**Репликация баз данных**

Репликация – это не просто новомодное слово, это совершенно удобный и мощный инструмент в правильно отшлифованных руках. Некоторые считают, что репликация – это синоним синхронизации. Если заглянуть в ABBYY Lingvo, то среди возможных переводов слова replication ты не увидишь синхронизации, зато будут такие слова как: эхо, отражение, дублирование, повторение, размножение... Эти слова хорошо отражают данную технологию и то, что мы будем рассматривать сегодня.

**Издатель-дистрибутор-подписчик**

Чаще всего репликацию связывают с базами данных и мы в основном будем говорить о базах данных на примере MS SQL Server. Причем не только с классическими базами, но и такими специализированными, как **Active Directory**. Но на этом мир не перевернулся, репликацию можно удачно использовать и для простых файлов, главное правильный подход.

Репликация в MS SQL Server строиться на трех понятиях – **издатель**, **дистрибутор** и **подписчик**. Чтобы понять, что это означает, достаточно обратиться к нашей реальной жизни, где издатель выдает какую-то информацию дистрибутору, а тот рассылает ее подписчикам. Точно также и в компьютерной жизни. Но обо всем по порядку.

**Издатель** - хранит источник базы данных, делая опубликованные данные из таблиц базы данных доступными для репликации, находит и отправляет изменения дистрибьютору.

**Дистрибьютор** – это сервер, который содержит распределенную базу данных и хранит метаданные, историю данных и транзакции. Роль дистрибьютора может быть разной и зависит от типа развернутой репликации.

Дистрибьютор и издатель могут быть на одном компьютере. Чаще всего нет смысла выделять для каждого отдельный сервер, но для большой базы данных, и наиболее активных сайтов, можно располагать дистрибьютора на собственном сервере для оптимизации производительности.

**Подписчик** – владеет копией данных, и получает изменения произведенные издателем. В зависимости от настроек репликации, подписчик может иметь права изменять данные и реплицировать их обратно издателю для репликации другим подписчикам. Это называется **обновляющий подписчик**.

**Фильтруй базар**

Возможно, для публикации вам нужен поднабор таблицы как отдельной статьи. Это называется фильтрацией данных. Фильтрация данных позволяет избавиться от конфликтов репликации, когда несколько сайтов имеют право обновлять данные. Ты можешь фильтровать таблицы вертикально, горизонтально или смешанно для создания отфильтрованной порции данных.

Вертикальный фильтр содержит поднабор колонок таблицы. Только реплицированные колонки отображаются подписчику. Для примера, можно использовать вертикальный фильтр для публикации всех колонок кроме «Заработная плата» в таблице «Работники».

Горизонтальный фильтр содержит поднабор строк таблицы. Подписчик получает только этот поднабор строк. Если ненужно реплицировать информацию о левых доходах, то ее можно отфильтровать запросом.

Возможно подписание на публикацию с помощью Push или Pop метода. Метод Push обычно используется в приложениях, которые должны отправлять изменения подписчику как можно быстрее после изменения. Этот метод более предпочтителен для публикаций требующих высокую защищенность и где высокая загрузка процессора у дистрибьютора не влияет на производительность.

Метод Pop более подходит в публикациях с меньшей защищенностью и может поддерживать большое количество подписчиков, например подписчики Internet.

**Типы репликации**

Существует три основных типа репликации: **снимок**, **журнальный** и **смешение**. Тип репликации назначается каждой публикации. Таким образом, возможно использование нескольких типов репликации в одной базе данных.

Репликация снимка распределяет данные напрямую как отображение на определенный момент, без мониторинга изменений. Это самый простой тип, при котором происходит банальное копирование снимка всех или отфильтрованных данных. Можете уронить свой взгляд на этот тип в следующих случаях:

* данные изменяются существенно, но редко;
* подписчику требуются данные только для чтения;
* возможна большая задержка, потому что данные обычно только периодически обновляются;
* подписчику требуется автономность.

