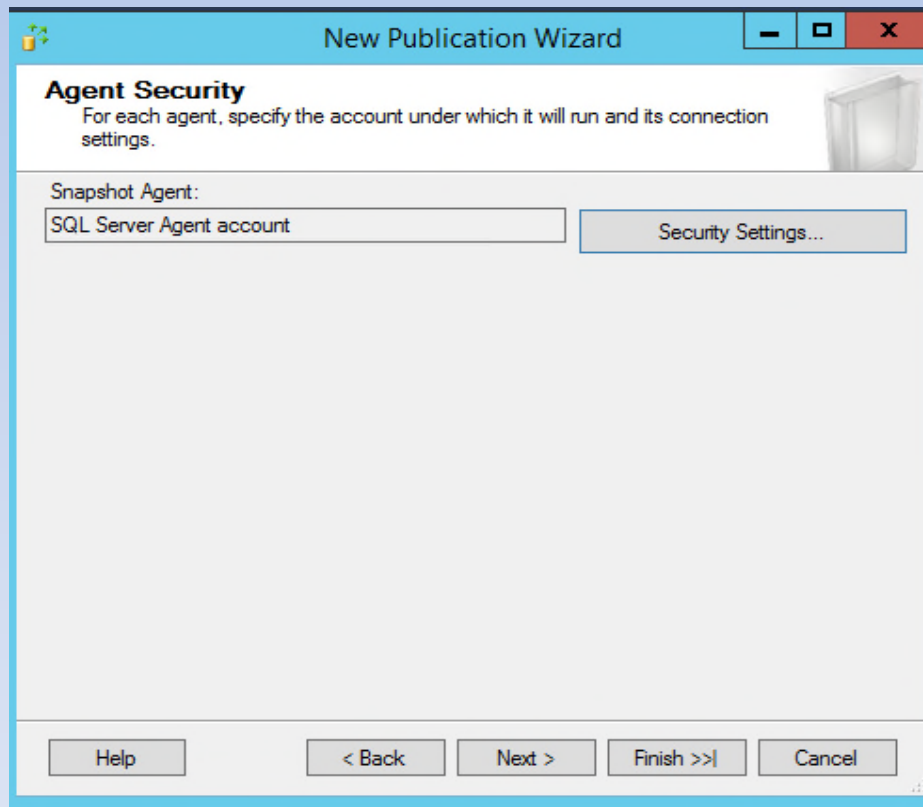
It is recommended that you have a unique user account for each replication agent; we chose to use the SQL Agent.



Demo: Creating the Publication

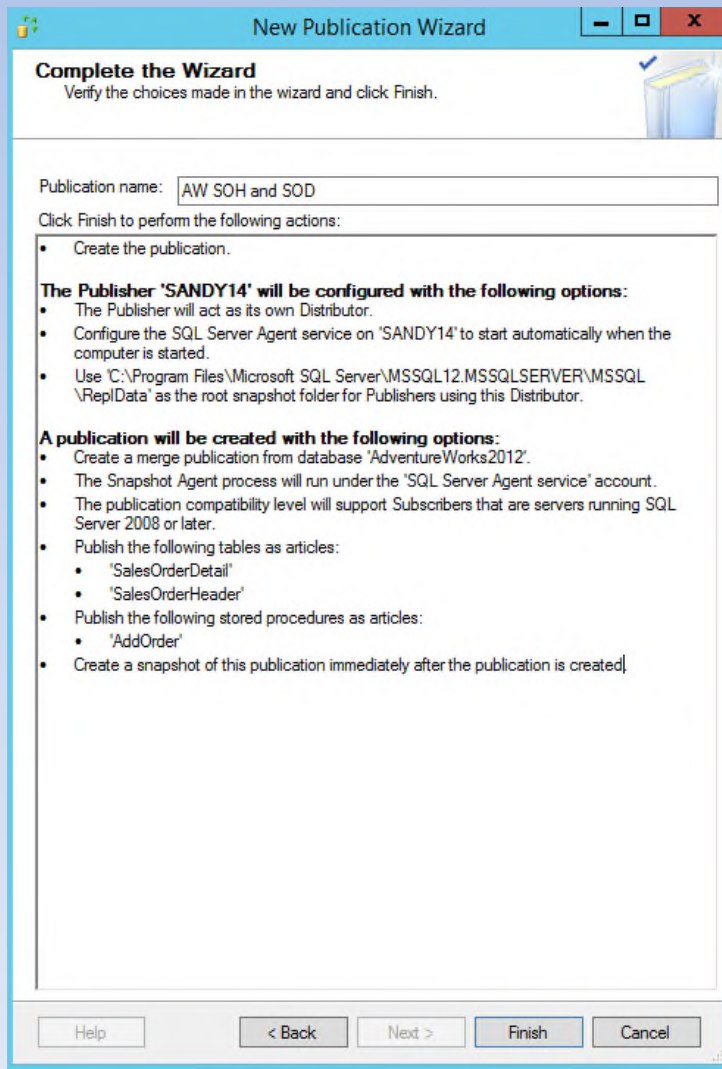
The **Agent Security** dialog shows the SQL Server Agent choice for the snapshot agent.

We choose to create the publication

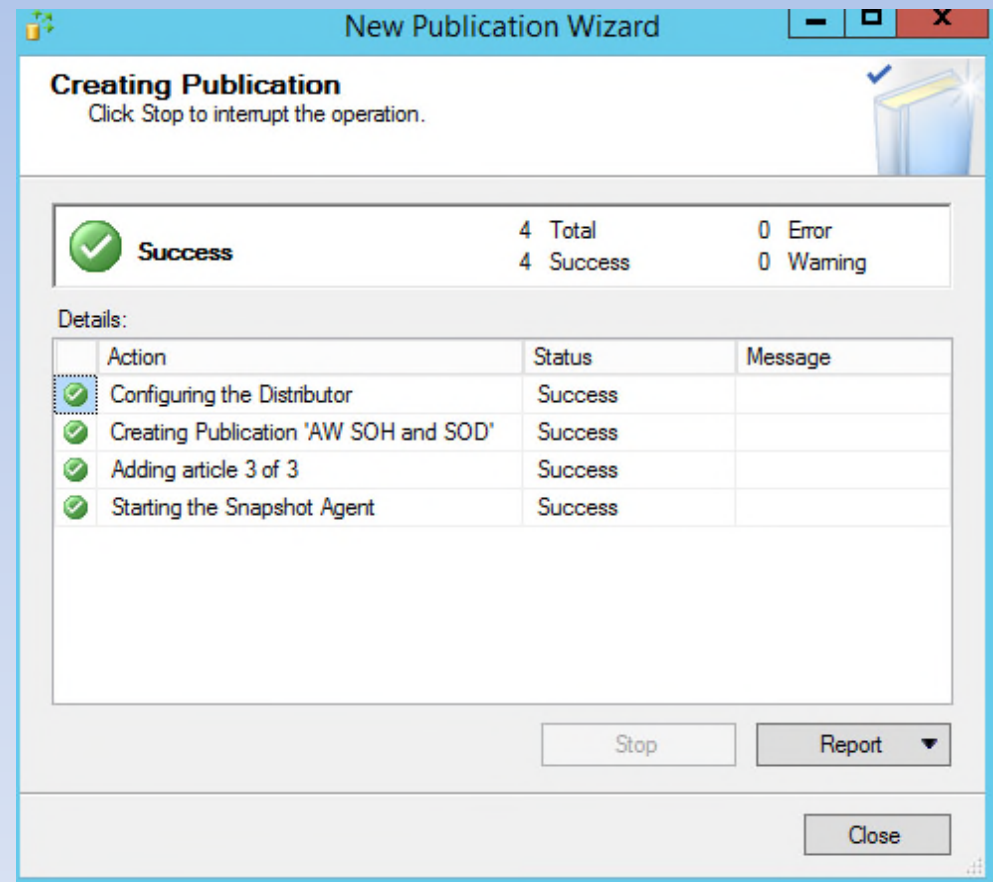


Demo: Creating the Publication

The actions the Wizard will perform are listed.

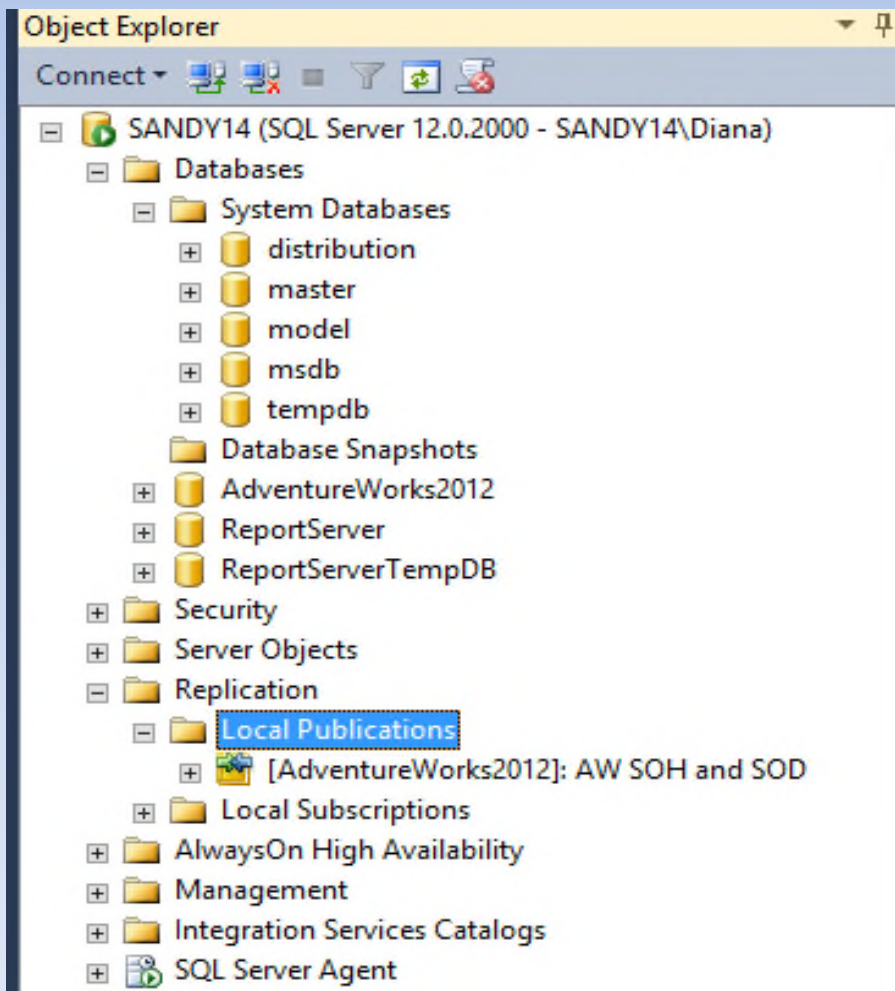


The publication is successfully created.

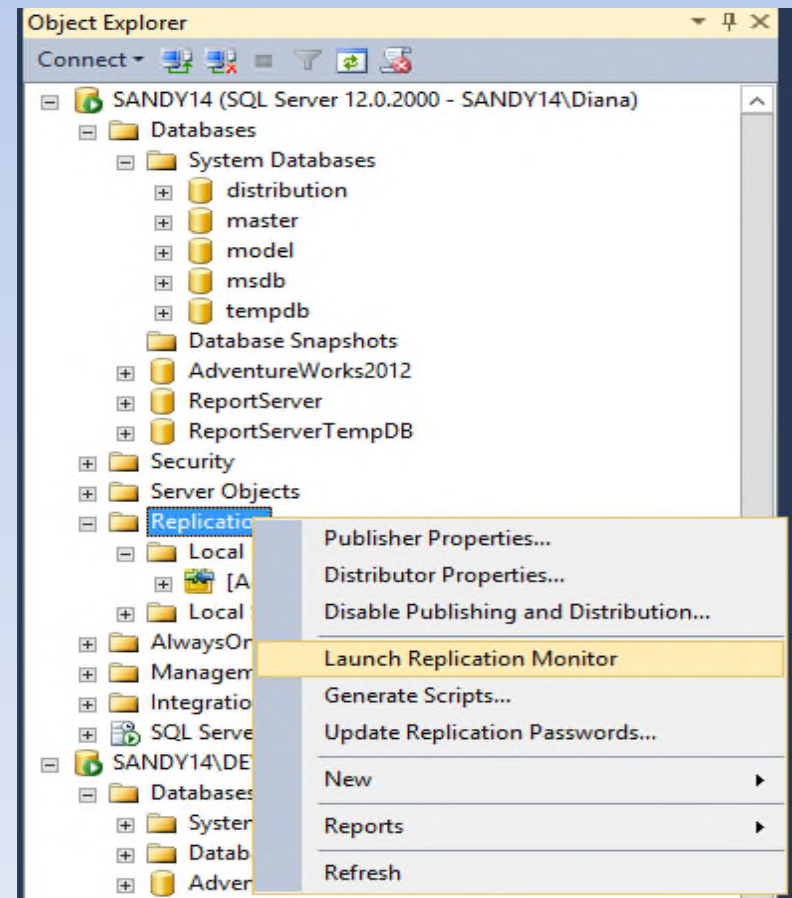


Demo: Creating the Publication

Object Explorer shows the publication.

