Изучаем OpenEdge Replication

(Учебное пособие)

Башкатов В.Г.

Санкт-Петербург, 2010 г

Изучаем OpenEdge Replication. (Учебное пособие.)

Материал предназначен для версии Progress OpenEdge 10.x и выше.

Все права сохранены. Автор оставляет за собой право на редактирование и распространение этого материала, а так же прочие права в соответствии с законом РФ «Об авторских и смежных правах».

© 2010 г. Башкатов В.Г.

Редактор: Мазина Ю.А.

ВВЕДЕНИЕ **5**

OE REPLICATION - ЧТО ЭТО ТАКОЕ? **6**

Терминология OE Replication 6
Файл свойств 8
Требования архитектуры 8
Сервер OE Replication 9
Агент репликации 10
Использование синхронной и асинхронной репликации 11
Обработка ошибочных ситуаций OE Replication 13
Синхронизация *13*
Server failure recovery *13*
Transition *13*
Особенности работы с базой данных 14
Целостность *14*
Доступность *14*
Режим ERO (Enhanced Read-Only) *14*
Блокировка схемы (schema lock) *14*
Принудительное усечение before-image файла source-базы данных *15*
Точки останова (quit points) target-базы данных *15*
Выполнение online backup *15*

ПЛАНИРОВАНИЕ **17**

OE Replication и After-imaging 17
Экстенты After-Image *18*
Расчет текущего объема After-image *19*
Использование информации об after-image из VST 19
Использование команды RFUTIL 19
Использование PROMON 20
Использование информации о before-image из VST 20
Репликация и After-Image экстенты *21*
Размеры After-image экстентов на Source-базе данных *21*
Расчет размера AI-экстента фиксированного размера 22
Использование экстентов переменного размера 23
Оценка сетевой пропускной способности OE Replication *23*
OE Replication и бизнес-требования 25
Подведение итога 27

РЕАЛИЗАЦИЯ **28**

Использование метода по умолчанию 28
Использование метода отложенного запуска агента 29
Настройка source-базы данных для OE Replication в offline 30
Настройка source-базы данных для OE Replication в online 31
Настройка target-базы данных 31
Настройка файла свойств 32
Настройка файла свойств для source-базы данных *32*
Настройка файла свойств для target-базы данных *33*
Запуск OE Replication 34
Старт source-базы данных *34*
Старт target-базы данных *34*
Журналирование запуска *35*
Процесс запуска и инициализации OE Replication *35*
Восстановление source и target баз данных 35
Доступ к базе данных с OE Replication 36
Доступ к target-базе 37
Доступ к source-базе 38
Останов OE Replication 38
Останов Source-базы *38*
Останов Target-базы *38*
Останов сервера OE Replication *39*
Останов агента OE Replication *39*
Очистка AI-экстентов на source-базе данных *39*
Нормальный режим работы OE Replication 40

Горячая копия базы данных....42
Условия обработки сбоев....42
Переход в target-базе данных....44
Одна или две?....44
Настройка автоматического перехода....44
Настройка ручного перехода....45
Процесс перехода....45
Ручной накат AI-экстентов....46
Активация OE Replication после перехода....47
Что такое Failback?....49
Как работает переход....50
Шаг 1: Первичная репликация до сбоя....52
Шаг 2: Сбой первичной машины....52
Шаг 3: Переход в состояние pre-transition....52
Шаг 4: Переход target-базы в source....53
Шаг 5: Failover....54
Шаг 6: Завершение восстановления первичной машины....56
Шаг 7: Начало перехода primary database....56
Шаг 8: Выполнение secondary replication....59
Шаг 9: Failback....60
Выполнение failback с использованием transition failover....60
Преимущества и недостатки использования transition failover для выполнения failback....63
Выполнение failback с использованием управляемого перехода....64
Достоинства и недостатки выполнения failback с использованием управляемого перехода....67
Восстановление после сбоя при переходе....67
Журнал выполнения transition....69
Свойства transition....70
Пример стартового файла параметров....73

МОНИТОРИНГ....75

DSRUTIL MONITOR....75
DSRUTIL STATUS....79

Введение

В простейшем случае репликация - это просто копирование данных из одной БД в другую. Но простейшие случаи, к сожалению, возникают редко, в результате чего репликация БД часто становится нетривиальной задачей. Проблема репликации – возможное возникновение конфликта данных, для разрешения которого необходима разработка специфических для каждой задачи алгоритмов. Продукт OpenEdge Replication (OE Replication) - это механизм баз данных OpenEdge Progress, который позволяет без особого труда организовывать репликацию между одной или двумя целевыми базами данных. В книге будет рассказано о том, как он работает, как настраивается, и как осуществляется мониторинг. В итоге вы увидите, что «не так страшен черт, как его малюют».

OE Replication - что это такое?

Копирование данных состоит из выполнения двух основных функций, работающих в режиме реального времени:

- копирование информации на одну и более машину;
- поддержка механизма failure recovery для обеспечения постоянной доступности данных.

OE Replication предназначен для автоматического копирования локальной базы данных в одну или более удаленных баз данных. Как только OE Replication установлен, настроен и запущен, процесс копирования начинает выполняться полностью автоматически.

Этот механизм фактически обеспечивает «горячим резервированием». В случае возникновения сбоя целевая база данных может автоматически стать основной, и, таким образом, пользователи всегда будут иметь доступ к необходимым данным.

В общем случае преимуществами OE Replication являются:

- доступность данных 24 часа в день 7 дней в неделю;
- минимальное или полное отсутствие потерь при возникновении сбоев и непредвиденных простоев.

Ключевыми характеристиками механизма являются:

- автоматическое копирование базы данных в реальном времени для обеспечения failover или disaster recovery;
- поддержка функции fail back;
- поддержка копирования исходной базы данных на одну или две целевые базы данных;
- продолжение работы исходной базы данных в процессе выполнения административных задач;
- возможность получения отчетов о выполнении копирования.

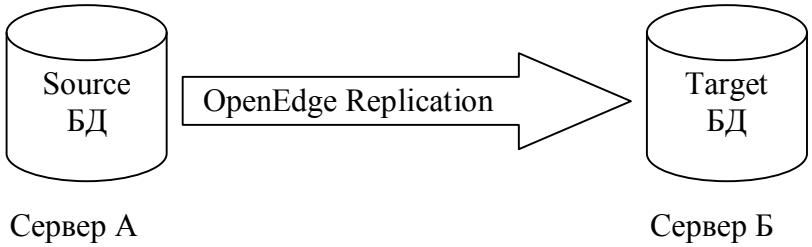
Терминология OE Replication

В этой части описана терминология, используемая OE Replication:

- primary и secondary базы данных;
- source и target базы данных;
- файл свойств OE Replication;
- сервер репликации;
- агент репликации.

База данных, изменяемая приложениями, называется Primary. Именно эта база данных при включении OE Replication будет называться Source-базой. База данных Secondary - это первоначальная копия базы данных Primary, которая при включении OE Replication будет называться Target-базой.

OE Replication обеспечивает копирование данных из одной базы данных в другую, поэтому копируемая база называется source, а база данных, в которую они копируются, называется target.



Source-база данных - это база, с которой работают пользователи, внося в нее изменения.

Характеристики source-базы данных:

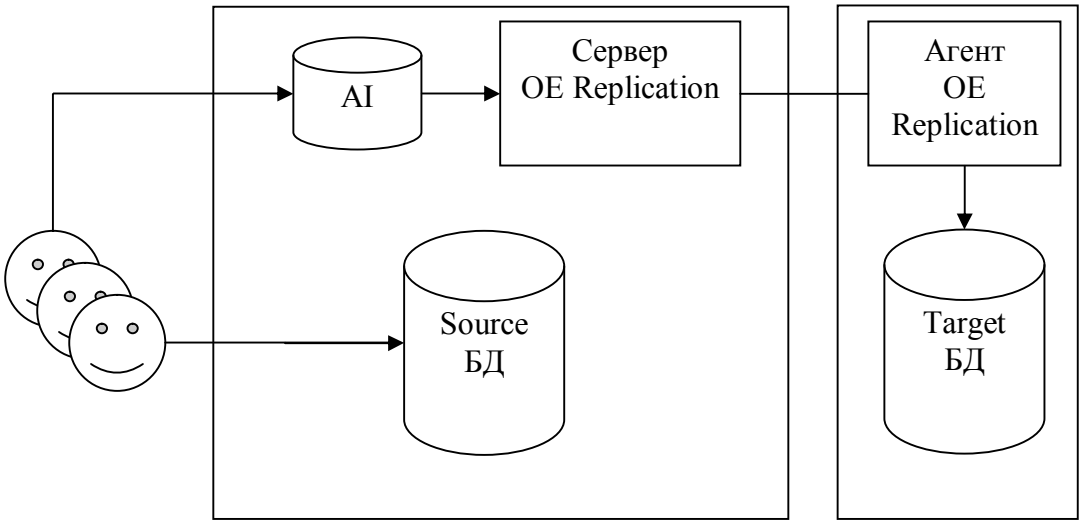
- Возможность получения доступа, как на чтение, так и на изменение данных.
- База данных не считается source, пока она не включена в механизм OE Replication как source-база.
- Сервер репликации выполняет копирование данных из этой базы в target-базу.
- Для работы механизма репликации должен быть активирован механизм After-imaging. Этот механизм позволит восстановить базу данных, если она окажется разрушенной, или будет повреждена область primary recovery (Before-image). При включении этого механизма менеджер базы данных начинает записывать любые изменения базы в качестве заметок в файл After-image (AI-файлы).
- Содержимое After-image файлов используется OE Replication для копирования на target-базу.

Target-база данных - это точная копия source-*баз*ы. Для обеспечения дополнительной избыточности, может быть создано две target-*баз*ы.

Характеристики target-базы данных:

- Формируется из source-базы и содержит те же данные, схему и логическую структуру.
- Корректируется исключительно только агентом OE Replication.
- Не может быть изменена никем и ничем, кроме как средствами OE Replication.

Модель работы OE Replication.



Примечание: для обеспечения полноценной работы OE Replication необходимо расположить source и target базы на разных машинах, что в случае возникновения сбоя на машине source-базы данных обеспечит возможность продолжения работы пользователей с target-базой.

Файл свойств

Настройки, используемые как source, так и target базами данных, хранятся в специальном файле свойств. Каждый такой файл содержит специфические настройки для каждой базы данных. Например, файл свойств source-базы содержит такие детали как имя source-базы данных; имена агента или агентов сервера репликации; тип перехода (transition), закрепленный за агентом; время ожидания агентом до начала выполнения операции transition после потери связи с сервером репликации. Так же в файле содержится информация об агентах и имени target-базы данных.

OE Replication поддерживает два различных типа файлов свойств. Это source.repl.properties и target.repl.properies. Оба файла расположены в каталоге $DLC/properties. Для использования этих файлов следует скопировать их в рабочий каталог базы данных, после чего можно приступать к редактированию. При необходимости оба файла можно объединить в один файл свойств.

Требования архитектуры

Требования архитектуры source и target баз данных предполагают перед началом процесса репликации обязательную проверку их идентичности. Проверяется следующее:

- логическая структура баз данных (все области хранения, за исключением AI-областей, должны быть одинаковыми);
- версия базы данных;
- размер блока баз данных;
- статус опции Large File Support (должен быть одинаковым – включено/невключено);
- размер BI-блока.

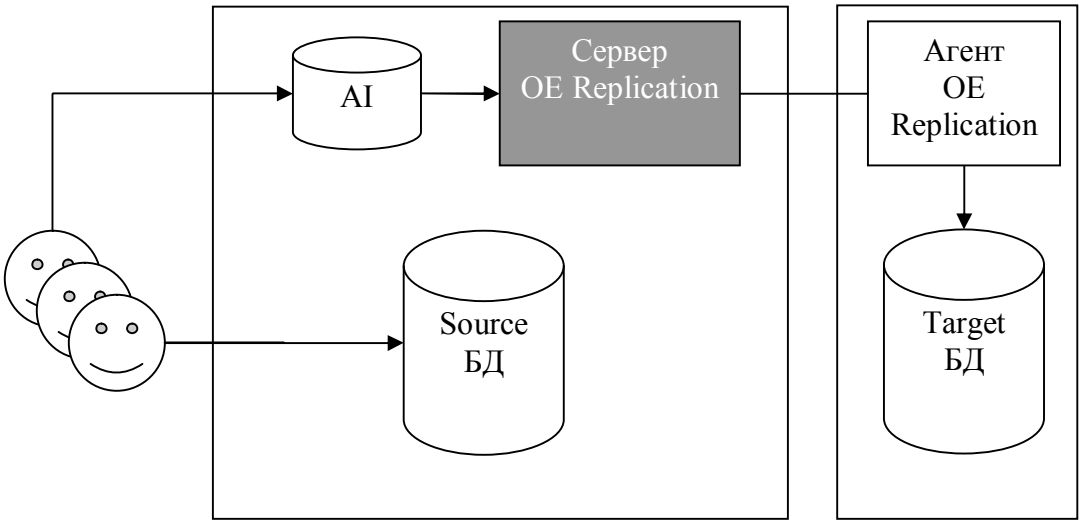
Как только OE Replication будет сконфигурирован и запущен, он начинает распространение любых изменений, сделанных на source-базе, на target-базу. Для обеспечения нормального функционирования OE Replication необходимо следовать основным принципам:

- Для безопасности target-база должна находиться на другой машине, отличной от той, на которой находится source-база.
- Как source, так и target база должны иметь одинаковый *endian ordering*, определяющий способ сохранения множества байт в памяти, который может быть MSB (most-significant byte) или LSB (least-significant byte). Термин *endianess* используется для описания двоичных данных, переносимых между платформами. Таким образом, платформы, имеющие одинаковые endianess, могут работать с двоичными данными, пересылаемыми между ними. Обычно Unix и Windows машины используют различные *endianess*. Следовательно, source-база, работающая на Windows, может копироваться только на другую Windows машину, и не может быть скопирована на машину с установленным HPUX. При этом HPUX машина может реплицироваться только с HPUX машиной.

Сервер OE Replication

OE Replication для свободной передачи данных от источника к цели должен иметь постоянную связь между source и target базами. Для этого используется сервер репликации, контролирующий изменения и пересылающий их специальному агенту репликации на target-базе. Так же сервер репликации отвечает за обеспечение подключения при старте, за блокировку схемы и восстановление сервера в случае сбоя.