При репликации транзакций от источника к приемнику поступают только изменения. Агент мониторит изменения в журнале транзакций на изменение реплицированных данных и переносит необходимые записи дистрибьютору. Агент дистрибьютора отправляет изменения подписчику. Прежде чем этот тип начнет работать, подписчику отправляется полный снимок реплицированных таблиц, а затем подписчик получает только изменения.

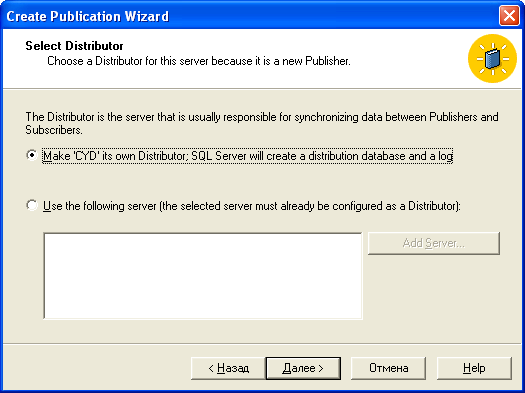
Репликация транзакций может использоваться там, где необходимо, чтобы подписчик получал изменения с минимальной задержкой.

Смешение - этот тип позволяет сайтам автономно изменять реплицированные данные. Позже, изменения с сайтов сливаются в одно целое. Этот тип не гарантирует целостности транзакций, но он гарантирует, что все сайты сливаются в один результирующий набор.

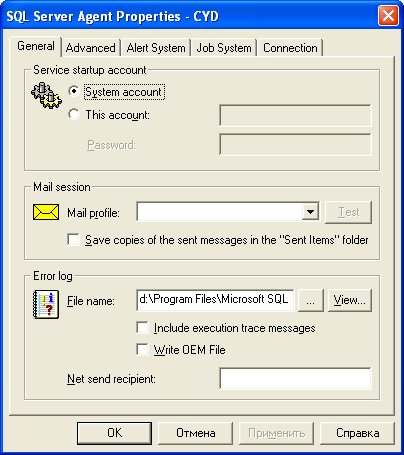
**Репликация MS SQL Server**

Очень удачно сделана возможность **репликации в MS SQL Server**. Настройка проста, как три копейки, потому что ее легко сделать с помощью двух мастеров, но есть подводные камни, о которых мастер не может рассказать, а мануалы просто умалчивают. Итак, давайте бегло пробежимся по процессу настройки репликации и сделаем упор на подводные булыжники, о которых все молчат как рыбы.

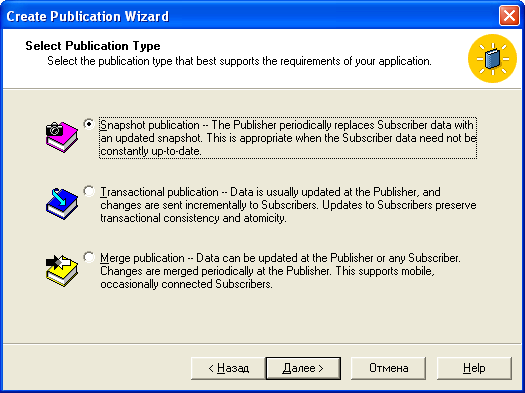
Для начала необходимо создать издателя и дистрибутора. Для этого на одном из серверов выбираем меню **Tools** | **Replication** | **Create and Manage Publication**. Я бы порекомендовал использовать для издателя машину помощнее. Первое, что у нас попросит мастер – выбрать базу данных. Выбираем, жмем **Create Publication**, и на следующем этапе сервер предложит создать дистрибутора. По умолчанию дистрибутором предлагается сделать ту же машину.

  
Мастер запрашивает создание дистрибутора репликации

Тут появляется первый подводный камень – если дистрибутор будет установлен на удаленно от издателя (на другой машине), то SQL Server Agent не может работать от имени системного аккаунта. Почему? Агент должен иметь возможность авторизоваться на машине дистрибутора и передать изменения, а для этого используется учетная запись, под которой работает агент. Под **Local Account** авторизоваться нигде не удастся, поэтому изменения никуда не пойдут.