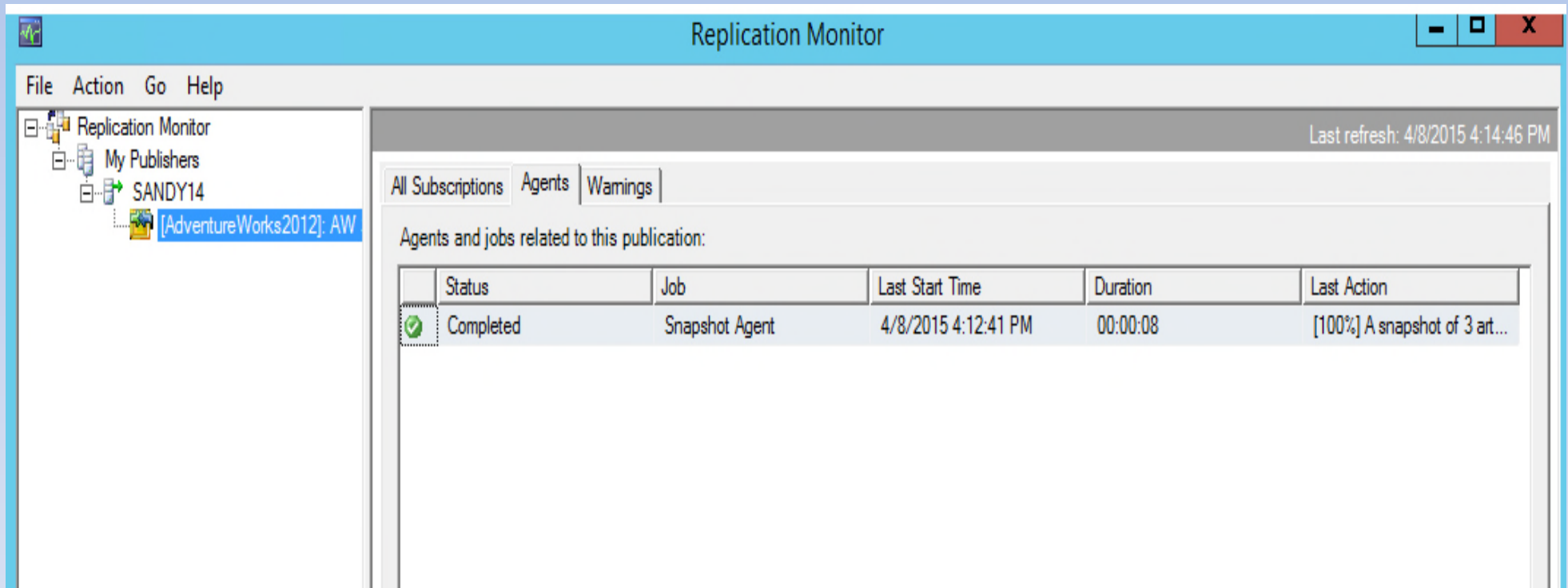


Right-click **Replication** and select **Launch Replication Monitor** to see if there are any problems.



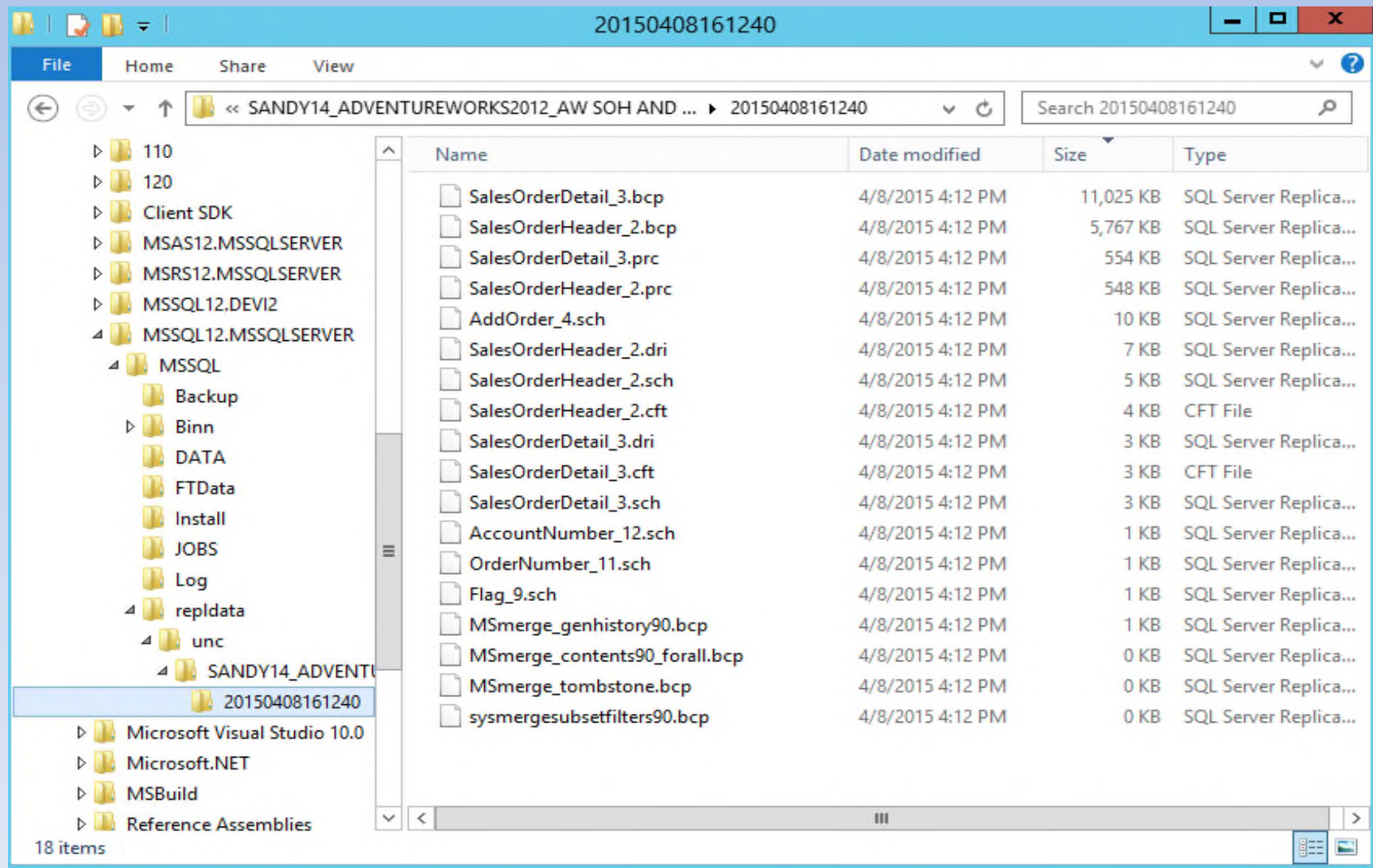
Demo: Creating the Publication

Replication Monitor shows the snapshot was created.



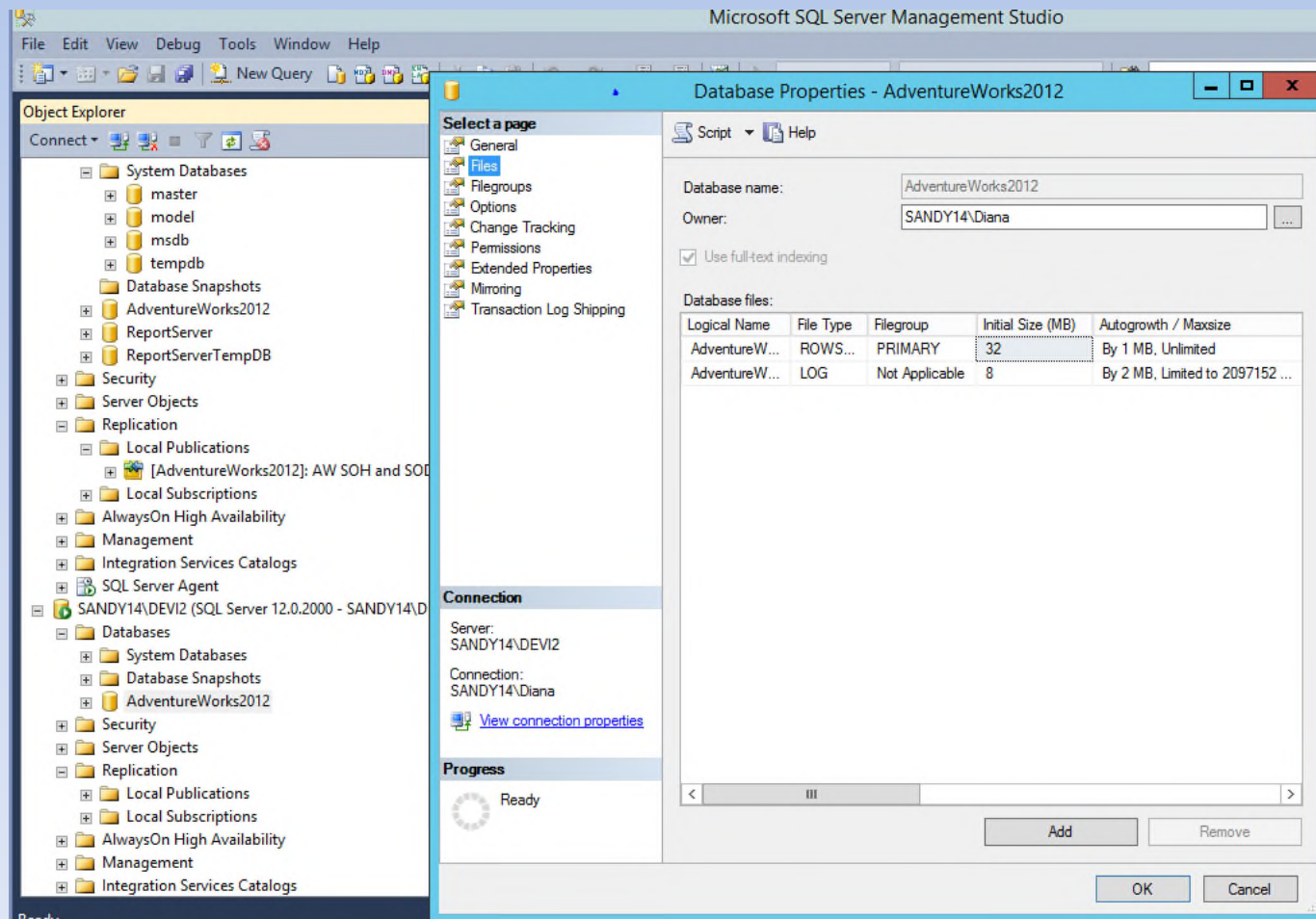
Demo: Creating the Publication, concl.

These are the snapshot files in the **repldata** folder.



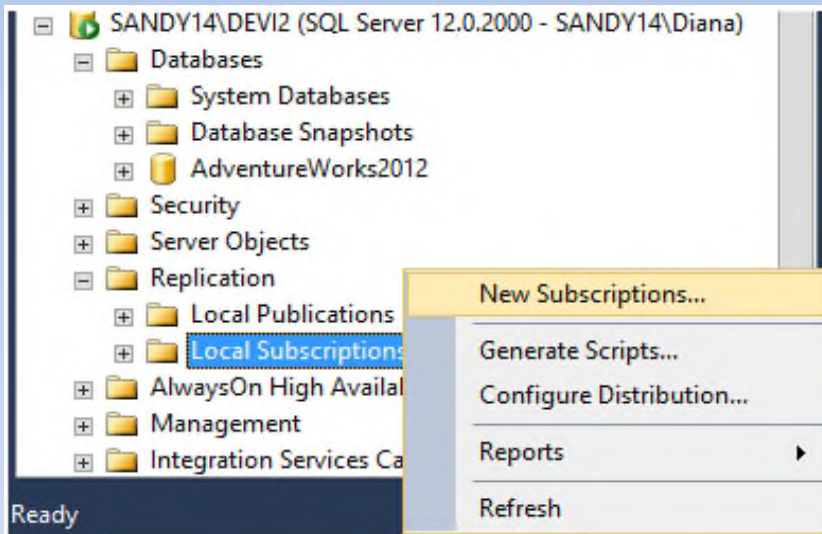
Demo: Creating the Subscription

A database with the same name, AdventureWorks2012, was created on the SQL Server instance, to hold the subscription.