Репликационная модель с обозначением сервера репликации:



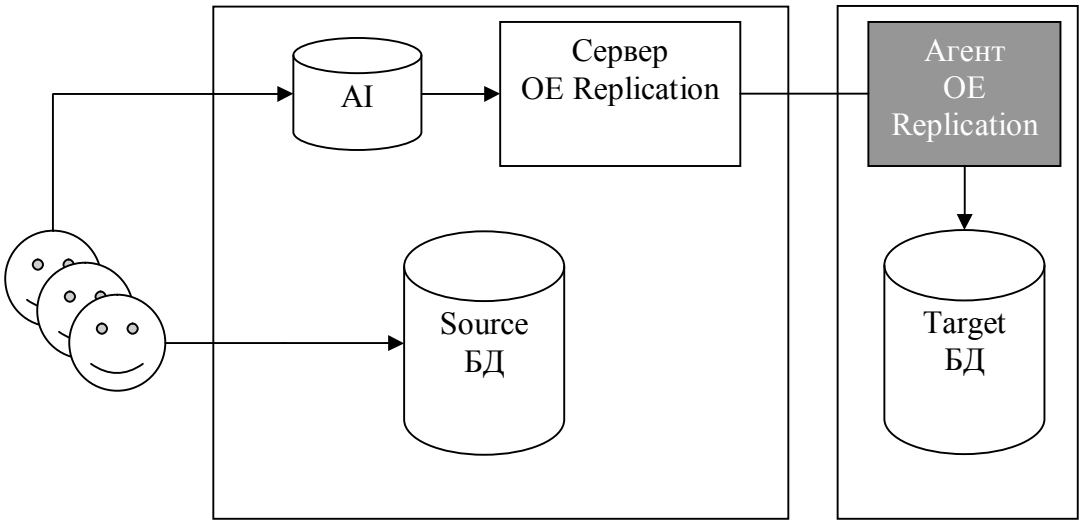
Сервер репликации располагается на той же машине, на которой расположена source-база. Основными задачами сервера является:

- подключение к source-базе, установление и поддержка постоянного соединения с ней;
- установка, поддержка и управление связью с агентом репликации на target-машине;
- отправка изменений с source-базы на target-базу. Этот механизм использует AI Transaction log для передачи информации из AI-блоков в target-базу.

Агент репликации

Для обеспечения идентичности source и target баз на target-машине требуется наличие агента репликации, в задачу которого входит получение информации и обновление target-базы. Агент получает от сервера репликации конфигурационную и операционную информацию, а также набор действий, необходимых для синхронизации в случае потери связи с сервером репликации.

Репликационная модель с обозначением агента репликации:



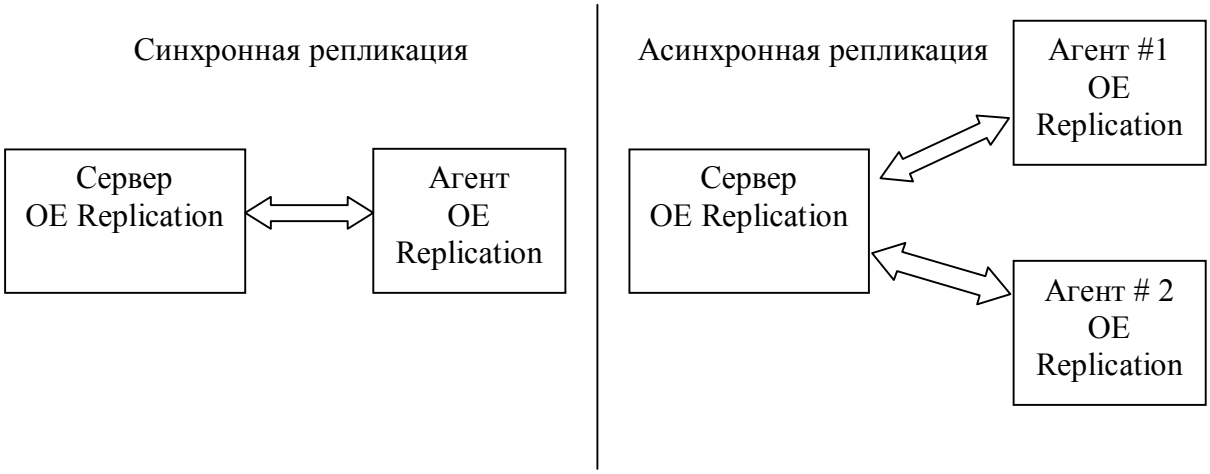
Агент обязательно должен располагаться на той же машине, на которой расположена target-база. Именно агент репликации фактически выполняет процесс обновления target-базы, обеспечивая поддержку следующих сервисов:

- обновление target-базы, используя AI-блоки, присылаемые сервером репликации;
- установка target-базы в режим Enhanced Read-Only (ERO).

Модель OE Replication с source и target базами данных, с сервером и агентом репликации, сосуществует параллельно со стандартной моделью базы данных со своими серверами, брокерами и прочими процессами.

Использование синхронной и асинхронной репликации

OE Replication поддерживает работу двух режимов копирования: синхронный и асинхронный. На рисунке продемонстрированы оба метода:

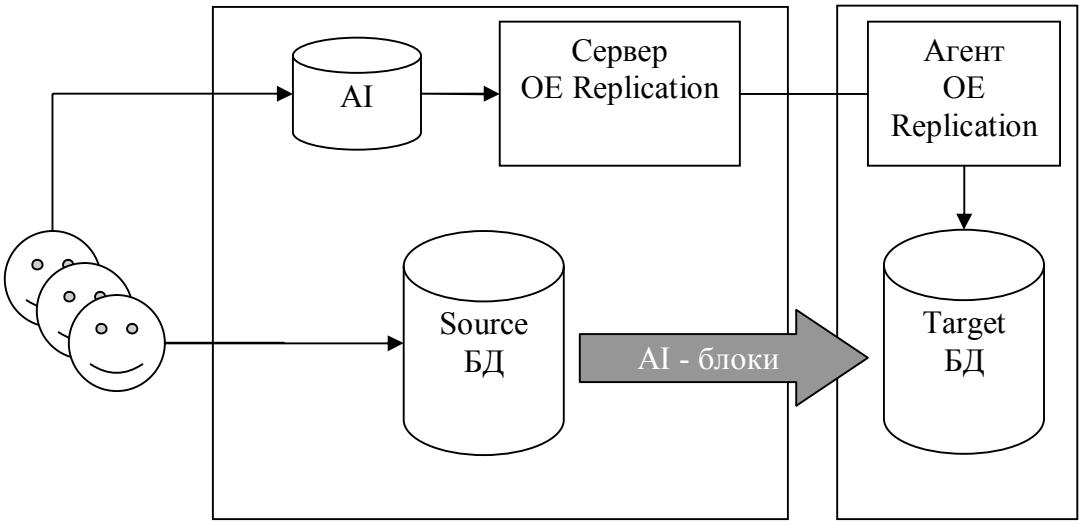


Для синхронной модели можно использовать только одного агента репликации, в то время как асинхронная модель позволяет работать с двумя агентами, по одному на каждую target-базу.

Во время **асинхронной** работы пользователь вносит изменения в базу, транзакция завершается без подтверждения и передается серверу репликации. Без ожидания сервер репликации из AI Transaction log отправляет AI-блоки агенту репликации, агент репликации вносит изменения в target-базу.

При сравнении двух режимов асинхронный режим является более производительным.

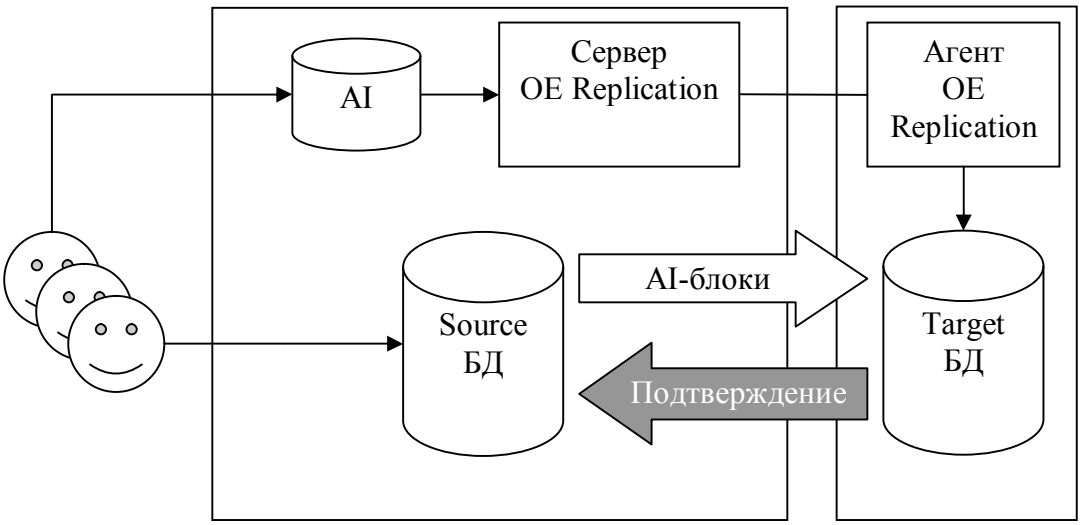
Схема работы OE Replication в асинхронном режиме:



Во время **синхронной** работы пользователь вносит изменения в запись и транзакция завершается. Когда агент репликации обнаруживает, что транзакция завершена, то отправляет подтверждение серверу репликации. При этом пользователь, совершивший транзакцию, на все это время блокируется. На других же пользователей этот процесс не влияет.

При сравнении двух методов синхронный режим является более безопасным, но при этом он более медленный, т.к. блокировка пользователей вызывает потерю производительности.

Схема работы OE Replication в синхронном режиме:



Обработка ошибочных ситуаций OE Replication

При возникновении сбойной ситуации, такой как например, потеря TCP/IP-соединения между сервером и агентом репликации, начинается процесс ее обработки. Если сбой произошел на стороне сервера репликации, то запускается процесс *failure recovery*. Если же на стороне агента репликации, то агент переходит в состояние *transition*, которое возникает при потере соединения с сервером. Для восстановления работоспособности OE Replication после сбоя и после восстановления соединения агент и сервер репликации должны выполнить процесс синхронизации.

Синхронизация

Синхронизация - это процесс, выполняемый сервером и агентом репликации, для приведения target и source баз к единому состоянию. Синхронизация выполняется каждый раз при старте базы данных, старте репликации, а также после каждого сбоя.

Server failure recovery

Процесс Failure recovery выполняется сервером репликации после разрыва (потери) соединения с агентом репликации. Во время работы процесса failure recovery сервер репликации делает попытки переподключения к агенту или агентам репликации, с которыми было потеряно соединение. После восстановления подключения агент и сервер репликации выполняют процесс синхронизации. Как только базы данных синхронизированы, работа OE Replication переходит в нормальный режим.

Если попытки переподключения не привели к успеху, сервер репликации удаляет не отвечающего агента из своего списка агентов. После удаления всех агентов сервер репликации будет автоматически остановлен. При этом деятельность source-базы будет продолжена, и AI-экстенты будут по-прежнему заполняться.

Transition

Процесс Transition выполняется агентом репликации после потери связи с сервером репликации. При этом в течение времени, установленного параметром *transition-timeout* (параметр настраивается на стороне сервера репликации в файле параметров *source-db-name.repl.properties*), агент «слушает» запросы, отправляемые сервером репликации. Как только соединение будет восстановлено, начнется процесс синхронизации между сервером и агентом. В случае удачного завершения нормальная работа OE Replication продолжится. Если соединение в течение времени, указанного параметром, установлено так и не будет, и если настроен *auto-transition*, то *target*-база данных будет переведена в нормальное состояние обычной OE базы данных. Если же используется manual transition, то агент репликации продолжит ожидание соединения до момента, пока ручной перевод базы в нормальное состояние не будет выполнен с помощью утилиты DSRUTIL, или база данных не будет остановлена.

Особенности работы с базой данных

В данной части будут кратко описаны некоторые моменты работы OE Replication.

Целостность

OE Replication для поддержания целостности базы данных обеспечивает обработку сбойных ситуаций (failure process). Поддержание целостности базы данных - это основная административная задача, т.к. как от целостности базы зависит работа всех пользователей системы. Поскольку цель работы OE Replication - это поддержка в идентичном состоянии минимум двух баз, целостность этих баз играет немаловажную роль и для репликации. Для достижения целостности OE Replication использует механизм синхронизации. Однако наличие двух синхронизированных баз не означает, что репликация может полностью заменить обычное резервное копирование баз данных. OE Replication является лишь одной из составляющих системы резервного копирования и восстановления.

Как только source-база запущена, менеджер базы начинает выполнять различные действия для обеспечения устойчивости базы данных к различным воздействующим факторам. Основным таким фактором является изменение содержимого базы данных пользователями, следовательно, OE Replication должен обеспечить копирование измененных данных на target-базы. Все изменения, которые могли произойти во время старта source-базы, реплицируются на target-базы путем синхронизации агентов и сервера репликации.

Доступность

Ещё одна функция OE Replication - это обеспечение доступности базы данных в случае возникновения сбойной ситуации. Поскольку OE Replication выполняет копирование данных на одну или более target-баз, любая целевая база может быть использована в случае недоступности основной, т.к. является её точной копией. Процесс перехода target-базы в source называется *transition*.

Режим ERO (Enhanced Read-Only)

После запуска агента репликации target-база переходит в режим *Enhanced Read-Only (ERO)*. В этом режиме все пользователи базы данных имеют доступ только на чтение данных. Это означает, что любой пользователь или процесс не может изменить ее данные или получить блокировку записей, исключением является агент репликации.

ERO режим в отличие от RO (Read-Only) режима является режимом сервера базы данных. Режим предоставляет пользователям все возможности базы данных, такие как буферный пул, разделяемые (shared) и частные (private) буфера.

Режим Read-Only (RO) устанавливает ограничения на стороне клиента, в то время как Enhanced Read-Only (ERO) устанавливает их на стороне сервера.

В случае возникновения сбойной ситуации на source-базе и после выполнения процесса transition на target-базе все пользователи получают полный доступ. Тем не менее имеющиеся соединения будут отключены, но как только они переподключатся, то также получат полный доступ.

Блокировка схемы (schema lock)

Когда схема source-базы изменяется, происходит ее блокировка. Сервер репликации передает информацию о блокировке схемы агенту репликации, тем самым вызывая блокировку на target-базе. При этом информация о блокировке записывается агентом репликации в лог target-базы.