  
Настройка учетной записи MS SQL Server Agent

После этого вам предложат сконфигурировать самого агента вручную или автоматически. Если выбрать ручной режим, то количество шагов мастера резко возрастает, но они просты и с минимальными знаниями английского ты разберешься в них. Если выбрать автомат, то остается только указать мастеру требуемый тип репликации – снимок (Snapshot publication), транзакции (Transaction publication) или смешение (Merge publication) и указать необходимые таблицы. Да, в репликации участвует не вся база, а указанные таблицы. Системные таблицы реплицировать незачем.

  
Выбор типа репликации

После создания издателя необходимо создать подписчика и настройку можно считать завершенной. Во время создания подписчика ты сможешь настроить план выполнения, указать дни, время или промежутки, через которые нужно выполнять репликацию.

Если ты настроил репликацию, и решил перенести базу данных на другой сервер, то можешь забыть про перенос через резервное копирование и восстановление. Дело в том, что в резервную копию не попадает информацию о репликации :(. Тут приходиться отключать базу Detach, копировать файлы на другой компьютер и подключать их заново Attach.

**Золотой ключик**

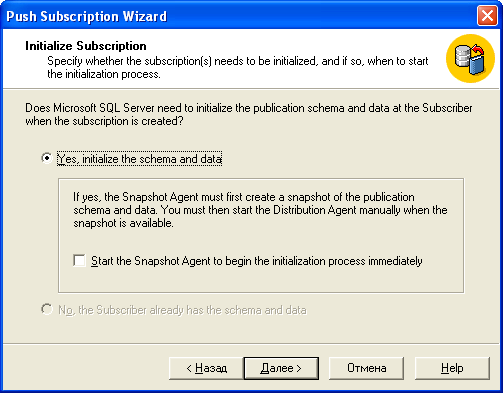
Следующая проблема, с которой ты можешь столкнуться – репликация ключевых полей. Если у тебя они имеют тип Guid, то никаких проблем тут не будет, но если Identity, то тебя ждут серьезные проблемы. Дело в том, что автоматически увеличиваемые поля не могут корректно реплицироваться с настройками по умолчанию, особенно при смешении, когда подписчик может изменять данные и должен уметь возвращать их издателю.

Допустим, что на двух компьютерах были созданы две разные записи с одинаковыми идентификаторами, что делать серверу? Какую из записей выбирать? По идее, в результирующую таблицу должно попасть обе записи, но изменять ID нельзя, особенно, если таблица связанная, а две записи с одинаковым ключом невозможны.

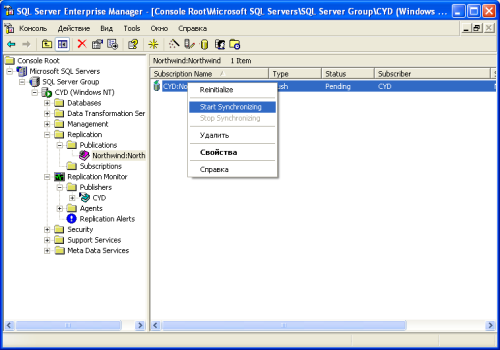
Проблема решается достаточно просто, нужно только выполнить следующие шаги:

* 1. Создаем копию базы данных издателя на компьютере подписчика.
* 2. Открываем окно редактирования таблицы или с помощью SQL запроса устанавливаем на издателе для ключа начальное значение 1, а для подписчика 1 000 000.

Все просто и красиво. Теперь, при добавлении записи на издателе новые записи будут нумероваться 1, 2, 3, 4..., а на подписчике 1000001, 1000002, 1000003.... Таким образом, записи пересекутся не скоро и конфликты могут никогда не появиться, особенно, если записи добавляются в таблицу не слишком интенсивно. Если же записи добавляются интенсивно, то откажись от автоувеличения и используй GUID поля в качестве ключа.