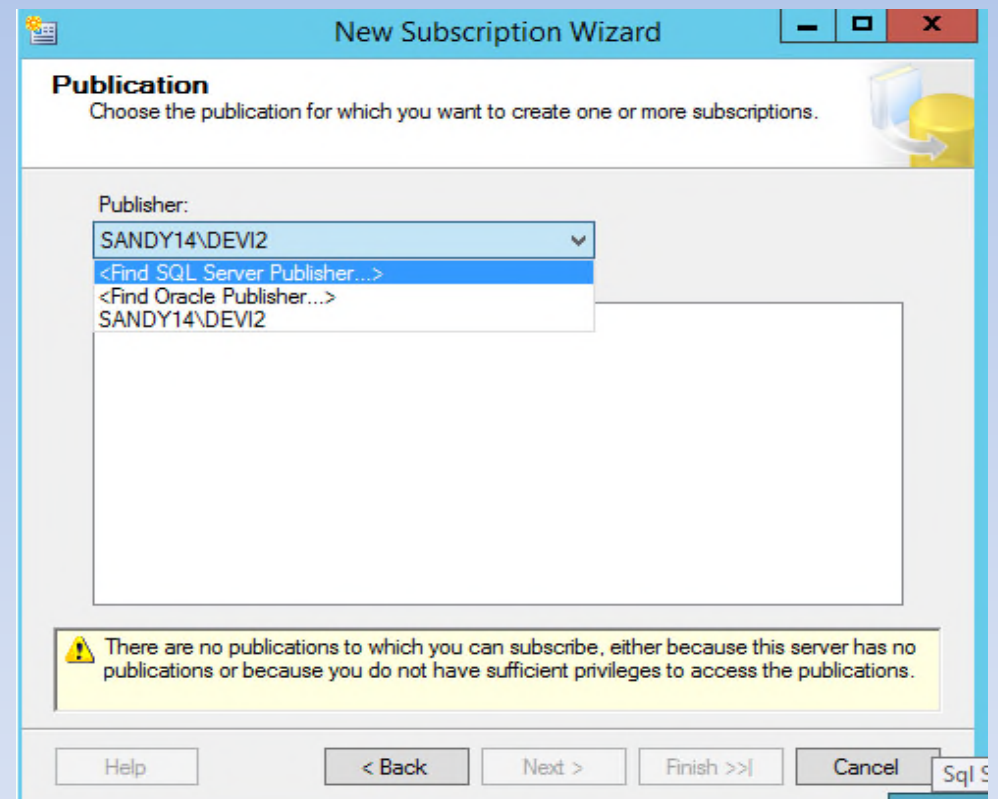


Demo: Creating the Subscription

On the subscription instance, start the **New Subscription Wizard**



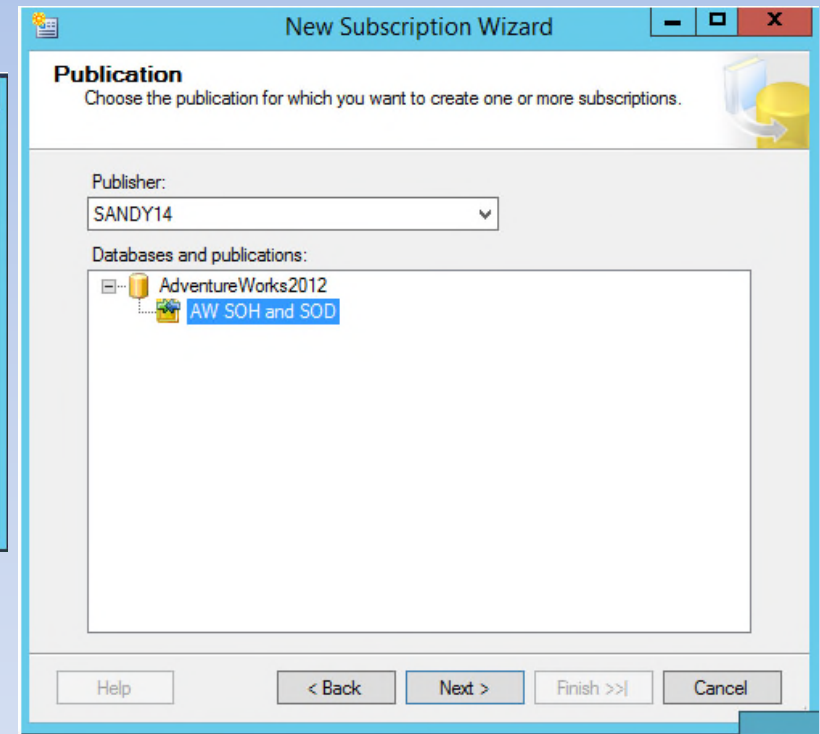
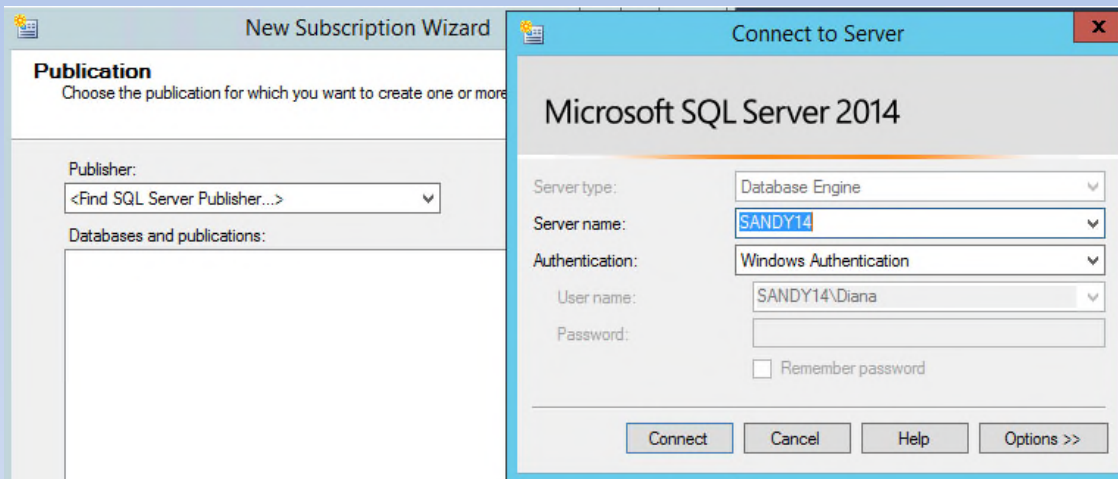
Find the SQL Server publisher.



Demo: Creating the Subscription

Log into (connect to) the publisher, and then ...

... select the publication.



Demo: Creating the Subscription

Select either a push or a pull subscription. We chose push.

Select the subscriber database. You could create a new database at this time. You can also add other push subscribers.

New Subscription Wizard

Merge Agent Location
Choose where to run the Merge Agent(s).

For the subscriptions I create in this wizard:

- ☒ Run all agents at the Distributor, SANDY14 (push subscriptions)
This option makes it easier to administer the synchronization of subscriptions centrally.
- ☐ Run each agent at its Subscriber (pull subscriptions)
This option reduces the processing overhead at the Distributor and lets each Subscriber administer the synchronization of its subscription.

Run the wizard more than once if you want some agents to run at the Distributor and some to run at Subscribers.

Buttons: Help, < Back, Next >, Finish >>, Cancel

New Subscription Wizard

Subscribers
Choose one or more Subscribers and specify each subscription database.

Subscribers and subscription databases:

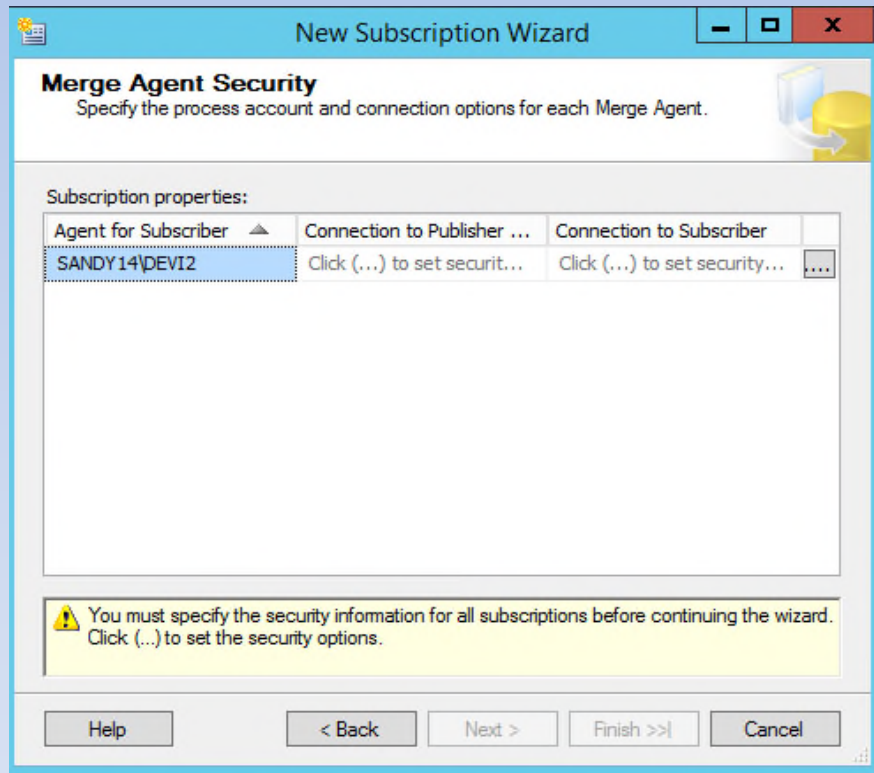
Subscriber	Subscription Database
<input checked="" type="checkbox"/> SANDY14\DEVI2	AdventureWorks2012

Add SQL Server Subscriber...

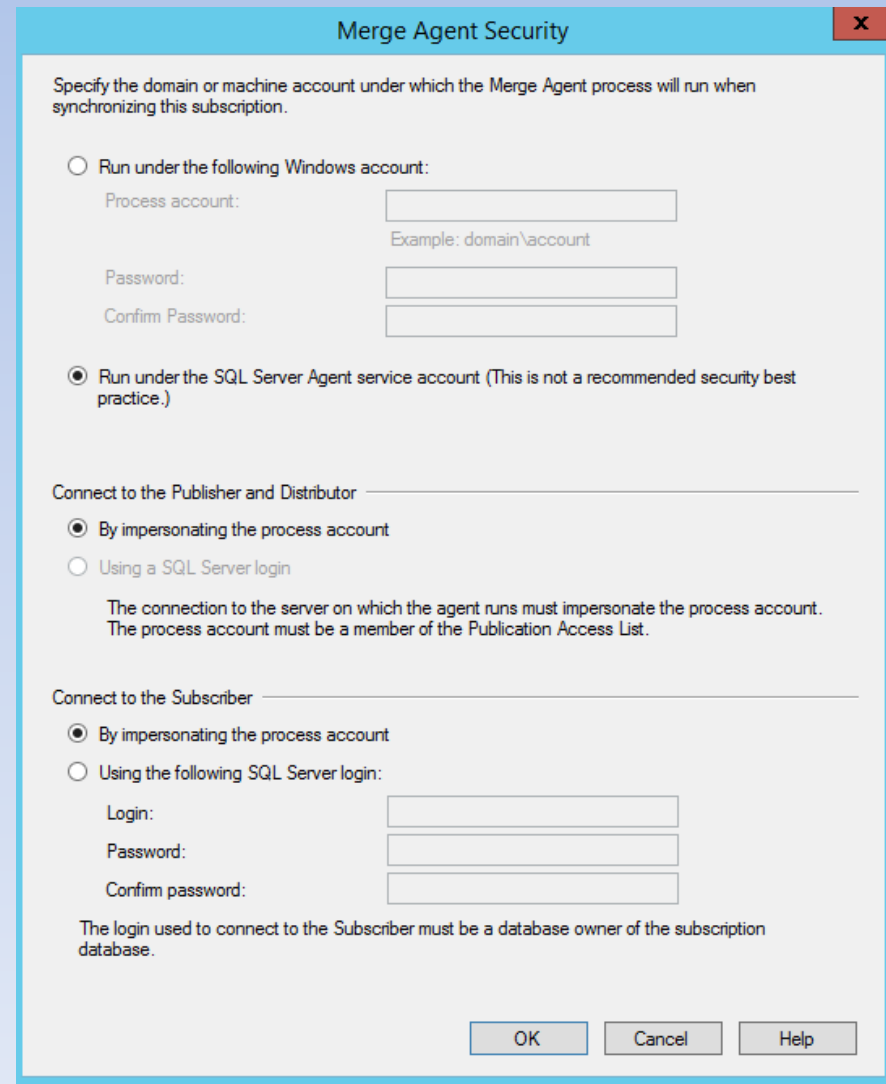
Buttons: Help, < Back, Next >, Finish >>, Cancel

Demo: Creating the Subscription

Set the merge agent account(s).

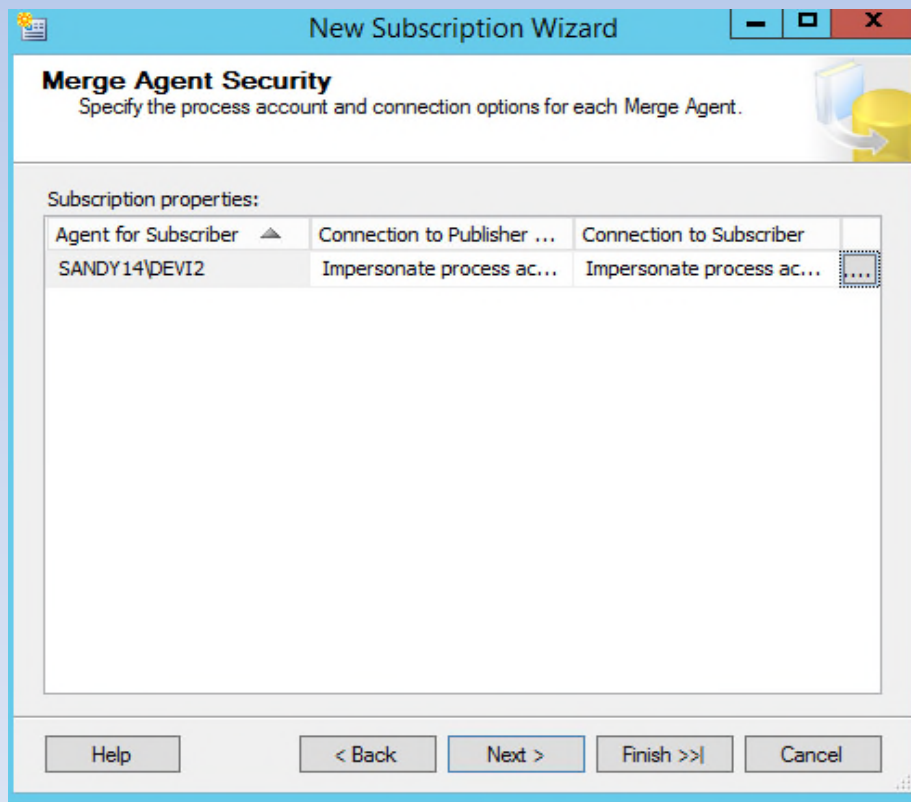


Again, we chose to use the SQL Agent account.

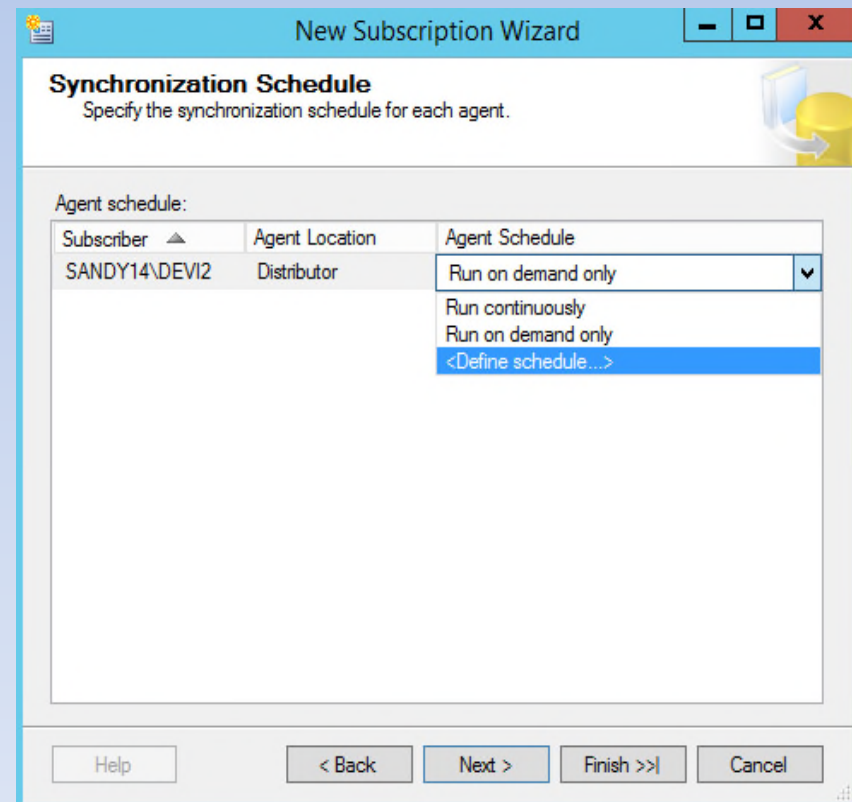


Demo: Creating the Subscription

The Security dialog displays our choices.



Next, we choose a schedule. **Run Continuously** is recommended. We chose a schedule so we could demonstrate conflicts.