По умолчанию блокировка схемы будет удерживаться до тех пор, пока изменения не будут завершены как на source, так и на target-базе. При этом если пользователь на target-базе работает с

таблицами, блокировка схемы не может быть выполнена, пока он их не освободит. Это связано с тем, что во время чтения таблицы устанавливается блокировка на ее изменение, пока чтение не будет завершено. Поэтому администратор базы данных должен убедиться, что никто из пользователей target-базы не работает с ней, поскольку это может вызвать проблемы в работе пользователя, изменяющего схему на source-базе.

Альтернативой этому может быть установка параметра Schema-Lock-Action. Когда параметр установлен, агент будет пять раз пытаться получить блокировку схемы target-базы. Если после пятой попытки блокировать схему так и не получилось, то агент принудительно отключит всех пользователей и повторит попытку блокировки. Если же и эта попытка окажется неудачной, работа сервера и агента репликации будет завершена. Это позволит source-базе продолжить нормальную работу. По завершении корректировки схемы source-базы, агент и сервер репликации должны быть перезапущены.

Принудительное усечение before-image файла source-базы данных

Для принудительного усечения BI-файла используйте следующую команду:

```
proutil db-name -C truncate bi -F
```

При этом следует помнить, что после выполнения команды механизм after-imaging перестает работать, а после перезапуска базы выполняется следующее:

- в логе базы будет зарегистрировано сообщение об отключении after-imaging;
- OE Replication будет отключен;
- база данных остановлена.

Отключение OE Replication требуется для обеспечения полного доступа административными утилитами к базе данных для выполнения процесса восстановления. После восстановления базы данных OE Replication должен быть заново активирован.

Точки останова (quit points) target-базы данных

Если на target-базе будет активирована точка останова (quit point), то source-база будет заблокирована. Блокировка будет удерживаться до тех пор, пока не будет снята точка останова на target-базе.

Выполнение online backup

В предыдущих версиях OE Replication выполнение online backup на target-базе было возможно только в процессе операции transition. Начиная с версии 10.1C online backup можно выполнять вне transition, т.е. в любое время, когда target-база запущена. Агент OE Replication - это по-прежнему единственный процесс, который может изменять target-базу, но в случае с online backup никаких изменений базы не происходит. Процесс backup только блокирует буфер и блоки, а не записи базы данных.

Во время выполнения online backup на target-базе активность source-базы сохраняется, пока выполняются следующие условия:

1. репликация выполняется в асинхронном режиме
2. сервер репликации способен получить блокировку разделяемой схемы source-базы, т.е. он должен иметь возможность получить эту блокировку на время работы процесса синхронизации
3. достаточно свободного пространства в AI-экстентах source-базы

Выполнение online backup на target-базе ничем не отличается от обычного online backup на обычной базе, синтаксис команды тот же:

```
probkup online db-name [incremental] device-name [parameters]
```

Что происходит во время выполнения этой команды?

Для того чтобы online backup был успешно выполнен на target-базе, необходимо согласовать работу этого процесса с работой OE Replication. Далее будет описано, что происходит с системой при запуске команды PROBKUP:

1. Если агент репликации активен:
   a. Утилита probkup посылает агенту репликации сообщение о том, что скоро начнется процедура online backup.
   b. Агент репликации информирует сервер репликации о начале online backup на target-базе, при этом проверяется, что:
      i. target-база на текущий момент не заблокирована (не активирована точка останова «quiet point», не включен BI stall или AI stall, или не выполняется другой online backup)
      ii. агент репликации имеет связь с сервером репликации
      iii. на текущий момент не выполняется процесс синхронизации
2. Затем происходит одно из следующих действий:
   a. Если сервер репликации способен получить блокировку разделяемой схемы на source-базе, и выполняется асинхронная репликация, то сервер репликации помечает базу данных как busy. Это позволяет базе продолжать нормальную работу. Затем сервер репликации отправляет положительный ответ агенту репликации.
   b. Если же сервер репликации не может получить блокировку разделяемой схемы, он уведомляет об этом агента репликации, который в свою очередь сообщает пользователю, запустившему probkup, о невозможности выполнения процедуры на текущий момент.
3. Если агент репликации получает положительный ответ от сервера репликации, то online backup-у позволяется продолжать работу.
4. Утилита probkup выполняет работу.
5. Когда online backup завершен, агенту репликации отправляется соответствующее сообщение о завершении резервирования.
6. В свою очередь агент репликации отправляет сообщение о завершении работы probkup серверу репликации.
7. Сервер репликации запускает процесс синхронизацию с целью передачи всех изменений, сделанных во время выполнения online backup, с source-базы на target-базу. Как только синхронизация будет завершена, сервер репликации вернется к нормальному процессу работы.
8. Процесс online backup завершен.

Планирование

Перед тем как начать использовать OE Replication необходимо выполнить ряд предварительных задач:

- Создать план резервного копирования.
- Оценить свою промышленную базу данных.

Примечание: OE Replication требует работы after-imaging, если этот механизм у базы еще не включен, необходимо его включить.

Обязательно убедитесь, что у Вас имеется полностью функциональный план резервного копирования и восстановления. В случае необходимости сначала создайте его и отладьте.

Итак, имея работоспособный план резервного копирования и восстановления, необходимо:

- Определить, какая база будет реплицироваться.
- Убедиться, что системных ресурсов достаточно для включения хотя бы after-imaging, поскольку после его включения дисковая активность будет увеличена, потребуется свободное пространство.
- Компьютер исходной базы данных, на котором будет включена репликация, должен иметь достаточное количество процессоров и памяти, чтобы справиться с дополнительной нагрузкой, которая будет вызвана OE Replication.
- Целевая база данных должна находиться только на другом компьютере, мощность которого не должна уступать мощности компьютера исходной базы данных. Это в первую очередь, связанно с формированием очередей из пересылаемых блоков, т.е. если целевой компьютер не будет вовремя справляться с обработкой данных, то вероятно ухудшение производительности на исходном компьютере.
- Обеспечить идентичность логических структур обеих баз данных.
- Обеспечить надежное TCP/IP-соединение между target и source базами.

OE Replication и After-imaging

Для работы OE Replication требуется активация After-imaging (AI). Механизм AI позволяет выполнять восстановление базы данных в случае ее повреждения на конкретный момент времени. Это возможно потому, что после включения AI, менеджер базы данных начинает записывать заметки обо всех изменениях базы в журнал after-image.

Перед тем как включить OE Replication, необходимо вычислить средний объем генерируемых базой AI-данных. Эту информацию можно использовать для оценки необходимых каналов связи для работы репликации. AI-данные играют огромную роль при определении необходимой ширины канала связи. Конечно, не всегда можно точно вычислить передаваемый объем, поскольку это зависит от таких факторов как размеры транзакций, активность пользователей, размеры и количество AI-экстентов и т.п.

Далее будет описано, каким образом AI-экстенты влияют на работу OE Replication и как рассчитать текущий объем AI-данных.

Экстенты After-Image

Экстенты After-Image содержат серии AI-заметок, сгруппированных в AI-блоки. Эти экстенты используются процессом roll-forward recovery для приведения базы данных к состоянию, в котором она находилась до момента повреждения, без потери транзакций, созданных после формирования последней резервной копии.

Существует два типа AI-экстентов: фиксированного размера (fixed-length) и переменного размера (variable-length). В основном экстенты фиксированной длины более предпочтительны, но бывают ситуации, когда предпочтительнее использование экстентов переменной длины. При определении, какого типа экстенты использовать, необходимо принимать во внимание бизнес-требования, а также преимущества и недостатки каждого типа. Основные характеристики типов приведены в таблице:

Тип экстента	**Преимущества**	**Недостатки**
Fixed-length	Экстенты фиксированной длины улучшают производительность за счет отсутствия необходимости распределения и форматирования блоков во время заполнения экстента.	Необходимость постоянного контроля заполнения экстентов, т.к. они могут быть заполнены во время цикла управления экстентами, поэтому следует принимать во внимание необходимость контроля.
Variable-length	Когда все блоки экстента заполнены и более не остается empty-блоков, менеджер базы данных вынужден размещать и форматировать новые блоки, что отразится на производительности.	Заполнение дисков – экстенты переменной длины не ограничены по размерам, а, следовательно, могут заполнить все свободное дисковое пространство. Возникает необходимость в управлении большими файлами в связи с тем, что размеры экстентов могут расти безгранично и могут возникнуть трудности с их обработкой.

Существует пять различных состояний экстентов After-Image в базе данных, в которой включен OE Replication:

Состояние экстента	**Описание**
EMPTY	Экстент пуст и готов к использованию.
BUSY	Экстент на текущий момент используется для записи. Транзакции будут собираться в экстенте, пока не произойдет автоматическое переключение на новый экстент менеджером базы или не будет выполнено ручное переключение администратором с помощью команды `rfutil` *`db-name`* *`-C`* `aimage new`.

Состояние экстента	Описание
LOCKED	В это состояние экстент переходит, когда он заполнен (FULL), но еще не скопирован на target-базу. К такому экстенту нельзя применить команду `rfutil db-name -C aimage empty`, пока агент репликации полностью не скопирует его содержимое в target-базу. Однако, содержимое экстента может быть выгружено в другой файл с помощью команды `rfutil db-name -C aimage extract`, не дожидаясь окончания копирования.
FULL	Экстент заполнен и готов к процедуре архивации.
ARCHIVED	В это состояние экстент переходит после того, как он будет заархивирован с помощью After-Image Management Utility.

Расчет текущего объема After-image

Существует несколько различных способов определения количества AI-данных, генерируемых базой. Для получения максимального значения из средних значений таких данных, необходимо снять статистику за короткий интервал времени. Например, раз в минуту в течение нескольких дней или месяцев. После определения среднего значения следует взять самое большое значение из всей статистики.

Четыре основных способа расчета объема AI-данных с помощью:

1. информации об after-image из VST (Virtual System Tables)
2. команды RFUTIL
3. PROMON
4. информации о before-image из VST

Важно отметить, что ни один из этих методов не принимает во внимание пиковых нагрузок.

Использование информации об after-image из VST

Для использования виртуальных таблиц необходимо обратиться к таблице _ActAILog. Поля, содержащие соответствующую информацию, называются _AiLog-TotWrites и _AiLog-BytesWritn. Поле _AiLog-TotWrites содержит информацию в блоках. Используя размер AI-блока можно точно узнать количество использованных данных. Разница между значениями в начале и в конце периода будет говорить о количестве записанных AI-блоков.

Использование этого метода более предпочтительно по отношению к другим, т.к. он является более гибким и наиболее простым, а так же не влияющим на производительность системы.

Использование команды RFUTIL

Команда RFUTIL используется для определения объема AI-данных из командной строки, для этого команду нужно выполнить в начале и в конце желаемого периода следующим образом:

```
rfutil db-name -C aimage extent list
```

Размер экстентов будет отображен в блоках по 1Кб. Значение количества блоков, записанных за период, получается путем вычитания количества блоков в начале периода из количества блоков в конце периода. Это значение необходимо для вычисления скорости формирования AI-данных относительно скорости работы сети. Важно отметить, что этот метод может привести к некорректным результатам, поскольку между временем начала и временем окончания периода AI-экстенты могут обнуляться (EMPTY).

Использование PROMON

Запустите PROMON для базы данных, которая должна стать source:

```
promon db-name
```

Как только утилита запустится, необходимо установить интервал сбора статистики. Для этого перейдите к пункту меню «Monitor sampling interval» по следующему пути R&D -> 5 -> 3. Введите значение 3600 (в секундах) и нажмите клавишу «Enter». Вернитесь на верхний уровень меню, нажав клавишу «P» и «Enter». Теперь перейдите к экрану «Activity: AI Log» для осуществления мониторинга (R&D -> 2 -> 6). Откроется окно, как показано:

```
                    Total  Per Min      Per Sec      Per Tx

Total AI writes     131    9      0.15    0.03
AIW AI writes       103    7      0.11   0.03
Records written     11298  752    12.54  2.92
Bytes written       842453 56101  935.02 218.08
Busy buffer waits   0      0      0.00   0.00
Buffer not avail    0      0      0.00   0.00
Partial writes      25     2      0.03   0.01
Log force waits     0      0      0.00   0.00
```

Следует обратить внимание на выделенное значение строки Total AI writes.

Для получения образца за интервал в 60 минут (3600 секунд) введите <S> и нажмите клавишу «Enter», выведется сообщение «Sampling for 3600 seconds ....»

По истечении 60-ти минут значения изменятся. Следует записать значение в пересечении строки Total AI writes и колонки Total. Это есть общее количество AI-блоков, записанных в течение часа. Процесс желательно повторить, чтобы получить среднее значение. Чем больше образцов, тем точнее среднее значение.

Теперь необходимо разделить полученное среднее значение сгенерированных AI-блоков на размер After-image блока. Размер AI - блока можно найти в меню R&D->1(Status Displays) -> 10 (AI Log), значение размера блока указано в предпоследней строке. Это значение необходимо разделить на 1024, чтобы получить размер блока в килобайтах (8192/1024=8). После чего полученное значение умножьте на среднее значение Total AI writes. В результате получится количество AI-блоков по 1 Кб, записанных во время часового периода.

Использование информации о before-image из VST

Для определения количества записанных AI-блоков можно использовать информацию из Before-Image VST. Для этого необходимо использовать виртуальную таблицу _ActBILog, а именно, поля _BiLog-TotalWrts и _BiLog-BytesWrtn. Поле _BiLog-TotalWrts формируется из блоков before-image. На основании размера BI-блока можно определить количество данных. Следует помнить, что размер BI-блока должен быть таким же, как и размер AI-блока. Разница между значениями начала и конца определенного периода и будет количеством блоков, записанных за период.

Этот метод является единственным для расчета объема AI-данных, если механизм After-imaging еще не активирован.

Репликация и After-Image экстенты

Все изменения в базе данных записываются как AI-заметки в AI-блоки. Сервер репликации считывает AI-блоки и, используя TCP/IP, отправляет данные агенту репликации. Агент в свою очередь, применяет AI-заметки, содержащиеся в AI-блоках, к target-базе.