  
Запрос на копирование схемы

Но и это еще не все. При создании подписчика, тебе предложат перенести всю схему с издателя. Это удобно, если структура таблиц разная и их необходимо синхронизировать, но в нашем случае неприемлемо. Если ты поведешься на это предложение и ответишь Yes, то схема издателя будет скопирована подписчику и у обоих начальное значение ключа станет единицей, и все твои старания пойдут прахом, т.е. затрутся. Чтобы этого не произошло, жми No и наслаждайся, главное, чтобы на подписчике структура таблиц была такой же, как и у издателя.

  
Ручной запуск репликации

**Репликация Active Directory (AD)**

Active Directory, которая активно используется серверами Windows – это тоже база данных. Она может быть распределенной, когда в работе участвуют несколько серверов, и при этом, пользователь должен иметь возможность войти на любой из них с одним и тем же паролем. Как это сделать? Зарегистрироваться на каждом сервере в отдельности? Глупо и бессмысленно. И тут на помощь приходит репликация. Достаточно зарегистрироваться на одном сервере и прописать необходимые права доступа и вся информация будет реплицирована куда надо.

Репликация в AD происходит автоматически и чаще всего не требует особого вмешательства, но требует хорошего понимания основной идеи. Если бы с Active Directory было бы все так просто, то я не рассматривал бы эту технологию отдельно. Для начала нужно понимать, что для аутентификации используется протокол Kerberos и ответственность за подлинность берет на себя контроллер домена. Когда ты заходишь на сервер, то имя и пароль направляются серверу, который проверяет эти данные и в случае удачи выдает белый билет. Нет, билет конечно не белый, но на его основе пользователь получает те или иные права. Если есть желание и нет знаний, то советую поближе познакомиться с Active Directory и Kerberos, а наша задача – репликация.

Если до Windows 2000 в сети мог быть только один контроллер домена, который хранил все самое важное и управлял репликацией (тогда не было и Kerberos), то в нынешних версиях может быть несколько контроллеров. При этом все они будут равноправными. Это усложняет задачу по управлению процессом репликацию и разрешению возможных конфликтов, но окна нашли простое решение. Контроллеры домена наблюдают друг за другом, определяя, какой из них в данный момент будет дистрибутором, т.е. одарит других изменениями.

Настраивать вручную соединения между контроллерами доменов в сети нет необходимости, хотя и есть такая возможность. Сервера сами через определенные промежутки времени отслеживают доступные контроллеры и хранят в памяти всю необходимую информацию.

По умолчанию, репликация происходит каждые пятнадцать минут. Через эти промежутки времени сервер направляет контроллерам домена сообщения о том, что есть изменения и конечно же становиться дистрибутором. Остальные участники репликации, получив подобное сообщение, подключаются к серверу и вытягивают данные. Сам дистрибутор без запроса свои изменения в сеть не выплюнет, чтобы злые хакеры не смогли перехватить подобный пакет. Просто нет смысла без надобности кидать в сеть такие важные данные, вдруг остальные контроллеры домена упали (пусть земля им будет пухом и прахом).

**Конфликты в Active Directory**

Благодаря тому, что репликация выполняется с задержкой, сервер реплицирует обновления пачками. Все изменения в Active Directory накапливаются и в определенный момент рассылаются всем контроллерам домена. Это хорошо, но за счет задержки возможны и проблемы. Допустим, что в определенный промежуток времени, одновременно произошло изменение на двух контроллерах домена. Чьи изменения будут реплицированы? Давай попробуем разобраться.

Все объекты Active Directory имеют версию, которая при создании получает значение единицы. После каждого редактирования объекта версия увеличивается, поэтому если приходит просьба реплицировать запись с меньшим номером версии, чем текущая, то такие изменения откатываются.

А что если серверу придет одновременно два предложения на репликацию от разных контроллеров, и при этом версии объектов будут одинаковыми, но сами объекты разными? Такое может случиться, когда один и тот же объект с идентичной версией изменяется на разных контроллерах. Оба контроллера увеличат версию, и она будет снова одинаковой. В этом случае побеждает тот сервер и соответствующее изменение, которое было сделано последним.