Demo: Creating the Subscription

The “every one minute” schedule will allow us to demo replication conflicts. Most people choose “Run Continuously”

New Job Schedule

Name:

Schedule type: ☒ Enabled

One-time occurrence

Date: Time:

Frequency

Occurs:

Recurs every: day(s)

Daily frequency

☐ Occurs once at:

☒ Occurs every: minute(s) Starting at: Ending at:

Duration

Start date: ☐ End date: ☒ No end date:

Summary

Description:

Demo: Creating the Subscription

The Schedule dialog shows the replicating agent schedule.

Synchronization Schedule
Specify the synchronization schedule for each agent.

Agent schedule:

Subscriber ▲	Agent Location	Agent Schedule
SANDY14\DEV12	Distributor	Occurs every day every 1 minute(s) bet... ▼

Help < Back Next > Finish >>| Cancel

We chose to initialize the subscription (apply the snapshot) immediately.

Initialize Subscriptions
Specify whether to initialize each subscription with a snapshot of the publication data and schema.

Subscription properties:

Subscriber ▲	Initialize	Initialize When
SANDY14\DEV12	<input checked="" type="checkbox"/>	Immediately ▼

A subscription database needs to be initialized with a snapshot of the publication data and schema unless it has already been specially prepared for the subscription.

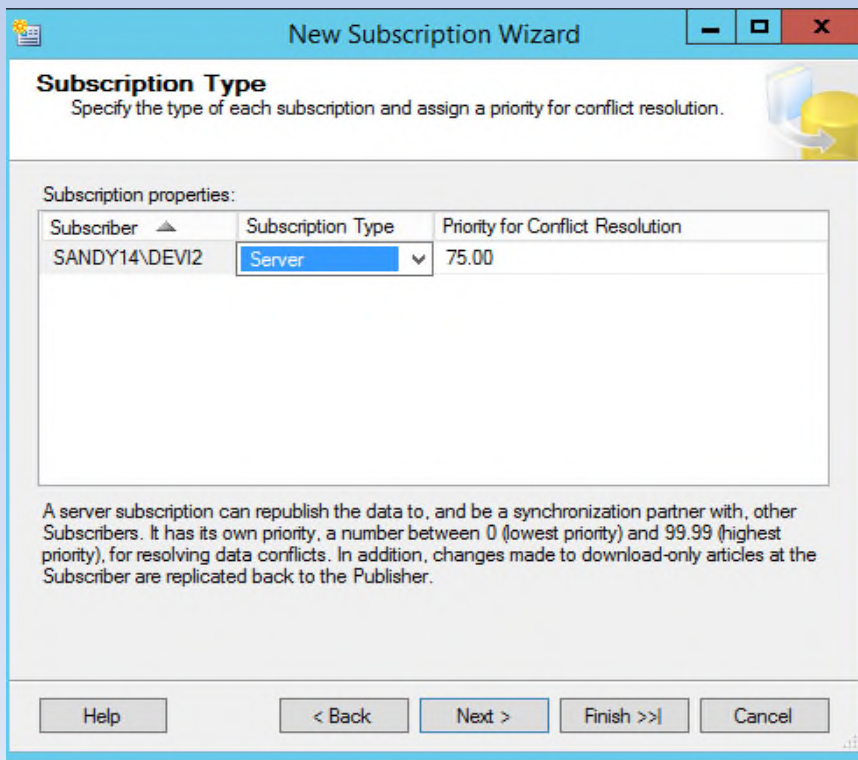
Help < Back Next > Finish >>| Cancel

Demo: Creating the Subscription

Subscription Type may be

- Server (May assign priority values to individual subscribers. Publisher priority is 100.)
- Client (First change written to Publisher wins.)

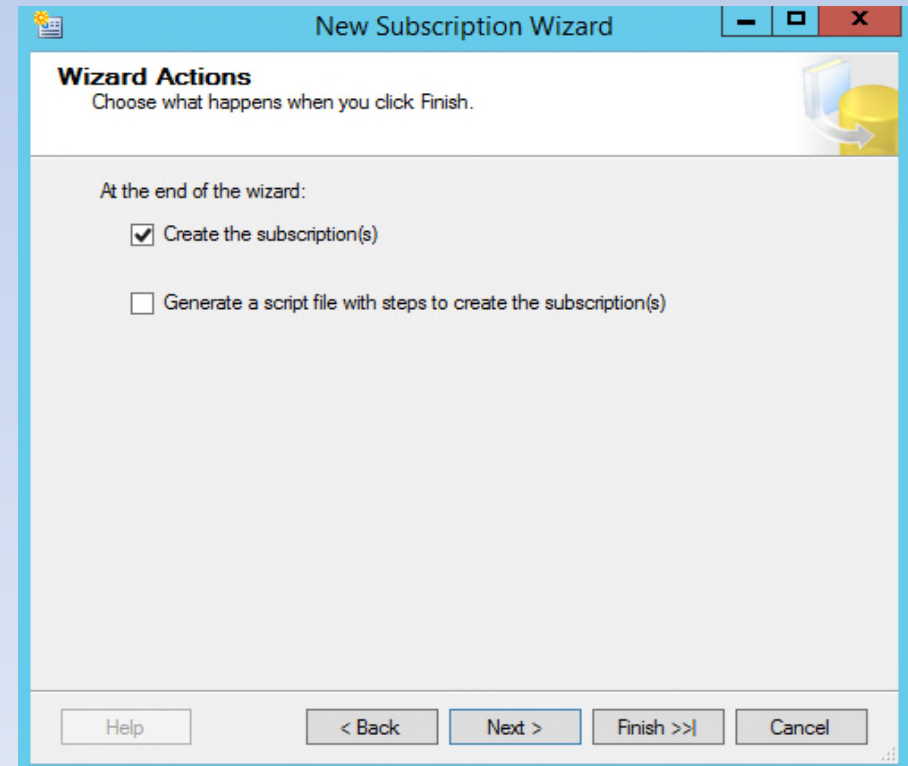
We chose to create the subscription.



The screenshot shows the 'New Subscription Wizard' window, specifically the 'Subscription Type' step. The title bar reads 'New Subscription Wizard'. Below the title bar, the section 'Subscription Type' is followed by the instruction 'Specify the type of each subscription and assign a priority for conflict resolution.' A table titled 'Subscription properties:' contains the following data:

Subscriber	Subscription Type	Priority for Conflict Resolution
SANDY14\DEVI2	Server	75.00

Below the table, a paragraph explains: 'A server subscription can republish the data to, and be a synchronization partner with, other Subscribers. It has its own priority, a number between 0 (lowest priority) and 99.99 (highest priority), for resolving data conflicts. In addition, changes made to download-only articles at the Subscriber are replicated back to the Publisher.' At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish >>|', and 'Cancel'.



The screenshot shows the 'New Subscription Wizard' window, specifically the 'Wizard Actions' step. The title bar reads 'New Subscription Wizard'. Below the title bar, the section 'Wizard Actions' is followed by the instruction 'Choose what happens when you click Finish.' Under the heading 'At the end of the wizard:', there are two options:

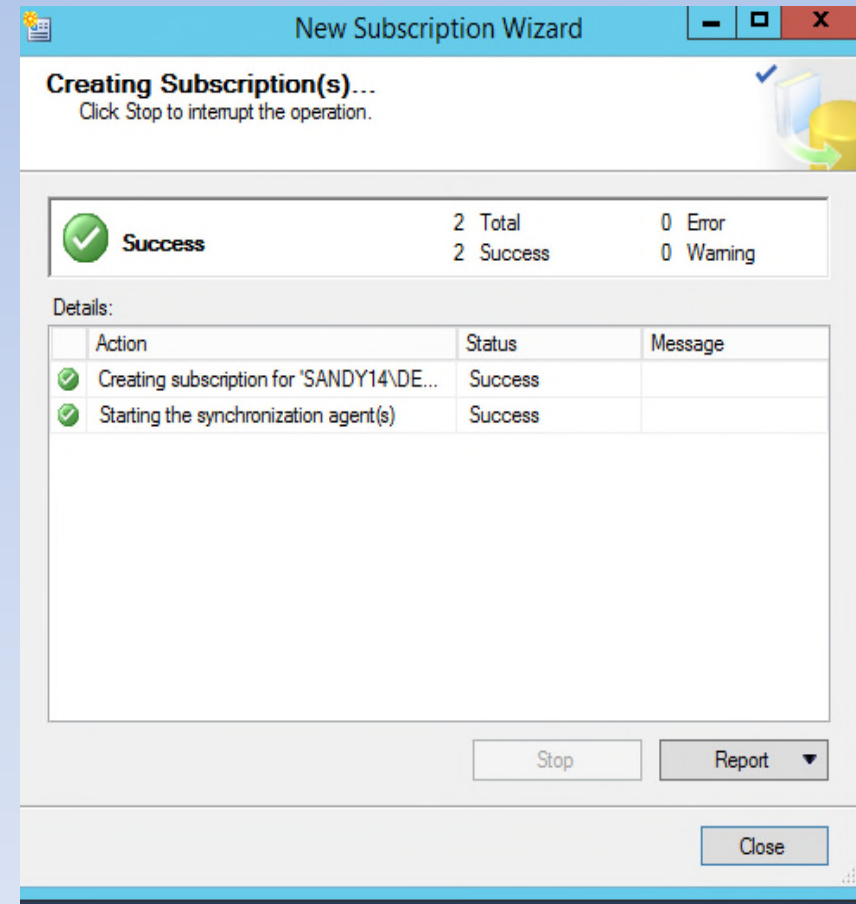
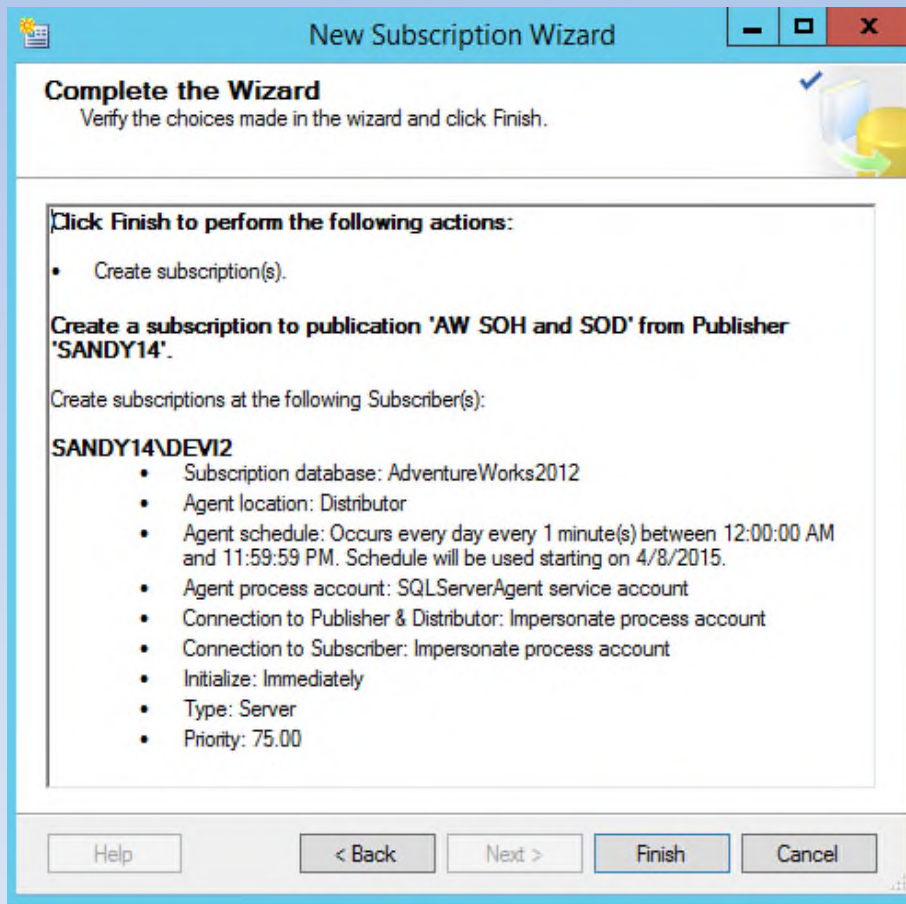
- ☒ Create the subscription(s)
- ☐ Generate a script file with steps to create the subscription(s)

At the bottom, there are buttons for 'Help', '< Back', 'Next >', 'Finish >>|', and 'Cancel'.

Demo: Creating the Subscription

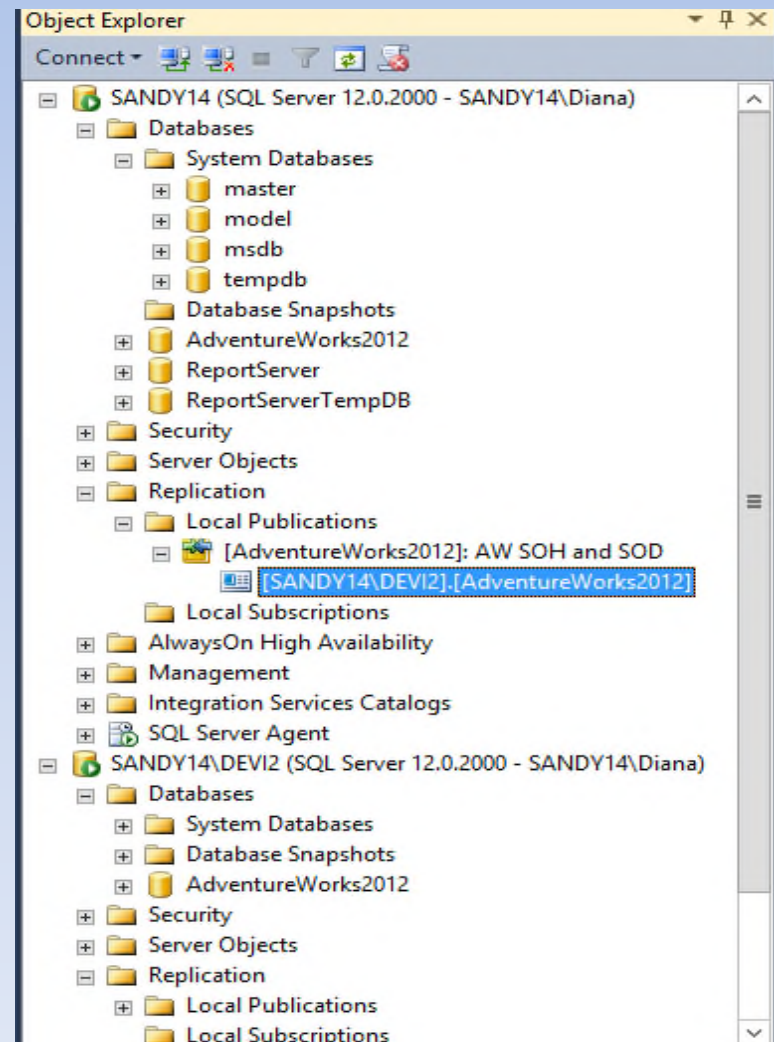
The actions to be performed by the Wizard are listed.