Когда для базы данных активирован OE Replication, но при этом механизм в силу определенных причин не выполняет копирования, экстенты After-Image должны обеспечить хранение всех изменений в source-базе, пока процесс копирования не будет восстановлен. Как только OE Replication будет перезапущен, все накопленные нескопированные данные в AI-экстентах будут скопированы. Для этого все AI-данные должны оставаться доступными, чтобы OE Replication смог корректно их обработать. Сразу после обработки AI-экстенты будут помечены для повторного использования менеджером базы данных. Полное разблокирование AI-экстента для повторного использования может быть выполнено с помощью следующей команды:

```
rfutil db-name -C aimage empty
```

Все операции изменения source-базы генерируют AI-заметки, записываемые в AI-блоки, которые передаются агенту репликации. Состояние AI-экстентов исходной базы данных зависит от того, как происходит копирование между исходной и целевой базой данных. Пока копирование происходит в нормальном режиме, AI-экстенты будут архивироваться и освобождаться. Как только репликация будет нарушена, AI-экстенты исходной базы данных по-прежнему будут заполняться, но их статус будет оставаться LOCKED до тех пор, пока копирование не возобновится. Поэтому в таких ситуациях, необходимо помнить, что AI-экстенты могут вырасти до размеров намного больших, чем в случае, если бы репликации выполнялась нормально.

Помимо увеличения вместимости AI-экстентов существует еще один фактор, влияющий на размер экстента. Когда база данных активирована как source, менеджер базы генерирует дополнительные AI-заметки, которые пишутся одновременно в BI и AI-экстенты. Эти записи должны учитываться при расчете размеров AI-экстентов и схемы использования OE Replication.

OpenEdge на низком уровне осуществляет защиту целостности записей, используя так называемые логические операции, которые предохраняют запись от доступа, пока все действия по её изменению не будут завершены.

Типичная логическая операция может состоять из следующих действий:

- создание нового блока при расширении записи.
- очистка существующего блока, если запись удалена.
- данные о записи должны быть записаны в каждый блок, который занимает эта запись.
- обновление индексов записи.

Пока выполняется логическая операция, доступ к записи для добавления, изменения или удаления блокирован до ее завершения.

Размеры After-image экстентов на Source-базе данных

После того как определен объем генерируемых базой AI-данных, можно поменять размеры AI-экстентов. Рекомендуется установить размер одного AI-экстента достаточно большим, чтобы вместить четыре часа типичной деятельности базы данных при работе OpenEdge Replication. Рекомендуется так же установить общий размер всех AI-экстентов таким образом, что бы они могли вместить минимум одну неделю обычной деятельности базы данных. Такие рекомендации основываются на следующем:

- размеры экстентов должны учитывать время возможного простоя целевой базы данных;
- наиболее простые аппаратные сбои могут быть устранены в течение четырехчасового периода, например, дисковый или сетевой контролер может быть заменен в течение короткого времени;
- в определенных ситуациях целевая машина может быть выведена из строя на достаточно длительное время; в таких ситуациях общий размер AI-экстентов должен быть способен вместить все AI-заметки, которые будут сгенерированы за этот период.

Когда база данных активирована как source, в AI-экстенты записываются дополнительные AI-заметки. Этим заметкам требуется дополнительное свободное пространство, называемое *replication overhead*. Чтобы рассчитать replication overhead, необходимо полученный путем расчета объем AI-данных умножить на коэффициент 1,5.

Далее приведены примеры расчета размера AI-экстента для установления фиксированного размера. Результаты расчета приведены в конце каждой таблицы и выделены жирным шрифтом.

Расчет размера AI-экстента фиксированного размера

Расчет на основе ежечасных данных

Пример расчета для 4-х часового периода.

Описание	Расчет
Рассчитайте среднее количество сгенерированных AI-данных за 4 часовой период	1 час. - 70Мб 2 час. - 67Мб 3 час. - 79Мб 4 час. - 74Мб Среднее: 72,5Мб
Коэффициент *replication overhead*	1,5
Количество рабочих часов в день	16 часов
Количество рабочих часов в неделю (пятидневка)	16*5=80
Рассчитываем количество сгенерированных AI-данных в день	72,5Мб * 16ч. = 1160Мб или 1,13Гб
Рассчитываем количество сгенерированных AI-данных в неделю	1160Мб * 5д. = 5800Мб или 5,66Гб
Среднее количество сгенерированных AI-данных в час с учетом replication overhead	72,5Мб * 1,5 = 108,75Мб
Количество сгенерированных AI-данных с учетом replication overhead в день	1160Мб * 1,5 = 1740Мб
Количество сгенерированных AI-данных с учетом replication overhead в неделю	1740Мб * 5д. = 8700Мб
Размер одного фиксированного AI-экстента	108,75Мб * 4ч. = **435Мб**
Общее количество AI-экстентов, необходимых для работы в течение недели	8700Мб/435Мб = **20 экстентов**

Расчет на основе еженедельных данных

Используется размер сгенерированных AI-данных в течение одной полной недели.

Описание	Расчет
Общее количество сгенерированных AI-данных для одной полной рабочей недели	15000Мб или 14,6Гб
Коэффициент *replication overhead*	1,5
Количество часов в рабочем дне	16 часов
Количество часов в рабочей неделе	16ч.*7д. = 112 часов
Количество AI-данных, генерируемых за один час	15000Мб/112ч. = 134Мб
Количество AI-данных, генерируемых в день	134Мб*16ч. = 2144Мб или 2,09Гб
Рассчитываем количество сгенерированных данных в час с учетом replication overhead	134Мб*1,5 = 201Мб
Рассчитываем количество сгенерированных данных в день с учетом replication overhead	201Мб*16ч. = 3216Мб или 3,14Гб
Рассчитываем количество сгенерированных данных в неделю с учетом replication overhead	3216Мб*7д. = 22512Мб
Размер одного фиксированного AI-экстента	201Мб*4ч. = **804Мб**
Общее количество AI-экстентов, необходимых для работы в течение недели	22512Мб/804Мб = **28 экстентов**

Использование экстентов переменного размера

Приведенные выше примеры расчета размеров AI-экстентов не могут быть применены к экстентам переменного размера. Следует учитывать, что количество записываемых данных в экстенты переменного размера не отличается от записи в экстенты фиксированного размера. При использовании экстентов переменного размера необходимо обратить внимание на следующие факторы:

- размеры таких экстентов ограничены только размерами файловой системы при условии включения поддержки больших файлов;
- экстенты переменного размера будут заполняться, пока не достигнут максимально возможного в системе размера файла.
- нет необходимости контролировать заполнение множества экстентов, их переключение и т.п.
- так как экстенты предварительно не форматируются и не перераспределяются, никаких правил по формированию их размера нет.

Оценка сетевой пропускной способности OE Replication

Процесс репликация генерирует дополнительный replication overhead, связанный с передачей AI-блоков по сети. Эта дополнительная нагрузка оценивается в 10% от объема AI-данных и состоит из заголовка и контрольной информации для репликации.

Некоторые элементы, влияющие на дополнительную нагрузку сети:

- Упаковка блоков. OE Replication добавляет overhead в каждый AI-блок.
- По умолчанию, размер AI-блока равен 8K. Увеличение размера блока до 16K уменьшает нагрузку на сеть приблизительно на 5%.
- Стандартные средства связи между агентом и серверов OE Replication.

Определение пропускной способности сети зачастую может оказаться трудной задачей. Следует исходить из того, что пропускная способность сети составляет приблизительно 1,5 Мбит/сек. Данное значение является теоретическим пределом, на самом деле, оно значительно меньше из-за коммуникационного оборудования (маршрутизаторы, концентраторы и пр.).

Расчет эффекта присутствия OE Replication в сети включает множество различных факторов.

Наиболее влиятельным фактором расчета пропускной способности являются AI-данные. При этом нельзя точно вычислить объем AI-данных, т.к. точность расчета зависит от типа транзакций в приложениях и общей активности пользователей в базе данных. Тем не менее, зная средний объем AI-данных в системе, можно сделать приблизительный расчет. Формулы расчета объема AI-данных, передаваемых по сети с учетом replication overhead:

AID x AIRM = AIDR

AIDR x AIRO = AIDRO

AIDRO x (PRD) = AIDROP

Где:

AID - общий размер всех AI-блоков;

AIRM – коэффициент (1,5) для расчета дополнительных AI-заметок, генерируемых механизмом OE Replication;

AIDR - количество AI-блоков с учетом репликации;

AIRO – коэффициент (1,1) для расчета replication overhead.

AIDRO - количество AI-блоков с репликационными AI-заметками и с replication overhead;

(PRD) - период времени:

- 1 - выборки по 60 минут, и вам необходимы часовые значения;
- 1/60 - выборки по 60 минут и необходимы ежеминутные значения;
- 24 - выборки по 1 часу и необходимы ежедневные значения;
- 60 - выборки по 1 минуте и необходимы часовые значения;
- 1/24 - выборки по 24 часа и необходимы часовые значения;
- 1/1440 (24 * 60) - выборки по 24 часа и необходимы ежеминутные значения.

Пример 1:

- AID = 89MB
- Выборки по 60 минут
- PRD = 1

89MB x 1.5 = 133.5MB

133.5MB x 1.1 = 146.85MB

146.85MB x 1 = 146.85MB

В этом случае пользователь отправит примерно 146,85MB данных на target-базу за час.

Пример 2:

- AID = 360MB
- Выборки по 24 часа
- PRD = 1/1440

360MB x 1.5 = 540MB (566 231 040 байт)

5540MB X 1.1 = 594MB (622 854 144 байт)

594MB/1440 = 422 KB (432 537 байт)

Здесь пользователь будет отправлять в среднем 422K данных на target-базу каждую минуту.

По умолчанию размеры TCP/IP пакетов равны 1564 байт, из них 64 байта (приблизительно 4%) это заголовочная информация TCP/IP. Вероятно необходимо добавить эти данные в расчет для более точного представления об использовании сети.

OE Replication и бизнес-требования

Перед началом настройки after-imaging с OE Replication необходимо четко понимать бизнес-требования. Здесь кратко описаны основные моменты, которые стоит учесть во время настройки.

Время ожидания

Время ожидания в понятии OE Replication определено, как разница между временем начала выполнения изменения на source-базе и временем выполнения изменения на target-базе. Это время зависит от некоторых факторов, которые обязательно должны быть приняты во внимание:

- Количество и частота изменений на source-базе. Если изменения в базе данных происходят часто, и при этом AI-блоки быстро заполняются, то время ожидания будет коротким. Если же активность низкая, то время ожидания будет большим.
- Пропускная способность и загрузка сети. Если сеть медленная или работает на пределе возможностей, то время ожидания между source и target-базами будет возрастать.
- Время простоя target-базы. Если target-база будет длительное время недоступна, то будет расти и время ожидания. В этом случае придется принять решение о необходимости отключения работы OE Replication.

Допустимое время простоя target-базы

Существует несколько критериев, по которым можно определить допустимое время простоя базы:

- Как используется target-база? Если target-база используется только для получения отчетности, то длительное время простоя (более одного дня) не станет проблемой. Тем не менее, если база данных предназначена для резервирования на случай возникновения различных сбоев, то время простоя значительно уменьшается (допустимо 5 - 10 минут).
- Порог устаревания базы данных. Определение базы данных, как устаревшей может быть основано на некоторой временной метке или на объеме данных, который необходимо накатить на target-базу для синхронизации с souce-базой. Пороговое значение для принятия решения о перезапуске репликации определяется по времени или по объему данных.

Факторы, влияющие на восстановление

Способ восстановления может напрямую зависеть от времени ожидания и от допустимого времени простоя базы данных. Дополнительно на выбор способа восстановления оказывает влияние характер повреждения базы данных. Например, потеря базы данных может быть вызвана ее обычным отключением, что исправляется просто перезапуском базы. Потеря базы может быть также связана с аппаратными повреждениями, такими как разрушение дисков. В этом случае доступ к базе невозможен, из методов восстановления выбирается восстановление базы из резервной копии. В последнем случае необходимо определить, можно ли восстановить базу данных за короткое время, продиктованное требованиями вашего бизнеса. Например, если определено, что база данных может быть недоступна не более 15-ти минут, то исходя из этого времени и нужно выбирать метод восстановления базы данных.

Помните, что в результате сбоя некоторые данные могут быть потеряны. Например, если сбой на source-базе произошел между записью в BI и AI файлы, а текущий busy-экстент был применен к target-базе, возникает ситуация, при которой данные будут записаны только в AI и никуда более. Следовательно, source и target базы больше не могут быть синхронизированы.

Перед тем как начать работать с OE Replication, нужно четко понять следующее:

- Выполнение truncate BI на базе после сбоя может повлечь за собой откат большого количества транзакций. Этот процесс будет генерировать большое количество AI-записей, а поскольку OE Replication в этот момент не работает, то все AI-экстенты потенциально могут быть заполнены. Следовательно, перед выполнением truncate BI необходимо убедиться, что есть достаточно AI-экстентов для размещения большого количества AI-записей.

- База данных с включенной репликацией не может быть восстановлена или перезаписана без отключения OE Replication.

- В базе данных с OE Replication не может быть изменена как структура, так и данные, пока OE Replication не работает. Во время работы OE Replication только некоторые действия разрешены на source и target базах. При этом, в случае возникновения неавторизованного действия, выводится соответствующее сообщение.

- OE Replication не поддерживает базы данных, у которых включен механизм two-phase commit.

- OE Replication требует запуска минимум одного ABL-брокера. При этом ABL-брокер должен быть запущен первым по отношению к SQL-брокеру. Так же SQL-брокеру не нужно определять параметр запуска <-DBService>.

- Если свободное пространство на диске закончилось и больше нет AI-экстентов со статусом EMPTY (даже если AI-экстенты переменного размера), необходимо экстренное вмешательство. В противном случае база данных будет аварийно остановлена. Во избежание такой остановки необходимо при страте базы использовать параметр запуска After-image Stall (-aistall).

- Необходимо архивировать и управлять AI-экстентами, как частью стандартного after-image процесса.

Подведение итога

Запуск совместной работы OE Replication и After-imaging требует:

- принятия во внимание Бизнес-требований при настройке OE Replication и After-imaging;
- понимания, как должен быть настроен OE Replication, что важно при реализации проекта;
- поддержания работы механизма After-imaging (являющегося критичным процессом), обеспечивающим оптимальную работу OE Replication.
- соответствия количества AI-экстентов и их размеров бизнес-требованиям и возможностям аппаратных средств.

Реализация

Перед началом использования OE Replication необходимо определиться с методом его реализации. Собственно, методов немного, всего два: реализация по умолчанию (default implementation) и реализация с запуском отложенного агента (deferred agent startup implementation).