Самый крайний случай – когда версии одинаковы и даже время изменения идентично. Конечно, вероятность этого слишком мала, но она есть, поэтому разработчики Active Directory в данном случае предпочли выбирать то изменение, которое пришло с сервера с большим глобальным идентификатором GUID. Конечно, это глупый выбор и может оказаться далеко не точным, но он хоть как-то решает конфликт.

Каждый контроллер, получив изменения, пытается втулить их другим контроллерам вашей сети. Тут тоже есть проблема - представим, что у нас три контроллера домена. Редактируем объект на первом и он, конечно же, должен уведомить остальных об изменения. Они забрали эти изменения, но в этот момент второй контроллер пытается эти же изменения впихнуть нам обратно или на третий домен, который уже забрал изменения. Что делать в этом случае? Все очень просто – у нас же есть версия изменений и по ней можно узнать, нужно забирать запись, или она уже реплицирована.

Эта проблема частично решается и тем, что репликация может пройти не дальше трех контроллеров. Если сервер получил изменения третьим, то дальше он уже никому его передавать не будет. Передача репликации по цепочки происходит, только если контроллер получил новую версию объекта первым или вторым.

**Репликация AD через 56к**

Реплицировать данные каждые 15 минут удобно и приятно, но только если все контроллеры домена связаны между собой высокоскоростным соединением. А что если два контроллера находятся в другом районе или деревне, где они могут быть подключены к общей сети только по DialUp? В этом случае, трафик репликации может отнять слишком много ресурсов, и полосы пропускания не хватит на решение других задач.

Чтобы этого избежать, можно и даже нужно разделить эти сервера на сайты. Все контроллеры, подключенные по высокоскоростной связи поместить в один сайт, а два удаленных в другой сайт. Внутри сайтов репликация может происходить по правилам, установленным по умолчанию, а вот между сайтами, можно настроить обмен так, чтобы не перегрузить полосу и оставить ее для передачи более важных данных. Такое разделение ты без проблем можешь настроить с помощью оснастки AD Sites and Services.

В качестве возможных вариантов сохранения трафика, в technet от MS предлагаются варианты:

* репликации данных по ночам;
* репликации в обеденные перерывы;
* репликации с большими промежутками времени.

Мне импонируют первые два варианта, особенно, если сайты находятся в одной временной зоне. Но если один расположен в Москве, а другой на Чукотке, то ты не то, что в обеденный перерыв не сможешь попасть, когда в Москве день, в Петропавловске уже полночь.

**Еще о репликации Active Directory (AD)**

Если хочешь узнать больше о репликации в Acive Directory, и нет проблем с английским, то рекомендую скачать следующий документ: www.certmag.com/bookshelf/C0617953.pdf. Это 92 страницы полезного и халявного чтива. Если и этого мало, то бегом на technet от Microsoft. Там информация изложена не так удобно и последовательно, но очень много хороших рекомендаций.

По Active Directory и репликации в частности могу посоветовать сайт only4gurus.com и конкретно ссылочка - http://www.only4gurus.com/v3/sitemap\_active\_directory.shtml. По репликации здесь можно найти хорошие презентации, рисунки которой были взяты за основу к данной статье. Я лишь перевел эти рисунки на мой родной русский и немного подкорректировал, чтобы они были нагляднее.

**Итого**

Я надеюсь, что я смог тебя убедить, что репликация – это не просто синхронизация, а более продвинутый и интеллектуальный шаг вперед. При правильном подходе, этот шаг будет большим. Если ты хорошо разберешься с этой темой, то без проблем сможешь даже сделать ручную репликацию так, где ее нет изначально. Ведь не во всех базах данных реализована такая возможность.

За кадром данной статьи осталась очень интересная тема – репликация Exchange сервера. У нее очень много похожего на Active Directory и SQL Server, но есть и интересные нюансы.