The Subscription is successfully created.



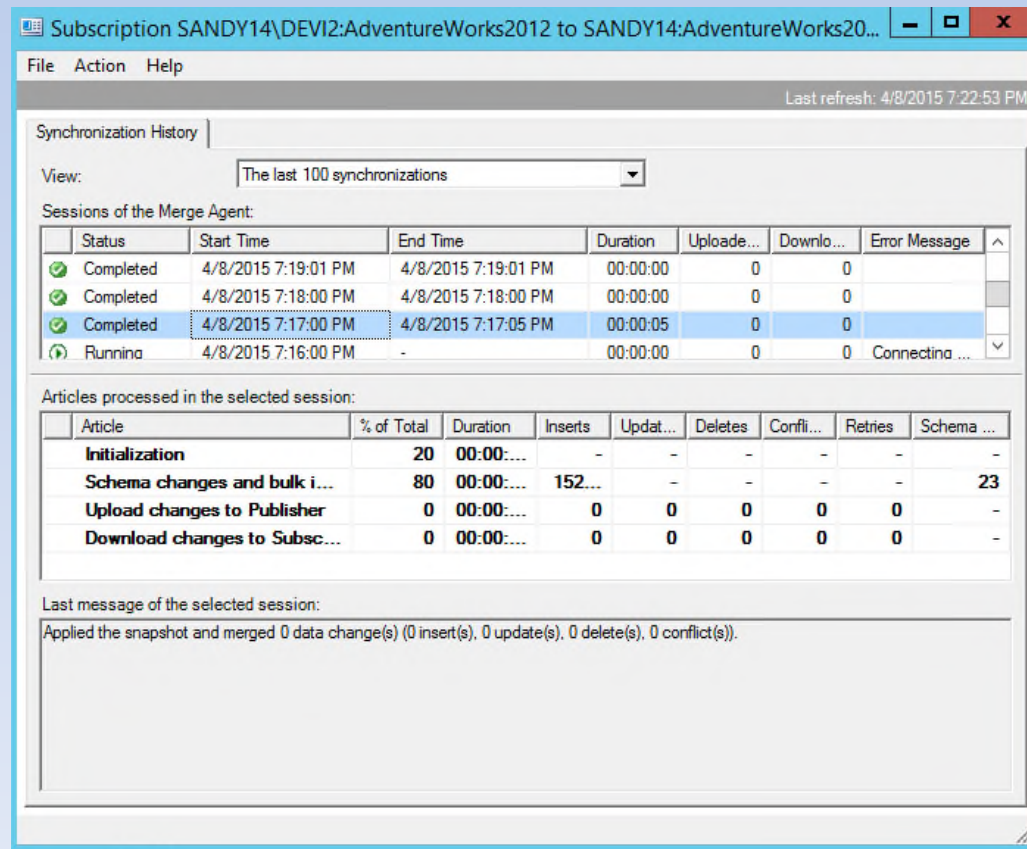
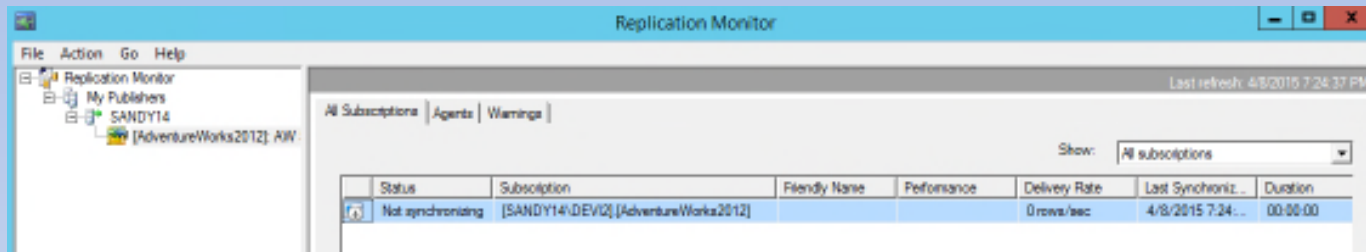
Demo: Creating the Subscription

The push subscription shows on the Publisher instance. (A pull subscription would show on the Subscriber instance.)



Demo: Creating the Subscription

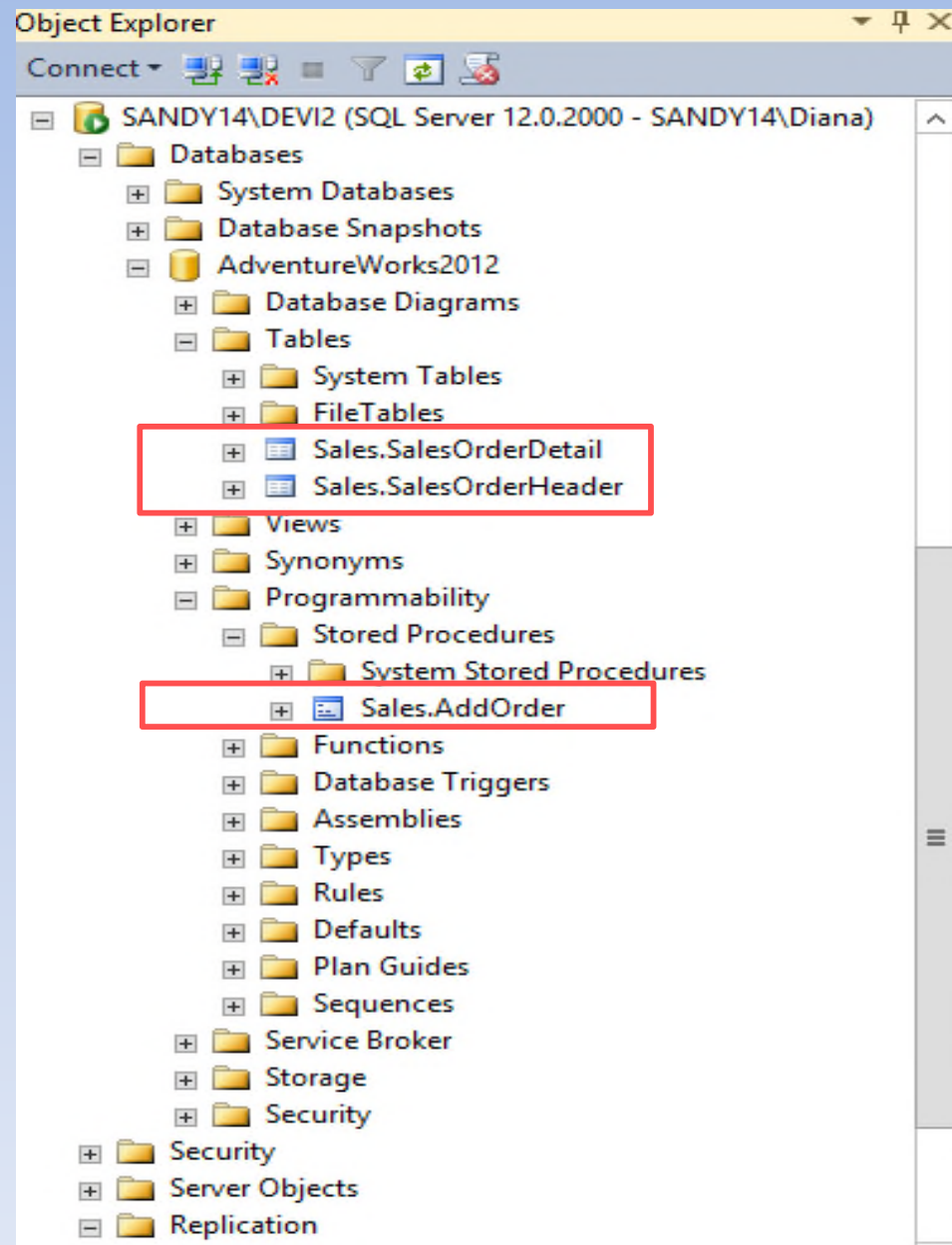
Replication Monitor shows that the Subscriber has been synchronized. (Double-click on the Subscription to see the Synchronization History).



Demo: Creating the Subscription

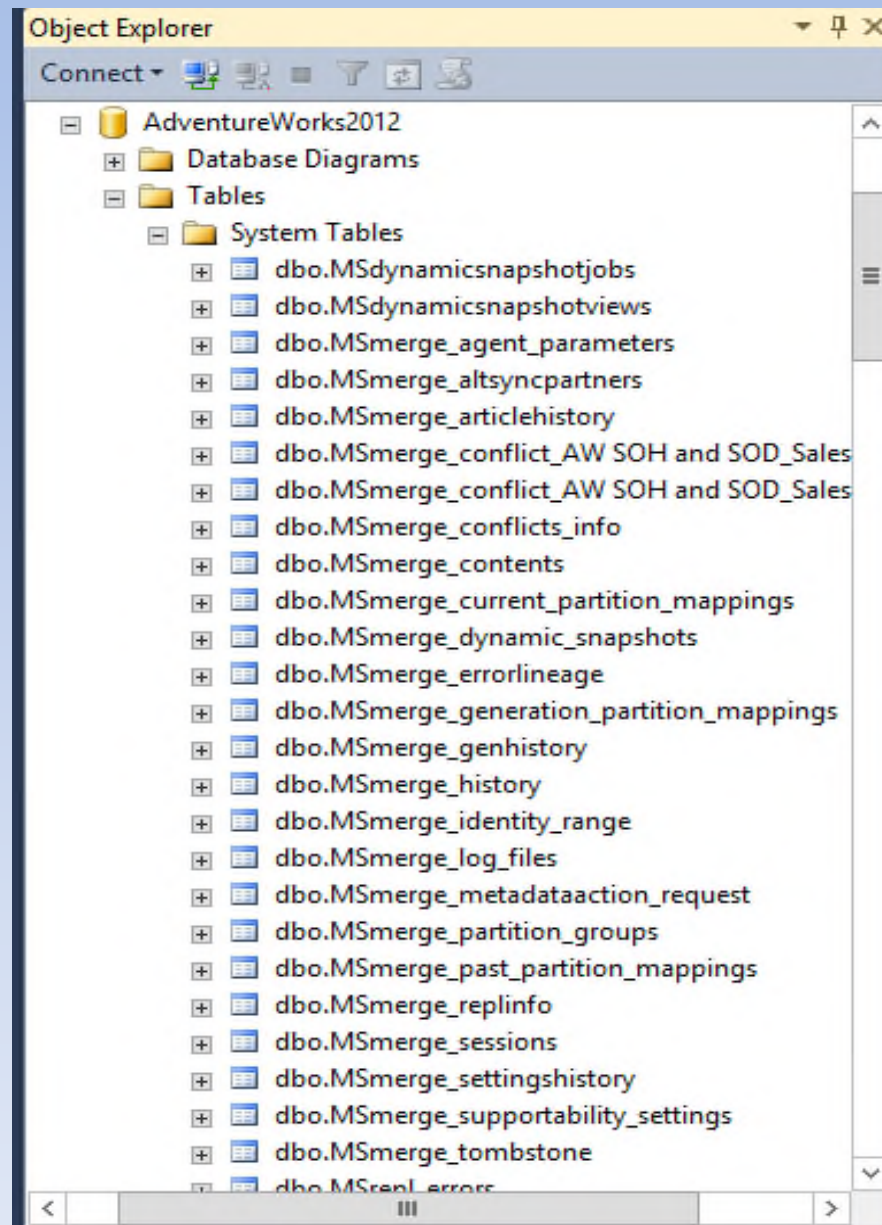
Object explorer shows that the Subscriber now has the replicated tables and stored procedure.

(User-defined data types used in SOH are also replicated.)



Demo: Creating the Subscription, concl.

Merge replication adds many system tables and triggers to track changes.



Demo: Replicating Data

Here are the latest rows in the SOH and SOD tables, before any additions are made. The max SalesOrderID IDENTITY value is **75123** and the max SalesOrderDetailID IDENTITY value is **121317**.

SQLQuery1.sql - S...ANDY14\Diana (51)) * Merge Repl.Add Or...NDY14\Diana (51))

```
USE AdventureWorks2012;
GO

SELECT TOP(5) * FROM Sales.SalesOrderHeader
ORDER BY SalesOrderID DESC;

SELECT TOP(8) * FROM Sales.SalesOrderDetail
ORDER BY SalesOrderDetailID DESC;

EXEC Sales.AddOrder;
```

100 %

Results Messages

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	PurchaseOrderNumber
1	75123	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75123	NULL
2	75122	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75122	NULL
3	75121	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75121	NULL
4	75120	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75120	NULL
5	75119	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75119	NULL

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	75123	121317	NULL	1	712	1	8.99	0.00	8.990000	73646D26-0461-450D-8019
2	75123	121316	NULL	1	879	1	159.00	0.00	159.000000	75A89C6A-C60A-47EA-8A5
3	75123	121315	NULL	1	878	1	21.98	0.00	21.980000	C18B6476-429F-4BB1-828E
4	75122	121314	NULL	1	712	1	8.99	0.00	8.990000	84F1C363-1C50-4442-BE16
5	75122	121313	NULL	1	878	1	21.98	0.00	21.980000	8CAD6675-18CC-4F47-8287
6	75121	121312	NULL	1	707	1	34.99	0.00	34.990000	AF25E491-73B2-4EB8-B2F1
7	75121	121311	NULL	1	930	1	35.00	0.00	35.000000	86638D4E-63C5-4014-AC4
8	75121	121310	NULL	1	921	1	4.99	0.00	4.990000	8B8A0C91-BF1A-4D8C-BFD

Query executed successfully. SANDY14 (12.0 RTM) SANDY14\Diana (51) AdventureWorks2012 00:00:00 13 rows

Demo: Replicating Data

After adding two orders on the Publisher, we see the SalesOrderID IDENTITY value of 75124 and 75125 for the added orders in SOH and some line items in the SOD table.