Реализация по умолчанию подразумевает, что агент OE Replication должен быть запущен до истечения времени соединения. Этот метод не допускает начала какой-либо активности как в target, так и в source базах до тех пор, пока сервер и агент репликации не завершат фазу запуска и инициализации.

Реализация с отложенным запуском агента, наоборот, позволяет запустить работу source-базы и сервера репликации без подключения к агенту репликации. Метод более удобен, например, при необходимости запустить работу репликации без остановки source-базы с помощью online backup. Преимущество очевидно, source-база возвращается к нормальной работе намного быстрее.

Следует отметить, что если база данных достаточно большая и времени на формирование online backup уходит много, то использование метода отложенного запуска агента является наилучшим выбором.

Использование метода по умолчанию

1. Остановите source-базу.
2. На source-сервере выполните следующее:
   a. создайте полную резервную копию базы данных;
   b. обновите структурный файл (.st);
   c. проверьте, запущен ли after-imaging, если нет, то запустите его;
   d. активируйте OE Replication;
   e. выполните инкрементальную резервную копию базы данных;
3. На target-сервере:
   a. скопируйте все резервные копии source-базы на target-сервер и восстановите там базу данных;
   b. активируйте OE Replication для target-базы;
4. Настройте OE Replication:
   a. создайте файл параметров для сервера репликации;
   b. создайте файл параметров для агента репликации;
5. Запустите базы данных в следующем порядке:
   a. target-база;
   b. source-база.

   Поскольку для обеих бах данных активирован OE Replication, то вместе с их запуском запустятся сервер и агент репликации. После запуска и инициализации сервера и агента репликации базы данных станут доступны для использования.

Использование метода отложенного запуска агента

Использование этого метода предоставляет доступ к работе с базой данных раньше, чем при использовании предыдущего метода. Это обусловлено тем, что создать target-базу можно на основе online backup с source-базы, в то время как при использовании метода по умолчанию source-базу необходимо останавливать и выполнять offline backup.

Для использования метода отложенного запуска агента в секцию [server] файла свойств OE Replication необходимо добавить параметр defer-agent-startup. При правильной установке этого параметра (в ненулевое значение) source-база может быть доступна пользователям еще до того, как сервер и агент репликации будут запущены и соединены. Это обусловлено тем, что когда сервер репликации запускается, он пытается подключиться к своему агенту, и в случае, если подключение невозможно, то он переходит в режим отложенного запуска агента. После чего попытка подключения будет повторяться каждые пять минут, до тех пор, пока подключение не будет установлено, либо не истечет период времени, установленный параметром defer-agent-startup.

Если по каким-либо причинам нет возможности ждать пяти минут, то для подключения к агенту можно воспользоваться утилитой DSRUTIL с классификатором connectagent. Если необходимо прервать ожидания defer-agent-startup, также следует воспользоваться утилитой DSRUTIL, но в этом случае с классификатором cancelDefer.

Как только сервер будет подключен к агенту, начнется процесс инициализации и синхронизации баз данных. Во время всего процесса подключения активность source-базы находится в нормальном режиме. Но в момент переключения репликации в нормальный режим обработки AI-блоков произойдет короткая приостановка ее деятельности.

Для активации OE Replication методом отложенного запуска агента необходимо:

1. Остановить source-базу.
2. На source-машине выполнить следующее:
   a. создать полную резервную копию source-базы, этот шаг необязателен при выполнении online backup. Online backup в основном предпочтителен в случаях, когда база данных имеет большие размеры, следовательно, формирование резервной копии может растянуться по времени.
   b. обновить структурный файл;
   c. проверить, запущен ли after-imaging, если нет, запустить его;
   d. активировать OE Replication;
   e. выполнить инкрементальную резервную копию базы данных (необязательно, если выполняется online backup).
3. Настройте сервер репликации следующим образом:
   a. определите значение параметра defer-agent-startup;
   b. настройте другие параметры OE Replication.
4. Запустите source-базу.
5. Если вы не создавали offline backup, создайте online backup сейчас.
6. На target-машине теперь можно сделать следующее:

   a. используя резервную копию source-базы, восстановите target-базу;

   b. активируйте OE Replication на target-базе;

7. Настройте файл свойств агента репликации.

8. Запустите target-базу.

9. Дождитесь, пока агент и сервер подключатся, либо воспользуйтесь утилитой DSRUTIL для ускорения процесса.

10. После того, как соединение будет установлено, после завершения процесса синхронизации базы данных перейдут в нормальный режим работы репликации.

Изменение схемы не допускается до тех пор, пока OE Replication сервер не завершил процесс синхронизации. Если же обновление схемы всё-таки будет инициировано, source-база блокирует изменение до завершения процесса failure recovery synchronization.

Если используется способ синхронной репликации, то активность базы данных не может продолжаться без подключенного агента.

Настройка source-базы данных для OE Replication в offline

1. Остановите source-базу данных.

2. Создайте резервную копию.

3. Обновите структурный файл. Как только будет выполнен backup, необходимо создать структурный файл следующей командой:

   ```
   prostrct list source-db-name source-db-name.st
   ```

   (позже будет необходимо скопировать этот файл с source на target машину)

4. Включите after-imaging. Для работы OE Replication необходима активация after-imaging. Если этот механизм уже включен на source-базе, пропустите этот пункт.

   a. Добавьте After-image экстенты к структуре базы данных

      ```
      prostrct add source-db-name addai.st
      ```

      После добавления AI-экстентов необходимо обновить структурный файл базы данных

   b. Выполните полную резервную копию source-базы

      ```
      probkup source-db-name source-db-name.bak
      ```

   c. Включите After-imaging с помощью утилиты RFUTIL

      ```
      rfutil db-name -C aimage begin
      ```

5. Активируйте OE Replication для source-базы. Активация требует того, чтобы она была выполнена до старта базы данных. Для этого выполните следующую команду:

   ```
   proutil source-db-name -C enableSiteReplication source
   ```

6. Выполните инкрементальную резервную копию source-базы следующей командой:

   ```
   probkup source-db-name target-db-name incremental
   ```

Инкрементальная копия будет использована для восстановления target-базы.

7. Для дальнейшей настройки перейдите к части «Старт target-базы данных»

Настройка source-базы данных для OE Replication в online

Для запуска репликации без остановки source-базы необходимо использовать метод реализации отложенного запуска агента. Необходимо убедиться в том, что включен механизм after-imaging.

1. Обновите структурный файл. Как только будет выполнен backup, необходимо создать структурный файл следующей командой:

   ```
   prostrct list source-db-name source-db-name.st
   ```

   (необходимо скопировать этот файл с source на target машину)

2. Активируйте OE Replication для source-базы.

   ```
   proutil source-db-name -C enableSiteReplication source
   ```

3. Настройте работу OE Replication для работы с отложенным запуском агента. Online копирование требует, чтобы сервер был сориентирован на отложенного агента. Для этого необходимо в файле свойств репликации установить значение параметра defer-agent-startup (в минутах). Например, это может выглядеть так:

   ```
   [server]
   defer-agent-startup=240
   ```

4. Выполните online backup. Чтобы использовать online копию для восстановления target-базы, она должна быть сформирована с параметром <–REPLTargetCreation>:

   ```
   probkup online source-db-name source-db-name.bak
   –REPLTargetCreation
   ```

Для дальнейшей настройки перейдите к части «Старт target-базы данных»

Настройка target-базы данных

1. Скопируйте структурный файл source-базы на target-машину. Если физическая структура source-базы и target-базы данных различается, например, области и экстенты на target-базе располагаются на других дисках или в других каталогах, необходимо изменить содержимое скопированного структурного файла.

2. Восстановите target-базу из полной резервной копий source-базы:

   ```
   prorest target-db-name source-db-backup-name
   ```

3. Восстановите инкрементальную копию

   ```
   prorest target-db-name source-db-incrementalbackup-name
   ```

4. Включите OE Replication для target-базы

   ```
   proutil target-db-name -C enableSiteReplication target
   ```

Настройка файла свойств

Настройка репликации подразумевает конфигурирование файла свойств, как для сервера репликации, так и для агента. OE Replication поддерживает два типа файла свойств: source.repl.properties и target.repl.properties. Файл свойств необходимо изменить и сохранить в том же каталоге, в котором будут располагаться source и target базы.

Также можно объединить все настройки в один файл свойств, задав ему имя db-name.replication.properties.

Настройка файла свойств для source-базы данных

Перед запуском OE Replication на source-базе необходимо настроить сервер репликации. Это выполняется путем сохранения некоторых параметров в файле свойств source-базы.

1. Сделайте копию файла свойств файла source.repl.properties, который находится в папке properties установочного каталога OpenEdge Replication.
2. Поместите этот новый файл в тот же каталог, в котором расположена source-база, назвав файл *db-name*.repl.properties, где db-name - имя source-базы.
3. Отредактируйте файл свойств.

Изначально файл содержит настройки по умолчанию, которые можно редактировать. Для редактирования можно использовать любой удобный текстовый редактор.

Пример файла свойств сервера репликации с двумя агентами:

```
[server]
            control-agents=agent1, agent2
            database=source
            transition=manual
            transition-timeout=1200
[control-agent.agent1]
            name=agent1
            database=your target name
            host=yourhost
            port=your port or service name
            connect-timeout=120
            replication-method=async
            critical=0
[control-agent.agent2]
            name=agent2
            database=your target name
            host=yourhost
            port=your port or service name
            connect-timeout=120
            replication-method=async
            critical=0
```

В этом примере agent1, содержащийся в параметре control-agents секции [server], соответствует имени агента, определенного в названии секции [control-agent.agent1], а agent2 соответствует названию секции [control-agent.agent2]. Каждый агент должен иметь уникальное имя.

По окончанию редактирования файла его необходимо сохранить.

К агентам применяются следующие правила:

- Если один из агентов удален из файла свойств, он не может впоследствии быть добавлен без пересоздания target-базы из source-базы.
- Нельзя переименовывать агента после того, как механизм OE Replication запущен.

Настройка файла свойств для target-базы данных

1. Сделайте копию файла свойств target.repl.properties, который находится в папке properties установочного каталога OpenEdge Replication.
2. Поместите этот новый файл в тот же каталог, в котором расположена source-база, назвав файл *db-name*.repl.properties, где db-name - имя target-базы.
3. Отредактируйте файл свойств.

Изначально файл содержит настройки по умолчанию, которые можно редактировать. Для редактирования можно использовать любой удобный текстовый редактор.

Пример файла свойств агента репликации:

```
[agent]
          name=agent1
          database=your target name
          listener-minport=4387
          listener-maxport=4500
[transition]
          database-role=normal
          auto-begin-ai=0
          auto-add-ai-areas=0
          ai-structure-file=repl1_ai.st
```

Запуск OE Replication

Для запуска репликации необходимо запустить source и target базы с параметром <-DBService replserv> и <–DBService replagent> соответственно.

Порядок запуска значения не имеет. Тем не менее, если значение параметра defer-agent-startup не установлено в ненулевое значение в файле свойств сервера репликации, то рекомендуется сначала запускать target-базу, чтобы сервер репликации не простаивал.

Если же source-база запущена первой, то target-база должна быть запущена до истечения времени, установленного параметром connect-timeout в секции [control-agent.name] в файле свойств сервера.

Старт source-базы данных

Для старта source-базы необходимо использовать следующую команду:

```
proserve -db source-db-name -DBService replserv
```

где параметр <–DBService replserv> указывает брокеру базы на необходимость старта сервера репликации. При этом следует помнить, что если не был использован метод реализации отложенного агента (параметр defer-agent-startup), то доступ к базе будет закрыт до тех пор, пока репликация не будет запущенна и инициализирована.

Следует обратить внимание, что параметр <–DBService> может быть использован совместно с иными параметрами, которые ранее были использованы при старте базы данных. Кроме того, при использовании для target-базы тех же параметров, что и для source-базы, необходимо весьма осторожно выбирать значения параметров, работающих с разделяемой памятью (shared memory), например, с таким как number of users, clients per server, maximum number of servers и т.п.

В случае ошибки настройки сервер OE Replication не сможет запуститься, даже если он работает с некритичным агентом. Ошибками настройки могут быть:

- отсутствие или потеря AI-экстентов;
- неправильно созданная target-база;
- несоответсвие target-базы source-базе в плане поддержки больших файлов.

Старт target-базы данных

Для старта target-базы необходимо использовать следующую команду:

```
proserve -db target-db-name -DBService replagent -S port | service name
```

Брокер базы данных должен контролировать port или service name, используемые параметром <–S>, следовательно, этот параметр обязателен. Значения port или service name должны быть обязательно такими же, какими они указаны в файле свойств сервера репликации. Эти значения могут совместно использоваться ABL-клиентами и сервером репликации. Если брокер запущен и настроен как для ABL, так и для SQL-соединений, то SQL-клиент так же может использовать значения port или service name. Параметр <–DBService> указывает брокеру на старт агента репликации. Пока не завершится процесс синхронизации агента и сервера репликации, доступ к target-базе будет ограничен.

Журналирование запуска

Пока OE Replication запускается, все сообщения, сопутствующие этому процессу, записываются в специальный журнал, который может быть использован в случае сбоя во время запуска. Имя файла этого журнала: для сервера репликации – repl.server.startup, для агента – repl.agent.startup. Файл создается автоматически в текущем рабочем каталоге, из которого был запущен процесс старта базы. Он необходим, поскольку в момент старта недоступна запись в стандартный журнал базы данных (db-name.lg).

Процесс запуска и инициализации OE Replication

Во время запуска OE Replication и процесса инициализации сервер репликации пытается подключиться к агенту репликации target-базы, через брокера базы, используя порт, определенный в секции control-agent в файле свойств сервера репликации. Как только между ними устанавливается контакт, подтверждается связь. Во время этого процесса определяется идентичность source и target баз, выполняется синхронизация и включается возможность подключений к базе.

Если параметр defer-agent-startup установлен в ненулевое значение, то доступ к source-базе возможен сразу после ее старта. Деятельность source-базы не будет остановлена до самого конца синхронизации, т.е. пока OE Replication не завершит её. Когда синхронизация завершится, произойдет кратковременная блокировка базы, в течение которой сервер репликации вернется к процессу записи AI-блоков, после чего база данных будет разблокирована для нормальной работы с ней.