SQLQuery3.sql - S...ANDY14\Diana (51))* Merge Repl.Add Or...NDY14\Diana (51)) X

GO

```
SELECT TOP(5) * FROM Sales.SalesOrderHeader  
ORDER BY SalesOrderID DESC;  
  
SELECT TOP(8) * FROM Sales.SalesOrderDetail  
ORDER BY SalesOrderDetailID DESC;  
  
EXEC Sales.AddOrder;
```

100 %

Results Messages

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	Purcha
1	75125	5	2008-07-31 00:00:00.000	2008-08-07 00:00:00.000	2008-08-05 00:00:00.000	5	1	SO75125	NULL
2	75124	5	2008-07-31 00:00:00.000	2008-08-07 00:00:00.000	2008-08-05 00:00:00.000	5	1	SO75124	NULL
3	75123	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75123	NULL
4	75122	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75122	NULL
5	75121	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75121	NULL

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	75125	121320	NULL	1	921	1	4.99	0.00	4.990000	41644D0E-D313-4
2	75125	121319	NULL	1	929	1	29.99	0.00	29.990000	4A03B6D2-CD51-4
3	75124	121318	NULL	1	796	1	2181.5625	0.00	2181.562500	471D4F40-1870-4
4	75123	121317	NULL	1	712	1	8.99	0.00	8.990000	73646D26-0461-4
5	75123	121316	NULL	1	879	1	159.00	0.00	159.000000	75A89C6A-C60A-4
6	75123	121315	NULL	1	878	1	21.98	0.00	21.980000	C18B6476-429F-4
7	75122	121314	NULL	1	712	1	8.99	0.00	8.990000	84F1C363-1C50-4
8	75122	121313	NULL	1	878	1	21.98	0.00	21.980000	8CAD6675-18CC-4

Query executed successfully. SANDY14 (12.0 RTM) SANDY14\Diana (51) AdventureWorks2012 00:00:00 13 rows

Demo: Replicating Data

On the Subscriber, we see the replicated orders. After adding two orders on the Subscriber, we see the SalesOrderID and SalesOrderHeaderID values assigned from the Subscriber's IDENTITY value ranges.

SQLQuery3.sql - S...ANDY14\Diana (51))* Merge Repl.Add Or...NDY14\Diana (51))

```
USE AdventureWorks2012;
GO

SELECT TOP(5) * FROM Sales.SalesOrderHeader
ORDER BY SalesOrderID DESC;

SELECT TOP(8) * FROM Sales.SalesOrderDetail
ORDER BY SalesOrderDetailID DESC;

EXEC Sales.AddOrder;
```

100 %

Results Messages

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	Purcha
1	76125	5	2008-07-31 00:00:00.000	2008-08-07 00:00:00.000	2008-08-05 00:00:00.000	5	1	SO76125	NULL
2	76124	5	2008-07-31 00:00:00.000	2008-08-07 00:00:00.000	2008-08-05 00:00:00.000	5	1	SO76124	NULL
3	75125	5	2008-07-31 00:00:00.000	2008-08-07 00:00:00.000	2008-08-05 00:00:00.000	5	1	SO75125	NULL
4	75124	5	2008-07-31 00:00:00.000	2008-08-07 00:00:00.000	2008-08-05 00:00:00.000	5	1	SO75124	NULL
5	75123	3	2008-07-31 00:00:00.000	2008-08-12 00:00:00.000	2008-08-07 00:00:00.000	5	1	SO75123	NULL

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	76125	123320	NULL	1	708	1	34.99	0.00	34.990000	C86F2360-40C8-450
2	76125	123319	NULL	1	870	1	4.99	0.00	4.990000	BF5513B0-A12B-4E
3	76124	123318	NULL	1	957	1	2384.07	0.00	2384.070000	44932A59-2E68-4D
4	75125	121320	NULL	1	921	1	4.99	0.00	4.990000	41644D0E-D313-40
5	75125	121319	NULL	1	929	1	29.99	0.00	29.990000	4A03B6D2-CD51-46
6	75124	121318	NULL	1	796	1	2181.5...	0.00	2181.562500	471D4F40-1870-47
7	75123	121317	NULL	1	712	1	8.99	0.00	8.990000	73646D26-0461-45
8	75123	121316	NULL	1	879	1	159.00	0.00	159.000000	75A89C6A-C60A-47

Query executed successfully.

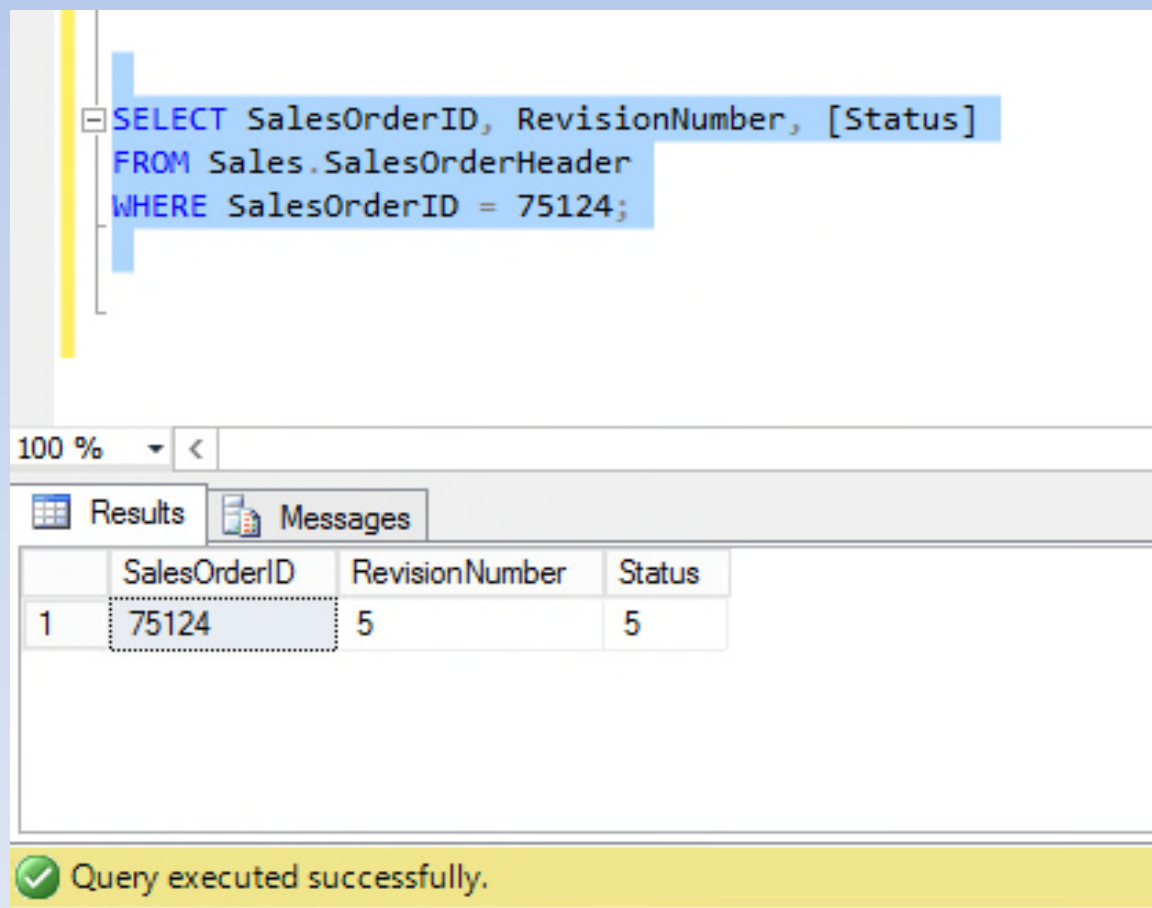
SANDY14\DEV12 (12.0 RTM) SANDY14\Diana (51) AdventureWorks2012 00:00:00 13 rows

Demo: Conflict Resolution

- We have set conflict resolution to column-level tracking, which allows for update of different fields in the same row during the replication interval.
- If the replication is set to *immediate*, and there is no likelihood of simultaneous update or delete targeting the same row on two instances, then row-level tracking should be chosen, as there will be fewer tracking tables required.
- We will show that with column-level tracking, updating two fields in the same row on different instances is not a conflict.
- Then we will update the same field on two instances, showing how the conflict is automatically resolved, and how the administrator can later change the conflict resolution.

Demo: Conflict Resolution

Here are the values of RevisionNumber and Status in one of the rows in SOH:




The screenshot displays the SQL Server Enterprise Manager interface. The query editor at the top shows the following SQL query:

```
SELECT SalesOrderID, RevisionNumber, [Status]
FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 75124;
```

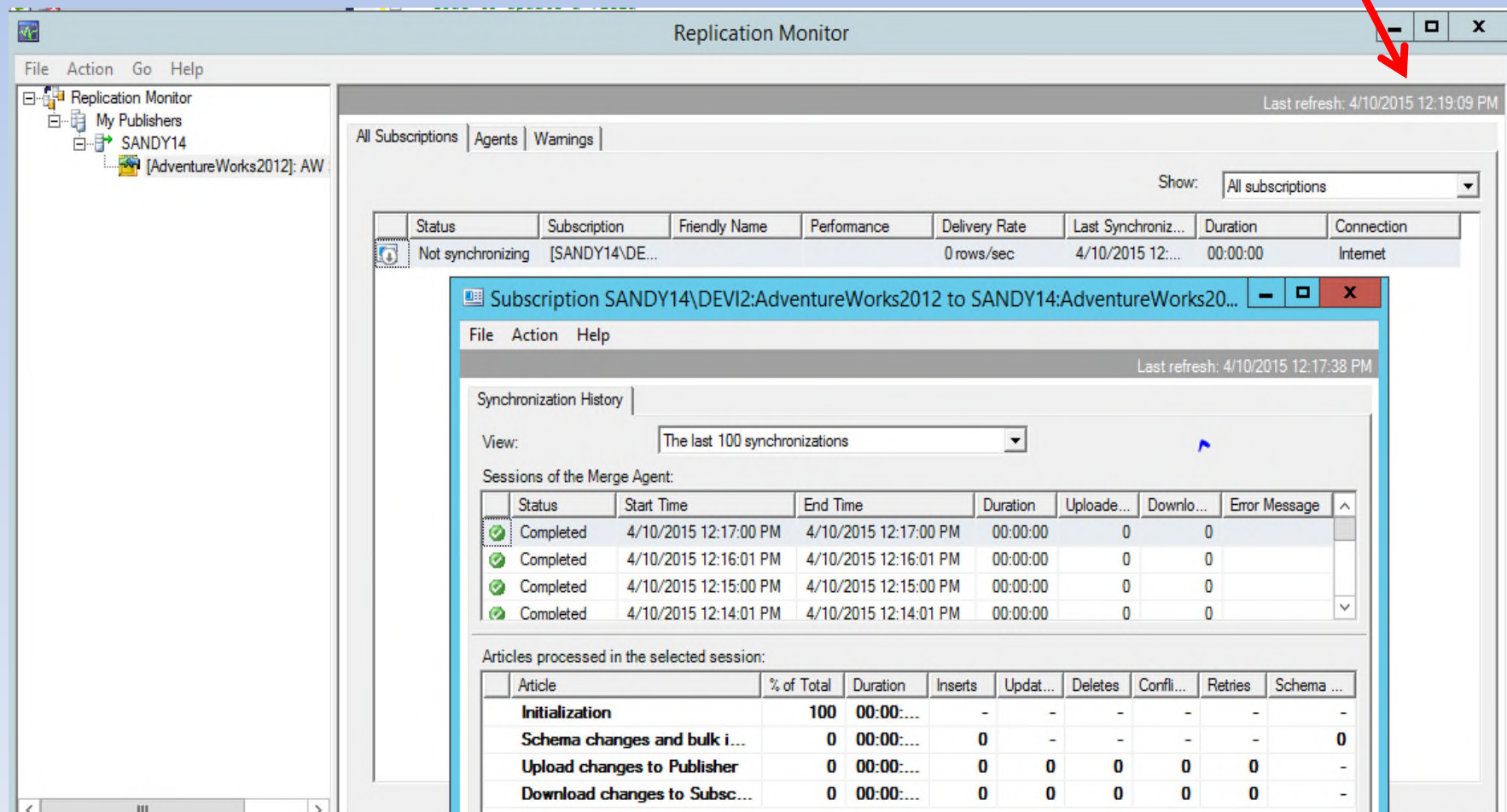
Below the query editor, the 'Results' tab is active, showing a single row of data. The columns are SalesOrderID, RevisionNumber, and Status. The values for the row are 75124, 5, and 5 respectively.

	SalesOrderID	RevisionNumber	Status
1	75124	5	5

At the bottom of the window, a yellow status bar indicates:  Query executed successfully.

Demo: Conflict Resolution

We wait for Replication Monitor to indicate that the one-minute replication interval is just beginning.



The screenshot shows the Replication Monitor application. The main window displays a subscription named 'SANDY14\DEV12:AdventureWorks2012 to SANDY14:AdventureWorks20...' with a status of 'Not synchronizing'. A red arrow points to the window's title bar. An inset window titled 'Subscription SANDY14\DEV12:AdventureWorks2012 to SANDY14:AdventureWorks20...' shows the 'Synchronization History' for the selected subscription. The history table shows four completed sessions, each with a duration of 00:00:00 and no errors.