До начала синхронизации, если сервер репликации не может с первого раза подключиться к агенту репликации, он переходит в режим отложенного запуска агента. В этом состоянии сервер ожидает пять минут, после чего повторяет попытку подключения. Он будет оставаться в этом состоянии до тех пор, пока все агенты репликации не будут подключены, либо пока не истечет время, установленное параметром defer-agent-startup. Процесс инициализации и синхронизации начнется только тогда, когда все агенты подключены. Для того, чтобы не ждать пяти минут между попытками подключения к агентам, можно использовать утилиту DSRUTIL с классификатором connectagent.

Изменение схемы базы данных, пока сервер репликации выполняет процесс синхронизации, не допускается. Если же будет предпринята попытка изменения схемы в момент выполнения синхронизации, то процесс изменения заблокируется до тех пор, пока процесс синхронизации не будет завершен.

Активность source-базы не может продолжаться без агента(ов), если используется синхронная репликация.

Восстановление source и target баз данных

Когда необходимо восстановить базу из резервной копии на место существующей базы, у которой включен механизм OE Replication, то предварительно OE Replication должен быть отключен.

Отключается механизм репликации в восстанавливаемой базе следующей командой:

```
proutil source-db-name -C DisableSiteReplication source
```

Теперь можно приступить к процедуре восстановления с помощью стандартной утилиты PROREST:

```
prorest source-db-name  file | device
```

Во время копирования, либо сразу после отключения репликации необходимо удалить файл db-name.repl.recovery.

Если база восстанавливается из резервной копии, созданной с базы, активированной для репликации, то после восстановления из такой копии информация о включении механизма OE Replication останется, но он будет неработоспособным. Поэтому сразу после восстановления базы OE Replication должен быть в ней отключен, и при необходимости активирован заново. Target-базы после такого восстановления необходимо создать заново.

Доступ к базе данных с OE Replication

Пока выполняются процессы запуска OE Replication и синхронизации агента и сервера репликации, любые подключения к базе данных невозможны. Исключение составляет случай использования параметра defer-agent-startup. В следующей таблице приведены описания реакций на попытку подключения к базе различных типов клиентов.

Тип клиента	Результат подключения
Self Serve Client	Блокируется, пока подключение не будет разрешено.
Remote Client	Завершается с сообщением о невозможности создания сервера брокером. Сообщение отправляется в лог базы данных.
Open Edge Utility	Блокируется, пока подключение не будет разрешено.
Forced Shutdown	Всегда доступно.
Java ODBC Client (SQL Server access)	Завершается с ошибкой сетевого демона.
AppServer Self Serve Client	Блокируется, пока подключение не будет разрешено.
AppServer Remote Client	Завершается с сообщением о невозможности создания сервера брокером. Сообщение отправляется в лог базы данных.
WebSpeed Self Serve Client	Блокируется, пока подключение не будет разрешено.
WebSpeed Remote Client	Завершается с ошибкой, сообщающей о невозможности создания сервера брокером. Сообщение отправляется в лог базы данных.

Доступ к target-базе

Тип доступа к target-базе зависит от установленного типа продукта OE Replication и от деятельности агента репликации на текущий момент. Если установлен продукт OE Replication Plus, то любой процесс может получить доступ к target-базе в режиме read-only. При попытке изменения базы этими процессами будет сгенерировано сообщение об ошибке. Если же установлен просто OE Replication, то доступ к target-базе ограничен на системном уровне согласно таблице, представленной ниже, так же запрещен доступ клиентов в режиме read-only, он будет возможен только после перевода target-базы к нормальному состоянию.

Когда агент репликации запущен, разрешаются ограниченные подключения к target-базе. Нормальная возможность подключения будет восстановлена только после завершения процесса синхронизации.

Стандартные системные уровни доступа к target-базе.

Инструмент	**Описание**
Shut down	Остановка базы данных командой PROSHUT.
Monitor	Отображение информации о базе данных утилитой PROMON.
AIW	Фоновый процесс, записывающий AI-буфера на диск после их заполнения.
BIW	Фоновый процесс, который непрерывно записывает заполняемые BI-буфера на диск.
WDOG	Процесс watchdog, который выполняет чистку за некорректно завершенными процессами.
OpenEdge Replication agent	Получает информацию от OE Replication Server и изменяет target-базу.
OpenEdge Replication Utility DSRUTIL	Допускается выполнение определенных команд на target-базе.
Database agent	Обеспечивает OE Management информацией о состоянии базы данных.

Если агент репликации выполняет переход к нормальному состоянию базы данных (transition), то доступ к ней получают только PROSHUT, PROMON и WDOG.

При завершении перехода процесса к нормальному состоянию базы данных (transition) возможность нормальной работы с базой данных восстанавливается.

Доступ к source-базе

Независимо от того, какой продукт установлен (OE Replication или OE Replication Plus), доступ к source-базе открыт для любых подключений.

Останов OE Replication

Останов Source-базы

Для остановки работы базы данных используются утилиты PROSHUT или PROMON, останавливать следует сначала source-базу и только затем target-базу.

Для нормальной остановки source-базы используйте команду:

```
proshut source-db-name -by
```

при этом происходит следующее:

1. Все процессы, кроме брокера базы и сервера репликации, останавливаются.
2. Брокер останавливает всю деятельность в базе данных.
3. Сервер репликации после сброса AI-буферов начинает их обрабатывать.
4. Сервер репликации по завершению обработки сообщает агенту репликации о необходимости остановки работы.
5. После завершения процесса остановки сервера репликации осуществляется остановка базы данных.

Для экстренной остановки source-базы предназначена команда:

```
proshut source-db-name –byF
```

При этом база данных будет немедленно остановлена с записью соответствующего сообщения в лог базы. Следует помнить, что при такой остановке AI-буферы на диск сброшены не будут, а, следовательно, source и target базы не будут идентичными. Поэтому процесс активации OE Replication необходимо будет выполнить с самого начала.

Останов Target-базы

Для остановки target-базы может быть использована принудительная или обычная остановка.

Обычная остановка target-базы выполняется командой:

```
proshut target-db-name –by
```

При этом происходит следующее:

1. Останавливаются все процессы кроме брокера базы и агента репликации.
2. Брокер сбрасывает все изменения на диск.
3. После сохранения изменений агент репликации информирует сервер репликации о том, что начинается остановка. Если это единственный агент, обслуживаемый сервером репликации, то сервер репликации будет остановлен. Также сервер репликации будет остановлен, если агент репликации будет критическим, даже если существует дополнительный агент репликации.

4. Остановка базы данных по завершению работы агента репликации.

Для экстренной принудительной остановки нужно выполнить команду:

```
proshut target-db-name –byF
```

Необходимо помнить, что после использования принудительной экстренной остановки работа OE Replication будет нарушена, процесс активации OE Replication придется выполнить заново.

Останов сервера OE Replication

Для аварийной остановки работы OE Replication сервера без остановки source-базы данных предназначена команда:

```
dsrutil source-db-name -C TERMINATE server
```

При получении сервером репликации этой команды его агенты также получают команду остановки. На статус базы данных остановка сервера репликации не влияет.

После остановки сервера репликации при необходимости можно деактивировать OE Replication. Для этого необходимо сделать следующее:

1. освободить любые незавершенные операции, связанные с репликацией, командой:

```
dsrutil source-db-name -C RELWAITS
```

2. деактивировать OE Replication командой:

```
dsrutil source-db-name -C DisableSiteReplication source
```

После деактивации source-базы данных единственным способом возобновления репликации является повторная активация механизма OE Replication с созданием новых target-баз.

Останов агента OE Replication

Для остановки агента репликации предназначена команда:

```
dsrutil target-db-name -C TERMINATE agent
```

После выполнения команды агент проинформирует сервер репликации о том, что он (агент) будет остановлен. Если этот агент был единственным обслуживаемым сервером, то сервер репликации так же будет остановлен. Если агент репликации имеет статус критичного (параметр агента critical имеет значение 1), то, даже если будет существовать дополнительный агент, сервер репликации всё равно будет остановлен.

Для деактивации OE Replication после остановки агента выполните команду:

```
dsrutil target-db-name -C DisableSiteReplication target
```

Очистка AI-экстентов на source-базе данных

Все AI-экстенты на source-базе могут заполниться, при этом, если во время старта базы данных применялся параметр <-aistall>, то процесс будет приостановлен, если же параметр не использовался, то процесс остановится. Если такое произойдет, то AI-экстенты должны быть очищены. Для этого нужно воспользоваться командой:

```
rfutil source-db-name -C aimage extent empty
```

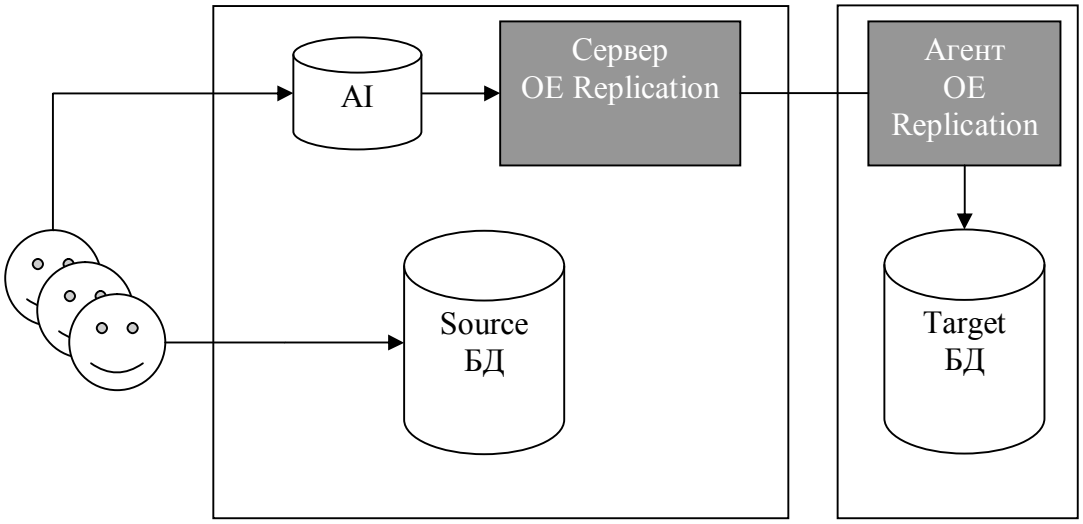
Следует отметить, что AI-экстенты не могут быть освобождены, пока они не скопируются на target-базу. При этом, если активность базы продолжается во время процесса обработки сбоя, или, если используется параметр defer-agent-startup, то AI-экстенты могут значительно вырасти в размерах. Необходимо обеспечить достаточно пространства для роста размера AI-экстентов. В противном случае, как только все AI-экстенты заполнятся, или не хватит свободного пространства на дисках, сработает aistall. Если устранить aistall нет возможности, то OE Replication должен быть интерактивно остановлен. Для этого выполните команду:

```
dsrutil source-db-name -C DisableSiteReplication source
```

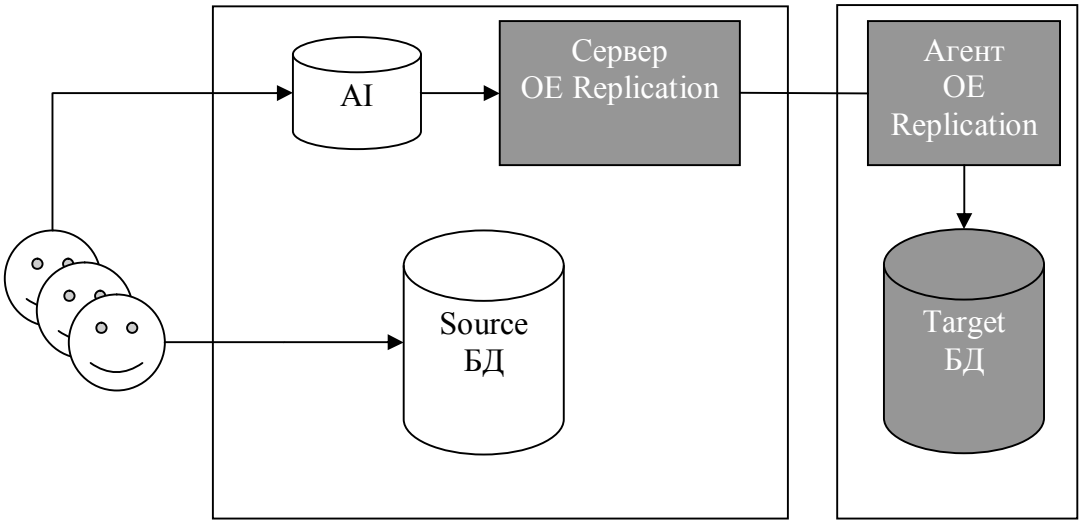
Нормальный режим работы OE Replication

Работа OE Replication начинается с момента запуска source и target баз данных. Пока сервер и агент репликации выполняют процесс запуска и инициализации, никакая иная деятельность в базе данных не разрешена, конечно, если не используется метод отложенного запуска агента. Если какой-либо процесс попытается подключиться к любой из баз данных, он будет блокирован до тех пор, пока OE Replication не завершит процессы запуска, инициализации и синхронизации.

Во время нормальной работы source-базы все AI-блоки, записанные в AI-экстенты, пересылаются серверу репликации, который, в свою очередь, отправляет эти блоки всем настроенным агентам репликации. Этот процесс выполняется, пока source-база не будет остановлена.



Во время нормальной работы агент репликации выполняет непрерывный накат AI-блоков (roll forward) на target-базу. Эти блоки он получает от сервера репликации. Процесс выполняется до тех пор, пока не будут остановлены source и target базы или одна из них.



Помимо наката AI-блоков на target-базу агент репликации отвечает за обработку таких событий как:

- формирование контрольных точек (checkpoints);
- переключение AI-экстентов;
- завершение транзакций.

Эти события используются агентом репликации для информирования сервера репликации о точке синхронизации, известной как sync point. Точка синхронизации указывает серверу репликации на то, что с этого места необходимо начать процесс синхронизации во время перезапуска баз данных, или во время обработки ошибочной ситуации.

Когда репликация выполняется в синхронном режиме, любой процесс, который завершает или откатывает транзакцию, будет блокирован или вынужден ожидать до тех пор, пока эта транзакция не будет обработана агентом репликации. Информация о завершении агентом обработки транзакции будет передана серверу репликации, который в свою очередь разблокирует ожидающий процесс.