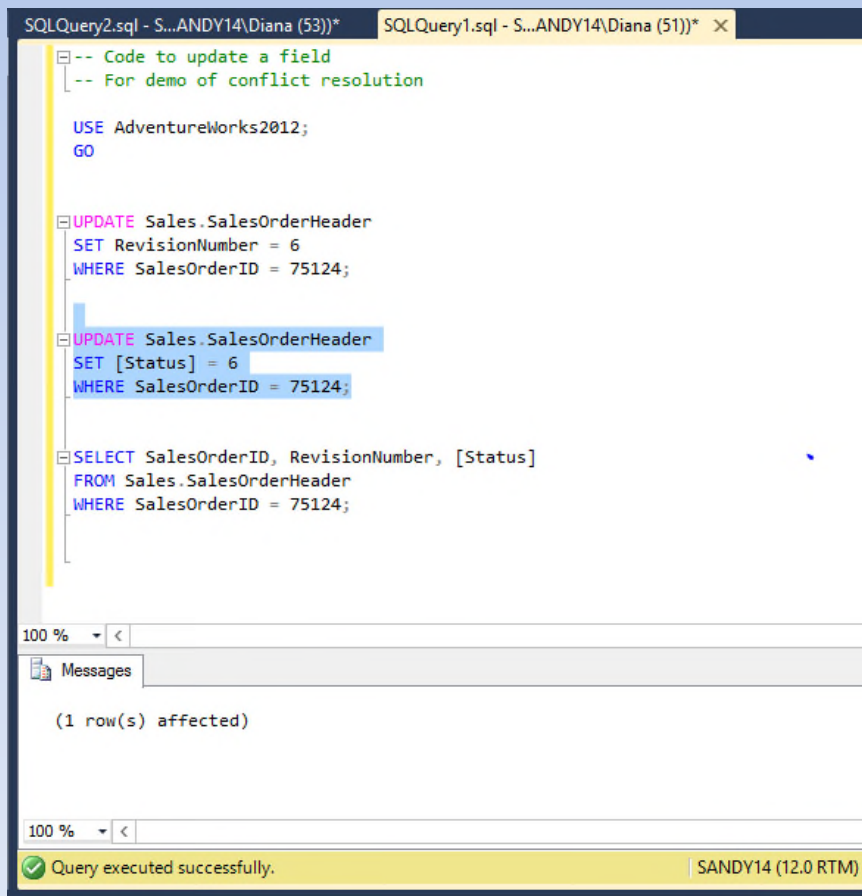
Status	Subscription	Friendly Name	Performance	Delivery Rate	Last Synchroniz...	Duration	Connection
Not synchronizing	[SANDY14\DE...			0 rows/sec	4/10/2015 12:...	00:00:00	Internet

Status	Start Time	End Time	Duration	Uploade...	Downlo...	Error Message
Completed	4/10/2015 12:17:00 PM	4/10/2015 12:17:00 PM	00:00:00	0	0	
Completed	4/10/2015 12:16:01 PM	4/10/2015 12:16:01 PM	00:00:00	0	0	
Completed	4/10/2015 12:15:00 PM	4/10/2015 12:15:00 PM	00:00:00	0	0	
Completed	4/10/2015 12:14:01 PM	4/10/2015 12:14:01 PM	00:00:00	0	0	

Article	% of Total	Duration	Inserts	Updat...	Deletes	Confi...	Retries	Schema ...
Initialization	100	00:00:...	-	-	-	-	-	-
Schema changes and bulk i...	0	00:00:...	0	-	-	-	-	0
Upload changes to Publisher	0	00:00:...	0	0	0	0	0	-
Download changes to Subsc...	0	00:00:...	0	0	0	0	0	-

Demo: Conflict Resolution

Within the next minute, we perform two updates, one on each instance. In the same row, the Publisher updates [Status]; the Subscriber updates RevisionNumber.



The screenshot shows a SQL Server Enterprise Manager window for the 'S...ANDY14\Diana (53)' instance. The 'SQLQuery2.sql' file is open, displaying the following T-SQL code:

```
-- Code to update a field
-- For demo of conflict resolution

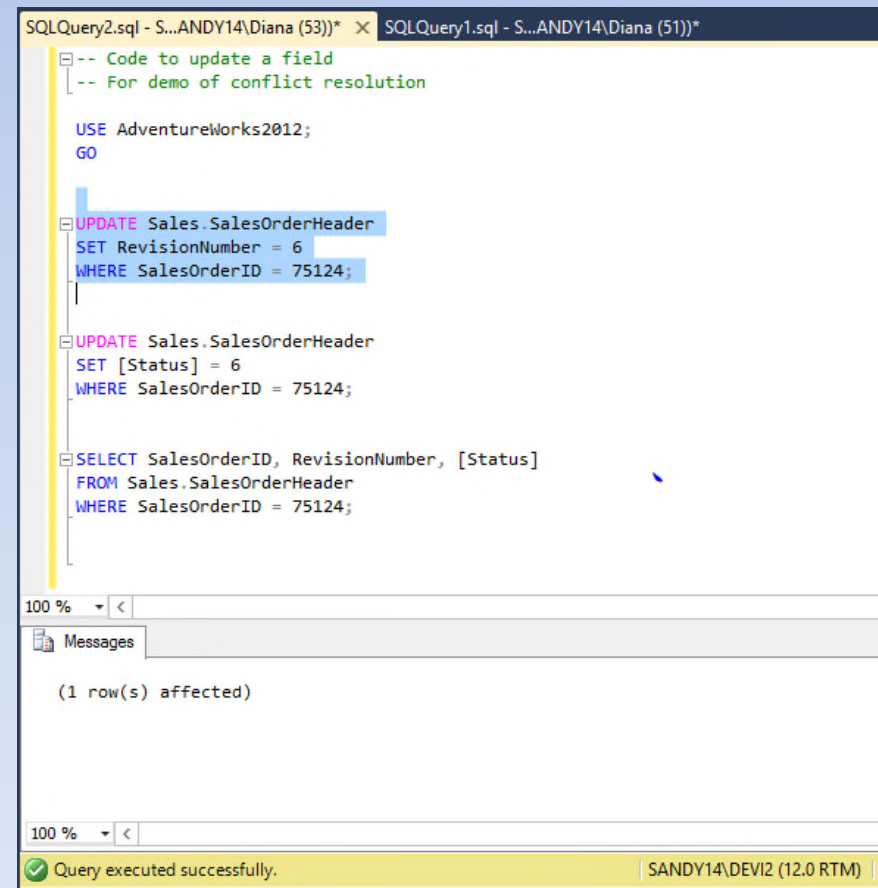
USE AdventureWorks2012;
GO

UPDATE Sales.SalesOrderHeader
SET RevisionNumber = 6
WHERE SalesOrderID = 75124;

UPDATE Sales.SalesOrderHeader
SET [Status] = 6
WHERE SalesOrderID = 75124;

SELECT SalesOrderID, RevisionNumber, [Status]
FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 75124;
```

The 'Messages' pane at the bottom shows '(1 row(s) affected)'. The status bar at the bottom indicates 'Query executed successfully.' and 'SANDY14 (12.0 RTM)'.



The screenshot shows a SQL Server Enterprise Manager window for the 'S...ANDY14\Diana (51)' instance. The 'SQLQuery1.sql' file is open, displaying the following T-SQL code:

```
-- Code to update a field
-- For demo of conflict resolution

USE AdventureWorks2012;
GO

UPDATE Sales.SalesOrderHeader
SET RevisionNumber = 6
WHERE SalesOrderID = 75124;

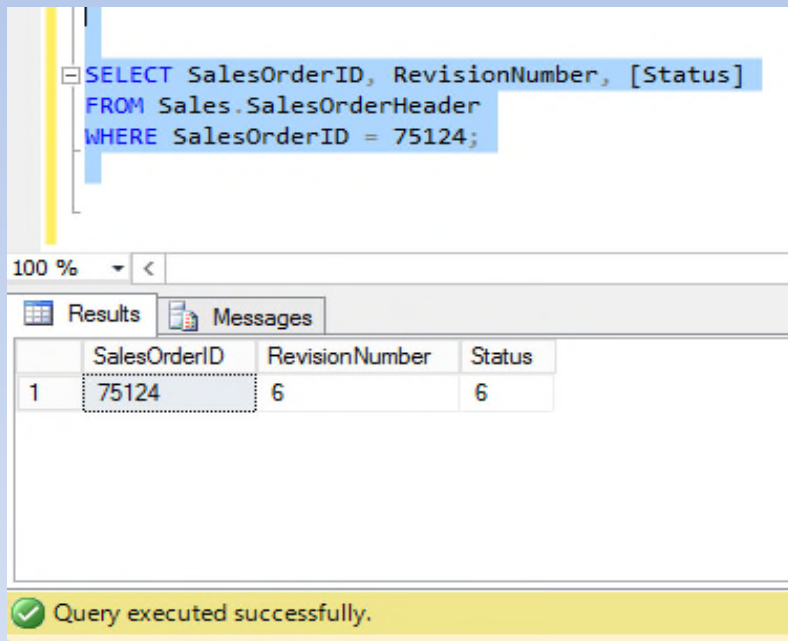
UPDATE Sales.SalesOrderHeader
SET [Status] = 6
WHERE SalesOrderID = 75124;

SELECT SalesOrderID, RevisionNumber, [Status]
FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 75124;
```

The 'Messages' pane at the bottom shows '(1 row(s) affected)'. The status bar at the bottom indicates 'Query executed successfully.' and 'SANDY14\DEV12 (12.0 RTM)'.

Demo: Conflict Resolution

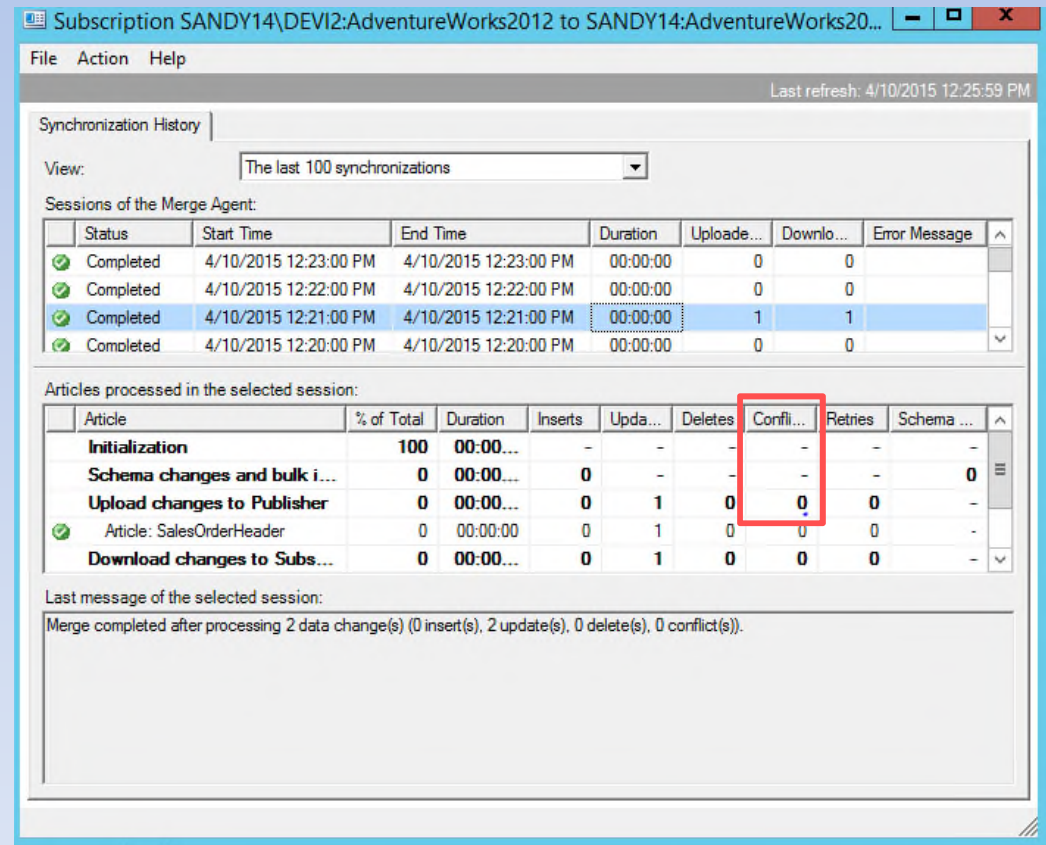
The results are merged, as column-level tracking is being used by merge replication. Replication Monitor shows no conflict.



```
SELECT SalesOrderID, RevisionNumber, [Status]
FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 75124;
```

	SalesOrderID	RevisionNumber	Status
1	75124	6	6

Query executed successfully.



Subscription SANDY14\DEV12:AdventureWorks2012 to SANDY14:AdventureWorks20...

File Action Help

Last refresh: 4/10/2015 12:25:59 PM

Synchronization History

View: The last 100 synchronizations

Sessions of the Merge Agent:

Status	Start Time	End Time	Duration	Uploade...	Downlo...	Error Message
Completed	4/10/2015 12:23:00 PM	4/10/2015 12:23:00 PM	00:00:00	0	0	
Completed	4/10/2015 12:22:00 PM	4/10/2015 12:22:00 PM	00:00:00	0	0	
Completed	4/10/2015 12:21:00 PM	4/10/2015 12:21:00 PM	00:00:00	1	1	
Completed	4/10/2015 12:20:00 PM	4/10/2015 12:20:00 PM	00:00:00	0	0	

Articles processed in the selected session:

Article	% of Total	Duration	Inserts	Upda...	Deletes	Confl...	Retries	Schema ...
Initialization	100	00:00...	-	-	-	-	-	-
Schema changes and bulk i...	0	00:00...	0	-	-	-	-	0
Upload changes to Publisher	0	00:00...	0	1	0	0	0	-
Article: SalesOrderHeader	0	00:00:00	0	1	0	0	0	-
Download changes to Subs...	0	00:00...	0	1	0	0	0	-

Last message of the selected session:

Merge completed after processing 2 data change(s) (0 insert(s), 2 update(s), 0 delete(s), 0 conflict(s)).

Demo: Conflict Resolution

Next, during one replication interval, update RevisionNumber to 7 on the Publisher and to 5 on the Subscriber. The Publisher wins.

Query executed successfully.

```
SELECT SalesOrderID, RevisionNumber, [Status]
FROM Sales.SalesOrderHeader
WHERE SalesOrderID = 75124;
```

	SalesOrderID	RevisionNumber	Status
1	75124	7	6

Subscription SANDY14\DEVI2:AdventureWorks2012 to SANDY14:AdventureWorks201...