Информация о точке синхронизации и прочие данные, необходимые для синхронизации, хранятся в файле db-name.repl.recovery, где db-name - имя source или target базы. Этот файл редактировать нельзя! Но его нужно вручную удалить перед восстановлением source или target базы или перед активированием OE Replication после предыдущего выключения этого механизма.

Заметьте, если произошел сбой в работе агента репликации, например, когда процесс rpagent аварийно прекратил работу, то единственным способом его перезапуска является перезапуск всей базы данных. При нормальных обстоятельствах откат активных транзакций, оставшихся после аварийного завершения процесса, которому они принадлежат, выполняют брокер базы и процесс watchdog. Но откат транзакций не может быть выполнен от имени агента репликации, поскольку все изменения в target-базе всегда исходят от source-базы. Таким образом, после сбоя агента репликации откат транзакции будет невозможен, и target-база данных останется в противоречивом состоянии. Это приведет к тому, что ABL и SQL процессы, которые будут подключаться к такой базе, завершатся аварийно. Для синхронизации target-базу с source-базой и восстановления нормальной работы необходимо выполнить перезапуск target-базы.

Горячая копия базы данных

Если произойдет сбой, то target-база может быть использована как горячая резервная копия. Такая база готова к работе в любое время. Можно иметь до двух target-баз, но только одна из них может быть настроена для автоматического перехода в состояние, когда с ней можно работать как с source-базой. Делается это через настройку параметров агента репликации critical (значение 1) и transition (значение auto). Только один агент может получить статус critical, поскольку пользователи не должны вносить изменения в две различные базы после выполнения автоматического перехода (transition). Вторая target-база может быть переведена в состояние source-базы только в ручном режиме. Какая из двух баз станет горячей копией, и в каком режиме (автоматическом или ручном) эта копия будет приведена к нормальной работе пользователей с ней, зависит от вашего решения.

При выборе автоматического перехода в нормальное состояние базы данных некоторые транзакции во время процесса обработки сбоя, возможно, будут потеряны.

Если работает синхронный режим и target-база переходит в нормальное состояние, то максимальные потери - по одной транзакции на клиента. Если используется асинхронный режим, то может быть потеряно большее количество транзакций, которое зависит от состояния сервера репликации на момент возникновения сбоя, а также от дальности расположения source-базы и сервера.

Для определения количества потерянных транзакций OE Replication представляет соответствующий отчет. Для этого используется утилита DSRUTIL MONITOR.

Преимущество использования горячей копии базы данных заключается в том, что в случае возникновения сбоя база данных будет доступна пользователям намного быстрее, чем если бы восстановление происходило из резервной копии.

Условия обработки сбоев

Когда OE Replication работает, активность базы данных находится в нормальном состоянии. Если же OE Replication сталкивается со сбойной ситуацией, выполняется обработка этой ситуации. При этом восстановление основывается на различных характеристиках сбоя, которые разделены на две категории:

- Сбой базы данных
- Потеря связи

Примечание: остановка source или target базы с помощью команды PROSHUT без применения параметра <–F> (force) не считается сбойной ситуацией и не приводит к запуску процесса восстановления. Остановка сервера репликации командой DSRUTIL db-name –C terminate Server или остановка агента с помощью команды DSRUTIL db-name –C terminate Agent не классифицируется как сбой.

Сбой базы данных означает что, в случае повреждения source или target базы OE Replication немедленно реагирует на это событие. Если повреждена target-база, сбой происходит в работе агента репликации. В этом случае сервер репликации выполнит процесс failure recovery, поскольку он потерял соединение с агентом. Если повреждена source-база, то нарушается работа сервера репликации. В этом случае агент репликации переходит к выполнению failure processing, например, при потере связи с сервером.

Потеря связи означает что, сервер репликации потерял контакт со своими агентами. Это может произойти по следующим причинам:

- Аварийная остановка сервера репликации
- Системный сбой машины, на которой запущен сервер репликации
- Разрыв TCP/IP-соединения между сервером и агентом репликации

После потери соединения source-база переходит в режим failure recovery, а target-база в режим transition.

Есть вероятность того, что потеря TCP/IP-соединения между сервером и агентом репликации пройдет незаметно. Например, в больших сетевых комплексах с большим количеством различного сетевого оборудования один из сегментов сети может выйти из строя, вызвав этим разрыв связи между хост-машиной агента и хост-машиной сервера. Но, тем не менее, TCP/IP по-прежнему остается работать в другом сегменте сети, и сервер или агент могут не подозревать о разрыве связи. Проверка наличия связи TCP/IP между агентом и сервером выполняется через пингование (от англ. ping). Для включения этой возможности необходимо использовать параметр Repl-Keep-Alive в файле свойств агента репликации. Пингование будет выполняться каждые тридцать секунд. Этот параметр позволяет задать время ожидания ответа в секундах. Если ответа за это время нет (по умолчанию 300 сек.), то считается, что возникла проблема со связью и начинается процесс обработки сбойной ситуации failure recovery.

Когда сервер репликации теряет соединение с одним из своих агентов (либо со всеми), он пытается восстановить связь с ним в течение времени, которое установлено параметром connect-timeout в файле свойств сервера репликации. Более детально этот процесс выглядит так:

1. Сервер репликации обнаруживает, что произошел сбой связи с агентом.
2. В течение определенного времени сервер пытается установить соединение с агентом репликации. При этом активность source-базы поддерживается в нормальном режиме, если не используется синхронный метод репликации, или пока схема базы может быть изменена процессами.
3. Если серверу удалось соединиться с агентом, он начинает передавать AI-блоки. Когда останется примерно 10 AI-блоков, сервер репликации приостановит активность базы данных и завершит процесс синхронизации. Во время выполнения синхронизации невозможно обновление схемы. Если в момент выполнения failure recovery схема будет изменяться, то source-база будет заблокирована до тех пор, пока процесс не будет завершен. Активность source-базы не может продолжаться без агента репликации, работающего в синхронном режиме.
4. Когда процесс синхронизации будет завершен, сервер репликации перейдет в нормальный режим обработки AI-блоков и работа базы и репликация продолжатся в нормальном режиме.

Если серверу репликации не удалось подключиться ко всем агентам репликации или к критичному агенту (если такой имеется) в период, определенный параметром connect-timeout, то сервер репликации будет остановлен и работа source-базы продолжится в нормальном режиме. Другими словами, если нет критичного агента, сервер должен иметь возможность подключиться ко всем некритичным агентам, либо его работа будет завершена. Если один агент является критичным, то сервер продолжит работу при возможности подключения только к нему. При этом следует помнить, должно быть достаточно свободного пространства в AI-экстентах, что бы вместить все изменения за время простоя сервера репликации, если активность source-базы продолжается.

Есть вероятность, что во время выполнения процесса синхронизации работа сервера репликации будет отставать от работы RDBMS, поэтому на это время source и target базы могут отставать по времени.

Когда агент репликации теряет соединение с сервером репликации, он переходит в состояние pre-transition. В таком состоянии, если настроен автоматический переход, агент для восстановления соединения «слушает» запросы от сервера репликации в течение времени, указанного параметром transition-timeout. При этом происходит следующее:

1. Когда связь теряется впервые, агент переходит в состояние pre-transition, в котором «слушает» запросы сервера репликации.
2. Если связь не установлена, и агент настроен на автоматический переход, target-база переходит в режим работы нормальной OpenEdge базы. Это означает, что все стандартные возможности работы с базой данных восстановлены.
3. Если настроен ручной переход, агент репликации продолжает ожидание соединения до тех пор, пока администратор вручную не выполнит transition. Пока администратор этого не сделает с помощью утилиты DSRUTIL база будет находиться в неопределенном состоянии.

Переход в target-базе данных

Transition – это процесс, после выполнения которого target-база становится базой. Для правильной работы этого процесса необходима его настройка, до которой необходимо определить, сколько target-баз будет использоваться, одна или две.

Одна или две?

OE Replication имеет возможность работать с двумя target-базами, но только одна из них может быть настроена для автоматического перехода. Target-база, которая автоматически перейдет в рабочее состояние после сбоя, это первая база у которой в секции [control-agent.agent] в файле server.repl.properties, значение параметра critical равно единице, а значение параметра transition равно auto. Механизм OE Replication использует только первую target-базу, в файле server.repl.properties для автоматического перехода. Перевод второй target-базы в рабочее состояние должен осуществляться в ручном режиме.

Target-база, настроенная как critical, может использоваться в качестве горячей копии. Она должна находиться на машине с надежным TCP/IP и с достаточным запасом ресурсов для того, чтобы обеспечить работу пользователей и самой базы, если source-база окажется недоступной.

Только одна база должна быть определена для перехода (transition). Перевод в нормальное состояние обеих баз повлечет возможность внесения пользователями изменений в обе базы, в результате чего обеспечить их целостность будет невозможно.

Настройка автоматического перехода

Как уже говорилось, когда агент теряет связь с сервером, он в течение времени, указанного свойством transition-timeout, ожидает восстановления соединения с сервером репликации. По истечению заданного времени, после которого связь с сервером не восстановлена, target-база будет переведена агентом в нормальное состояние рабочей базы.

Для настройки автоматического перехода должно быть выполнено следующее:

- параметру transition (секция server) должно быть присвоено значение auto;
- параметру critical (секция control-agent) должно быть присвоено значение 1;
- параметру transition-timeout (секция server) должно быть установлено значение, определяющее время ожидания отклика от сервера репликации.

Пример файла свойств агента с автоматическим переходом:

```
[server]
          control-agents=AutoAgent
          database=source
          transition=auto
          transition-timeout=1200
[control-agent.AutoAgent]
          name= AutoAgent
          database=target
          host=localhost
          port=4502
          connect-timeout=120
          replication-method=async
          critical=1
```

В этом примере время начала автоматического перевода target-базы в нормальное рабочее состояние агентом AutoAgent равно1200 секунд.

Настройка ручного перехода

При ручном переходе время ожидания агентом перехода не определено, так как переход может быть выполнен только администратором. При этом, если сервер репликации сможет подключиться до выполнения перехода, то работа OE Replication будет восстановлена.

Для ручного перехода нужно настроить следующие параметры агента репликации:

- transition присвоить значение manual;
- critical присвоить значение 0, т.е. некритичный агент;
- transition-timeout необходимо присвоить разумное значение, даже если используется ручной переход, то это значение желательно выставить на случай изменения значения параметра transition на auto.

Пример файла свойств агента с ручным переходом:

```
[server]
          control-agents=ManualAgent
          database=source
          transition=manual
          transition-timeout=1200
[control-agent. ManualAgent]
          name= ManualAgent
          database=target
          host=localhost
          port=4502
          connect-timeout=120
          replication-method=async
          critical=0
```

Для выполнения ручного перехода используется следующая команда:

```
dsrutil target-db-name -C transition Agent
```

Процесс перехода

Если произойдет сбой на source-базе, агент репликации может перевести target-базу в одно из двух состояний. Первое - это состояние source-базы, т.е. когда target-база станет source-базой. Второе состояние - это normal, т.е. когда target-база становится обычной базой данных без репликации.

Во время выполнения перехода агент фактически выполняет следующее:

1. все подключения к базе данных отвергаются;

2. все процессы, подключенные к target-базе, отключаются, чтобы предотвратить возможную блокировку ресурсов, необходимых менеджеру базы;
3. все активные транзакции завершаются или отменяются;
4. деактивируется OE Replication;
5. разрешается возможность подключения к базе данных;
6. теперь, уже обычная база данных, останавливается.

Сразу после выполнения этих действий база должна быть перезапущена. Как только source-машина станет доступна, изменения, выполненные в бывшей target-базе, должны быть скопированы на промышленную машину с использованием резервной копии.

Ручной накат AI-экстентов

При обычных условиях target-база автоматически обновляется данными, поступающими из source-базы. Изменения передаются блоками данных из AI-экстентов source-базы. Но после сбоя возможно, что последние блоки могут не достигнуть target-базы. Например, при сбое TCP/IP пакеты, ожидающие передачи, могут потеряться.

Есть возможность доставить эти данные вручную, используя AI-экстенты. Однако это можно сделать только при следующих условиях:

1. Параметр transition (секция [server] в файле свойств) должен иметь значение manual.
2. Агент репликации должен иметь статус pre-transition.
3. Устройство хранения, содержащее AI-экстенты, должно быть доступно на target-машине. Для этого AI-экстенты не должны храниться на той же машине, на которой расположена source-база, в случае разрушения системы они не будут доступны. Идеальным устройством для хранения AI-экстентов могут быть Network Area Storage (NAS) или Storage Area Network (SAN).

Для ручного наката AI-экстентов необходимо:

1. Отобразить информацию о failure recovery следующей командой:

```
dsrutil target-db-name -C recovery Agent
```

2. Определить номер AI-экстента по полученным данным. Выбрав последний примененный AI-экстент или, если последний экстент уже полностью обработан, выбрать следующий за ним номер. Ниже приведена необходимая часть информации, полученной командой RECOVERY:

```
Last AI Extent processed
AIMAGE BEGIN date:                Tue May 17 16:08:31 2010
AIMAGE NEW date:                  Tue May 17 16:28:23 2010
After Image File Number:          6
File Last Opened:                 Tue May 17 16:28:23 2010
Completely Applied to Target:     No
```

В этом примере последний обрабатываемый AI-экстент имеет номер 6 (After Image File Number). Поскольку поле Completely Applied to Target имеет значение No, то нужно использовать 6 в качестве значения extent-name из третьего шага. Если бы поле содержало значение Yes, то нужно было бы использовать следующее по порядку значение, т.е. 7.

3. Используйте полученный номер AI-экстента в следующей команде:

```
dsrutil target-db-name -C ApplyExtent extent-name
```

При этом команда проверяет следующее:

- Наличие экстента и его актуальность.
- Статус экстента, который должен быть FULL, BUSY или LOCKED.
- Номер AI-экстента, который должен соответствовать номеру AI-экстента последнего обработанного агентом AI-блока, или же должен иметь следующий номер, если предыдущий уже был обработан.

После наката AI-экстента можно выполнить ручной переход target-базы в рабочее состояние.