Last refresh: 4/10/2015 12:30:52 PM

Synchronization History

View: The last 100 synchronizations

Sessions of the Merge Agent:

Status	Start Time	End Time	Duration	Uploade...	Downlo...	Error Message
Completed	4/10/2015 12:30:00 PM	4/10/2015 12:30:00 PM	00:00:00	0	0	
Completed	4/10/2015 12:29:00 PM	4/10/2015 12:29:01 PM	00:00:01	0	1	
Completed	4/10/2015 12:28:00 PM	4/10/2015 12:28:00 PM	00:00:00	0	0	
Completed	4/10/2015 12:27:00 PM	4/10/2015 12:27:00 PM	00:00:00	0	0	

Articles processed in the selected session:

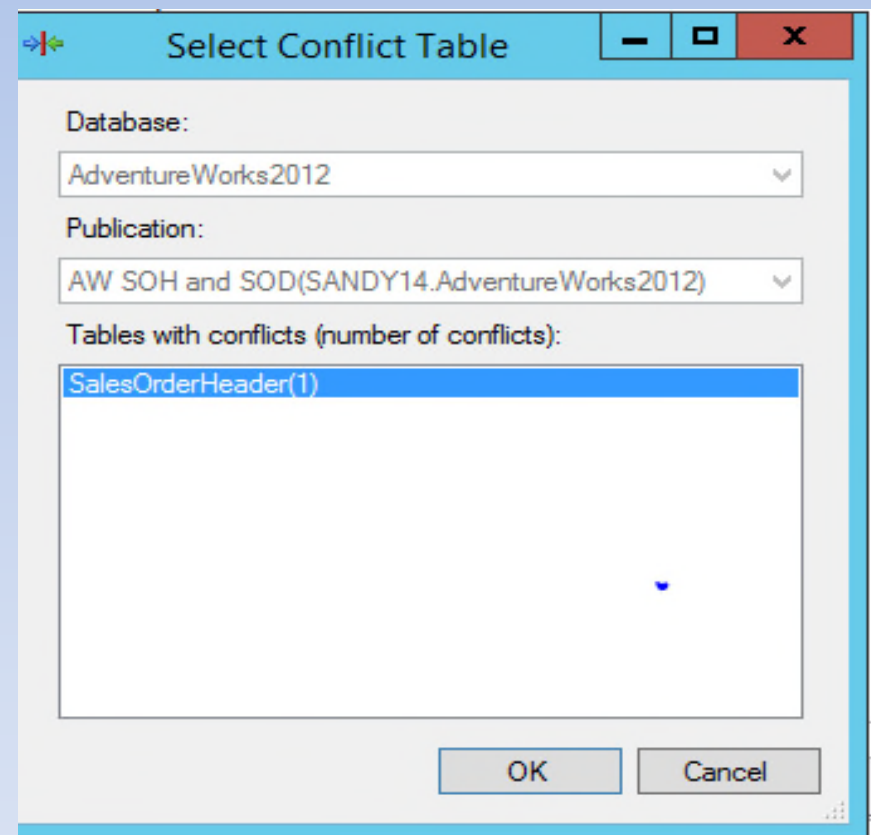
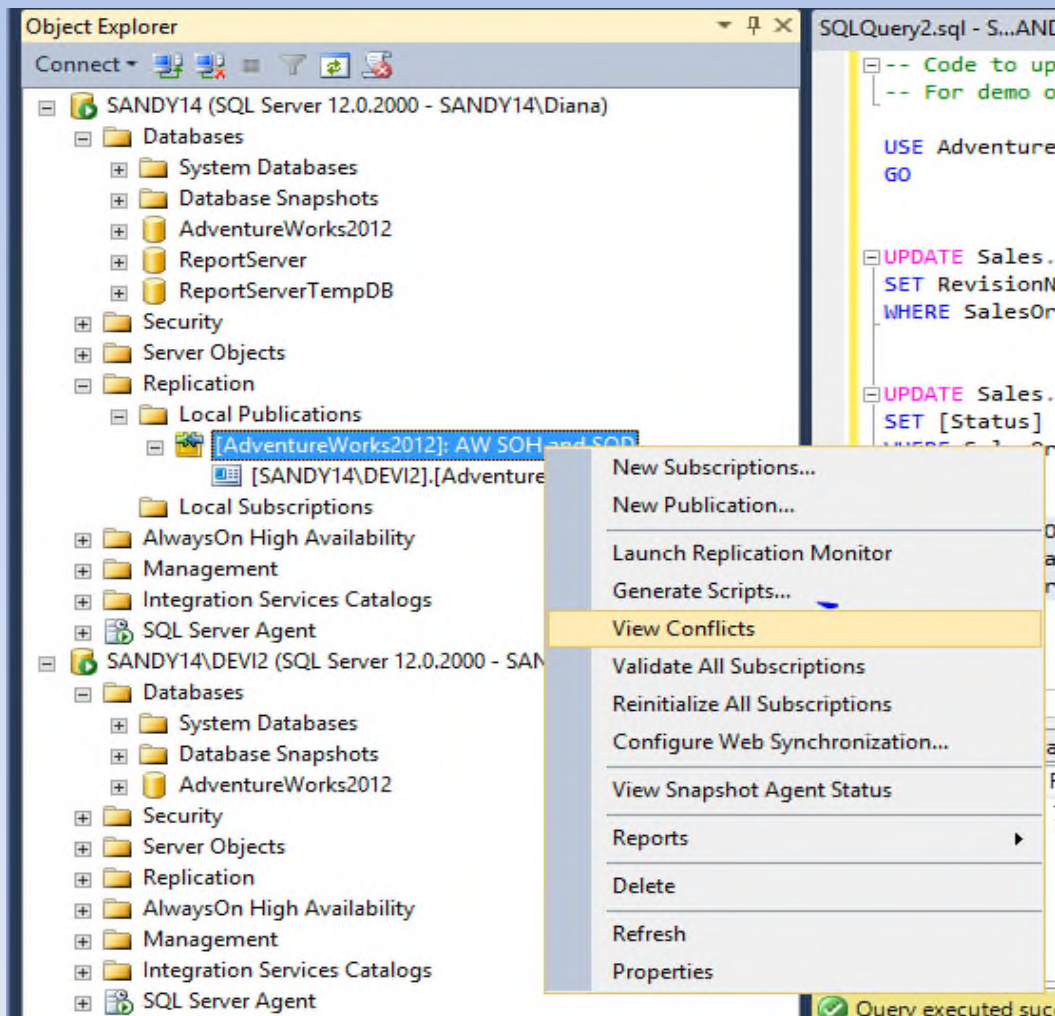
Article	% of Total	Duration	Inserts	Updates	Deletes	Conflicts	Retries	Schema...
Initialization	100	00:00:...	-	-	-	-	-	-
Schema changes and bulk ...	0	00:00:...	0	-	-	-	-	0
Upload changes to Publisher	0	00:00:...	0	0	0	0	0	-
Download changes to Subs...	0	00:00:...	0	1	0	1	0	-
Article: SalesOrderHeader	0	00:00:00	0	1	0	1	0	-

Last message of the selected session:

Merge completed after processing 1 data change(s) (0 insert(s), 1 update(s), 0 delete(s), 1 conflict(s)).

Demo: Conflict Resolution

To view the conflict, right-click the publication and then click View Conflicts. Select the conflict.



Demo: Conflict Resolution

The Conflict Viewer will open, showing the winning and losing values.

The screenshot shows the Microsoft Replication Conflict Viewer for 'AdventureWorks2012' -- '[Sales].[SalesOrderHeader]'. The interface includes a menu bar (File, View, Records, Help) and a toolbar with buttons for Select All, Select None, and Remove. The main area displays a table of conflicts with columns for Conflict loser, Conflict type, and Date. A single conflict is listed: SANDY14\DEV12.Advent... with a conflict type of 2(Column update conflict) and a date of 4/10/2015 12:29:01 PM. Below this, a detailed view of the conflict shows the same column(s) of the same row was updated at both 'SANDY14.AdventureWorks2012' and 'SANDY14\DEV12.AdventureWorks2012'. The resolver chose the update from 'SANDY14.AdventureWorks2012'. A table below shows the conflict details for SalesOrderID, RevisionNumber, OrderDate, DueDate, ShipDate, and Status, comparing the conflict winner and loser values. At the bottom, there are buttons for Submit Winner and Submit Loser, and a checkbox for Log the conflict details on remove. The status bar at the bottom indicates 'Connected to server 'SANDY14'', 'Filtered Conflicts: 1', and 'Total Conflicts: 1'.

Conflict loser	Conflict type	Date
SANDY14\DEV12.Advent...	2(Column update conflict)	4/10/2015 12:29:01 PM

The same column(s) of the same row was updated at both 'SANDY14.AdventureWorks2012' and 'SANDY14\DEV12.AdventureWorks2012'. The resolver chose the update from

Column name	Conflict winner	Conflict loser
SalesOrderID	75124	75124
RevisionNumber	7	5
OrderDate	7/31/2008 12:00:00 AM	7/31/2008 12:00:00 AM
DueDate	8/7/2008 12:00:00 AM	8/7/2008 12:00:00 AM
ShipDate	8/5/2008 12:00:00 AM	8/5/2008 12:00:00 AM
Status	6	6

☐ Log the conflict details on remove.

Submit Winner Submit Loser

Connected to server 'SANDY14' Filtered Conflicts: 1 Total Conflicts: 1

Demo: Conflict Resolution

Change the value to 6 under Conflict winner, then click the Submit Winner button. The conflict will be marked as resolved.

Microsoft Replication Conflict Viewer for 'AdventureWorks2012' -- '[Sales].[...]

File View Records Help

Database: AdventureWorks20 Publication: AW SOH and SOD Table: SalesOrderHeader

Conflict loser	Conflict type	Date
SANDY14\DEV12.Advent...	2(Column update conflict)	4/10/2015 12:29:01 PM

Select All
Select None
Remove

The same column(s) of the same row was updated at both 'SANDY14.AdventureWorks2012' and 'SANDY14\DEV12.AdventureWorks2012'. The resolver chose the update from

Column name	Conflict winner	Conflict loser
SalesOrderID	75124	75124
RevisionNumber	6	5
OrderDate	7/31/2008 12:00:00 AM	7/31/2008 12:00:00 AM
DueDate	8/7/2008 12:00:00 AM	8/7/2008 12:00:00 AM
ShipDate	8/5/2008 12:00:00 AM	8/5/2008 12:00:00 AM
Status	6	6

☐ Log the conflict details on remove.

Submit Winner Submit Loser

Connected to server 'SANDY14' Filtered Conflicts: 1 Total Conflicts: 1

Microsoft Replication Conflict Viewer

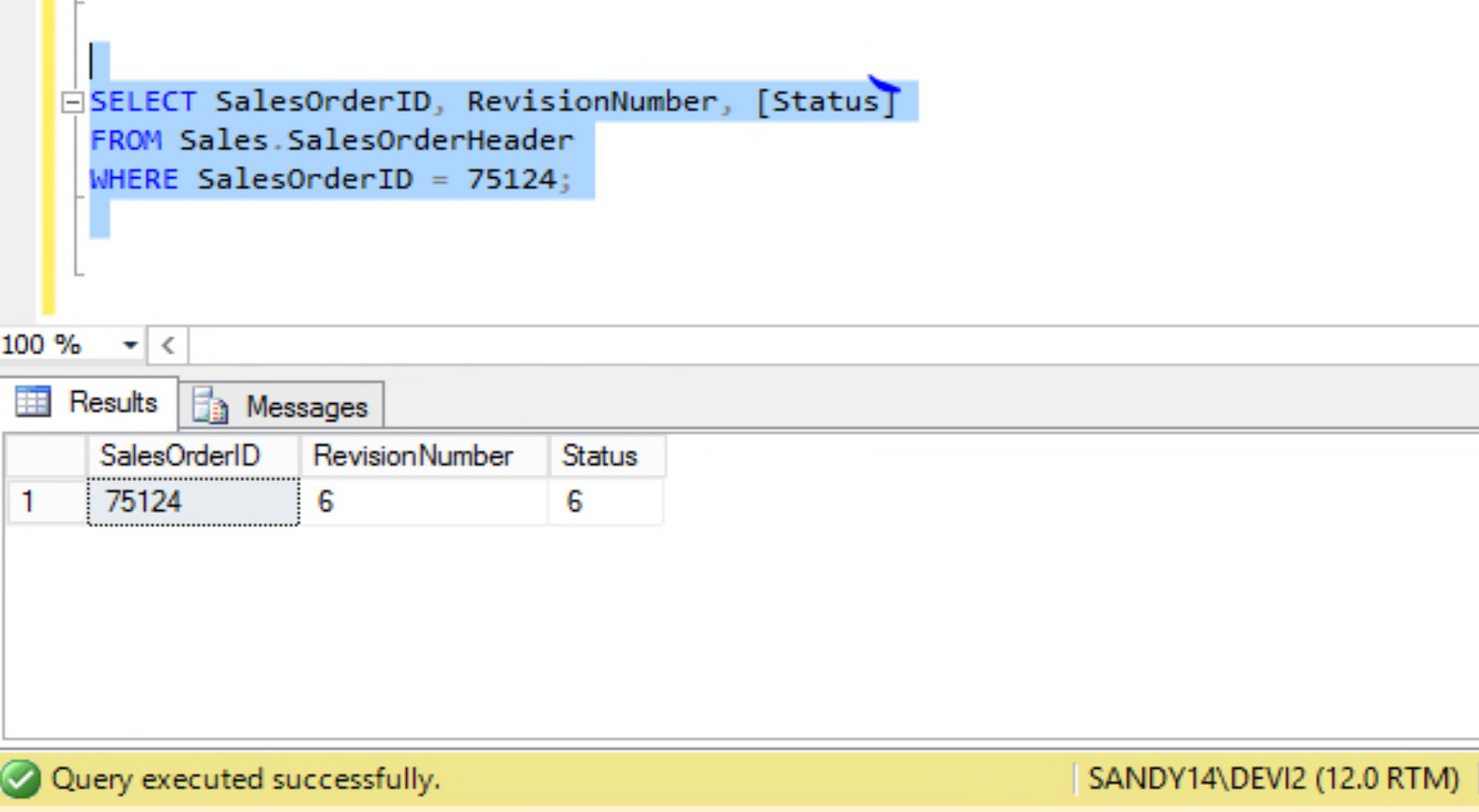
All conflicts in table 'SalesOrderHeader' have been removed.

☐ Do not show this message again

OK

Demo: Conflict Resolution

The result is propagated to the Publisher and Subscriber(s).



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL query: `SELECT SalesOrderID, RevisionNumber, [Status] FROM Sales.SalesOrderHeader WHERE SalesOrderID = 75124;`. The bottom pane shows the results of the query in a table with the following data:

	SalesOrderID	RevisionNumber	Status
1	75124	6	6

At the bottom of the interface, a yellow status bar indicates: "Query executed successfully." and "SANDY14\DEV12 (12.0 RTM)".

Why I Love Merge Replication

1. Merge replication is **resilient**. The publisher and/or one or more subscribers can be down for up to 14 days; when they come back online, all copies will synchronize.
2. Merge replication can provide **high availability**. If an instance goes down, point its clients to another replication partner instance.
3. The extra metadata required for merge replication will almost never be a “show stopper.”

Contact Info. and Further Reading

Presenter Diana Dee may be contacted at:

DrDianaDee@verizon.net

(Scripts and instructions for the demos will be provided on request.)

Further Reading:

SQL Server Books Online (BOL) (for whatever version you are using)

(Recommended BOL article: “Replication Agent Security Model”)

Stairway to SQL Server Replication

<http://www.sqlservercentral.com/stairway/72401/>