Активация OE Replication после перехода

После выполнения перехода можно восстановить работу базы на исходной машине, используя target-базу. Чтобы это сделать, необходимо использовать процесс failback или воспользоваться следующими командами:

1. Остановить target-базу командой:

   ```
   proshut target-db-name –by
   ```

2. Создать резервную копию этой базы:

   ```
   probkup target-db-name target-db-name.bak
   ```

3. Скопировать полученный файл target-db-name.bak на source-машину.
4. Отключить OE Replication на source-базе:

   ```
   proutil source-db-name -C disablesitereplication source
   ```

5. Восстановить source-базу из резервной копии target-базы:

   ```
   prorest source-db-name target-db-name.bak
   ```

6. Включить механизм after-imaging для source-базы:

   ```
   rfutil source-db-name -C aimage begin
   ```

7. Включить OE Replication для source-базы:

   ```
   proutil source-db-name -C enablesitereplication source
   ```

8. Удалить файл *source-db-name.repl.recovery*.
9. Включить механизм OE Replication для target-базы:

   ```
   proutil target-db-name -C enablesitereplication target
   ```

10. Удалить файл *target-db-name.repl.recovery*

Теперь обе базы готовы для нормальной работы. Запустите их.

Если имелась вторая target-база, которая не была настроена для выполнения transition, можно повторно активировать OE Replication и для неё, в качестве второй target-базы. Для этого необходимо:

1. Отключить OE Replication для второй target-базы:

```
proutil target2-db-name -C disablesitereplication target
```

2. Восстановить базу, используя резервную копию первой target-базы:

```
prorest target2-db-name target-db-name.bak
```

3. Активировать OE Replication:

```
proutil target2-db-name -C enablesitereplication target
```

4. Удалить файл *target2-db-name.repl.recovery*

Теперь OE Replication может работать для трёх баз данных (source, target1 и target2).

Что такое Failback?

При активировании OE Replication начинается копирование первичной базы данных в одну или, возможно, две вторичные базы, расположенные на отдельных машинах. В случае возникновения сбоя в первичной базе или машины, на которой она расположена, деятельность базы данных может быть переведена на одну из вторичных базы. Работа такой базы будет продолжаться до тех пор, пока не будет восстановлена первичная база или машина, на которой она расположена. Обычно сбой происходит из-за потери связи между сервером репликации первичной базы и агентом репликации вторичной базы. В этом случае агент репликации переходит в состояние *pretransition*, в котором он ожидает, пока сервер репликации не переподключится к нему, либо пока не будет выполнен переход (transition) вручную или автоматически. Сразу после выполнения перехода вторичной базы данных в исходную базу, вся активность базы будет переведена на эту базу, это называется *failover*. Как только первичная машина/база окажется доступной, все изменения будут возвращены назад, этот процесс называется *failback*. Процесс Failback – это, по существу, тот же failover, только от вторичной машины к первичной. По завершении этого процесса все изменения будут снова выполняться на первичной базе данных, а вторичная база вернется в режим read-only.

С введением процесса failback в OE Replication важно знать связанную с ним терминологию. Эти термины уже описывались ранее, но сейчас о них будет рассказано более подробно.

Primary database - это база данных, изменяемая рабочими приложениями, именно для нее в дальнейшем активируется OE Replication, в результате чего она становится *source databas*e. Во время failure recovery роль этой базы меняется от source к target и обратно. Тем не менее, хотя роль базы данных может быть изменена, сама база по-прежнему остается primary database.

Secondary database - это начальная копия первичной базы данных, которая в дальнейшем, при включении OE Replication становится *target database*. Во время failure recovery роль secondary database меняется от target к source базе и обратно к target.

Source database - это база данных, непосредственно изменяемая пользователями. База данных не может считаться source, пока не будет активирован механизм OE Replication. Именно из этой базы сервер репликации копирует данные в target database.

Target database - это точная копия source database. Для дополнительной избыточности может быть создано две target database. Target database содержит те же данные, схему и логическую структуру, что и source database. Изменять target database может исключительно агент репликации.

Ключевым компонентом failover и failback является процесс, известный как transition. Transition - это изменение роли базы для выполнения failover при появлении сбоя на primary-машине (на которой находится source-база) или на secondary-машине, когда primary-машина снова становится доступной.

Target-база может быть переведена в normal или source-базу, при этом база может находиться как в online, так и в offline. Если target-база переходит в source-базу, к базе должны быть добавлены AI-экстенты и включен механизм after-imaging. Если target-база переходит в состояние normal, то база станет доступна для операций изменения, но тогда она никогда не сможет перейти в состояние source-базы.

Source-база может перейти в состояние target-базы. База также может находиться в online или в offline. Если переход выполняется в online, то должен быть выполнен процесс transition failover.

Source-база может перейти в состояние normal, если предварительно будет отключен OE Replication

После успешного завершения перехода база данных может быть автоматически запущена (если будут сделаны соответствующие настройки).

Поведение процесса transition зависит от таких факторов как:

- Статус базы данных - source или target
- Находится ли база в online или в offline
- Завершен предыдущий transition успешно или нет
- Запущен сервер репликации или агент
- Время, когда произошел сбой в работе репликации

Все операции перехода могут выполняться автоматически. Одни автоматические процессы контролируется настройкой в файле свойств, другие основываются на статусе переводимой базы данных. Операции перехода включают следующие автоматические функции:

- Остановку переведенной базы
- Перезапуск базы данных
- Добавление AI-экстентов
- Запуск after-imaging

Эффективность работы процесса transition зависит от того, как хорошо будет продуман план его реализации. В противном случае полное выполнение перехода не сможет быть выполнено. Вместо этого переход будет ограничен обычной сменой роли базы с target на source.

План перехода можно составить, ответив на следующие вопросы:

- Какая база будет primary, а какая secondary?
- Какой тип перехода выбрать? Определить, будет ли target-база переведена в состояние source или normal?
- Какие функциональные возможности after-imaging должны быть осуществлены? Можно выбрать автоматическое добавление AI-экстентов и старт механизма after-imaging.
- Какой тип резервного копирования выполнять во время перехода?
- Нужен ли автоматический перезапуск базы после завершения перехода?
- Будет ли выполняться автоматическое восстановление в случае сбоя в процессе перехода? (для этого нужно иметь резервную копию базы).

После ответов на эти вопросы можно приступить к настройке файла свойств репликации.

Как работает переход

Работа механизма OE Replication осуществляется между сервером репликации source-базы и агентом репликации target-базы. Если агент репликации теряет соединение с сервером репликации, то выполняется failover, т.е. перевод target-базы в статус source-базы. Как только первичная машина станет доступна, необходимо перенести все изменения в базе с вторичной машины на первичную машину. Для минимизации времени простоя можно настроить процесс failback.

Работа механизма OE Replication заключается в следующем:

1. При нормальной репликации primary database играет роль source-базы, а secondary database роль target-базы. При этом на source-базе работает сервер репликации, а на target - агент репликации.

2. Если возникает сбой на машине, где работает primary database, агент репликации на target-базе теряет связь сервером репликации на source-базе.

3. Агент репликации переходит в состояние pre-transition.

4. Затем выполняется переход (transition), помечающий secondary database (формально target) базу как source.

5. Вся деятельность с primary database переходит на secondary database. Таким образом, теперь secondary database стала действующей промышленной базой данных, на которой и будет продолжена работа пользователей.

6. В это время машина с primary database восстановлена.

7. Инициируется переход, primary database на восстановленной машине меняет свою роль с source на target, а secondary database становится source.

8. Выполняется вторичная репликация, которая по существу является обратной первичной репликацией.

9. После этого в любое время может быть выполнен процесс failback, который вернет базам данных исходные роли, т.е. primary database опять становится source, а secondary database, опять становится target. Это может быть сделано одним из следующих способов:

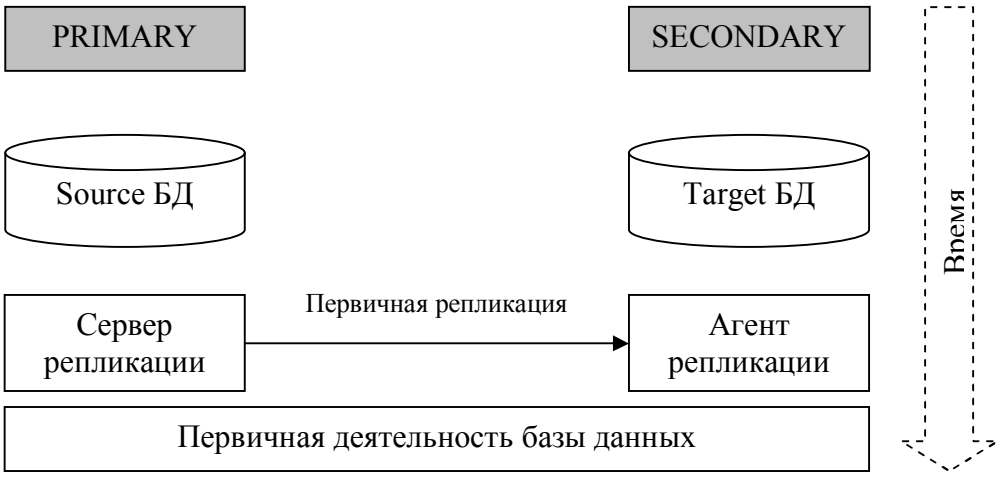
   - Переход вторичной базы с использованием команд failover.

   - Выполнение transition для primary database и затем отдельное выполнение transition для secondary database. Весь этот процесс известен как controlled transition.

Для большего понимания процесса далее представлено графическое описание перечисленных выше действий, где весь процесс разбит на последовательные шаги, которые выполняет OE Replication.

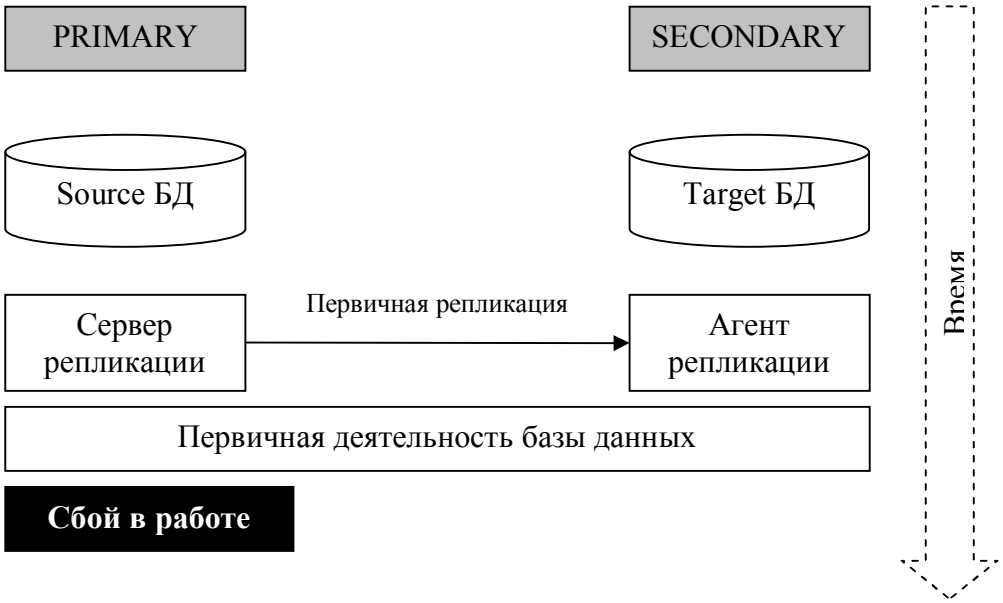
Шаг 1: Первичная репликация до сбоя

Выполняется обычное функционирование OE Replication:



Шаг 2: Сбой первичной машины

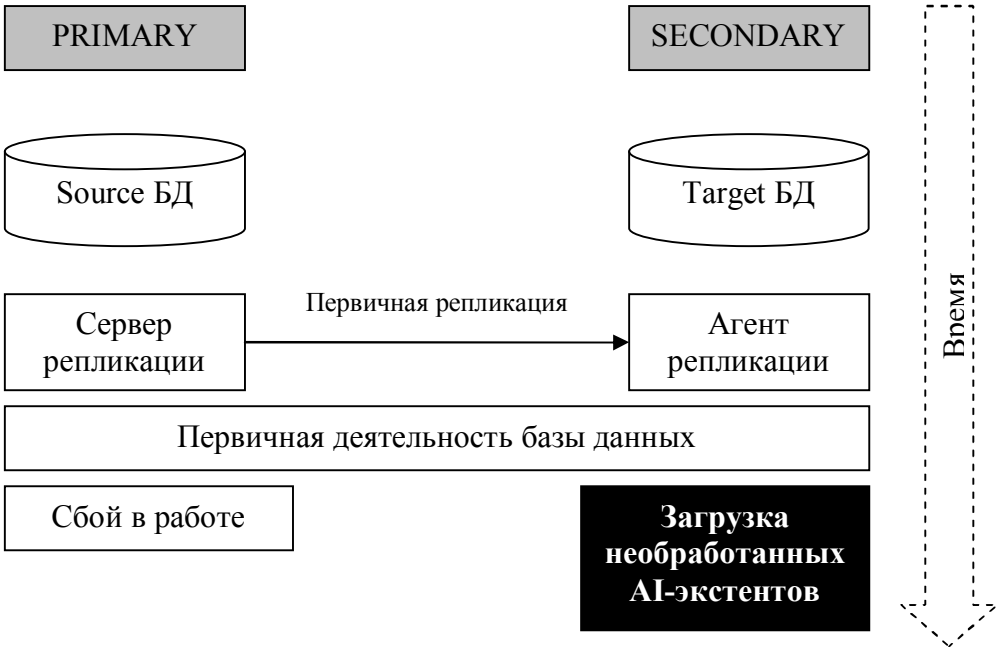
Работа репликации продолжается до сбоя на первичной машине



Агент репликации распознает сбой как ошибку TCP/IP-соединения. Обычно сбои могут происходить из-за потери соединения между первичной и вторичной машинами, аварийным отказом первичной базы данных или аварийным отказом аппаратных средств.

Шаг 3: Переход в состояние pre-transition

После сбоя агент репликации делает попытки восстановить работу OE Replication, тем не менее, если это невозможно, вся деятельность primary database переносится на secondary database. При этом важной задачей перед началом процесса transition является загрузка необработанных AI-экстентов в secondary database.



Обработка экстентов возможна только, если вторичная машина имеет доступ к этим экстентам. Например, экстенты могут располагаться на Storage Area Network (SAN).

Для загрузки AI-экстентов необходимо сделать следующее:

1. Использовать команду для определения первого AI-экстента для наката:

```
dsrutil target-db-name -C recovery
```

Результат этой команды покажет информацию о последнем AI-экстенте source-базы, который обрабатывался OE Replication на момент сбоя.

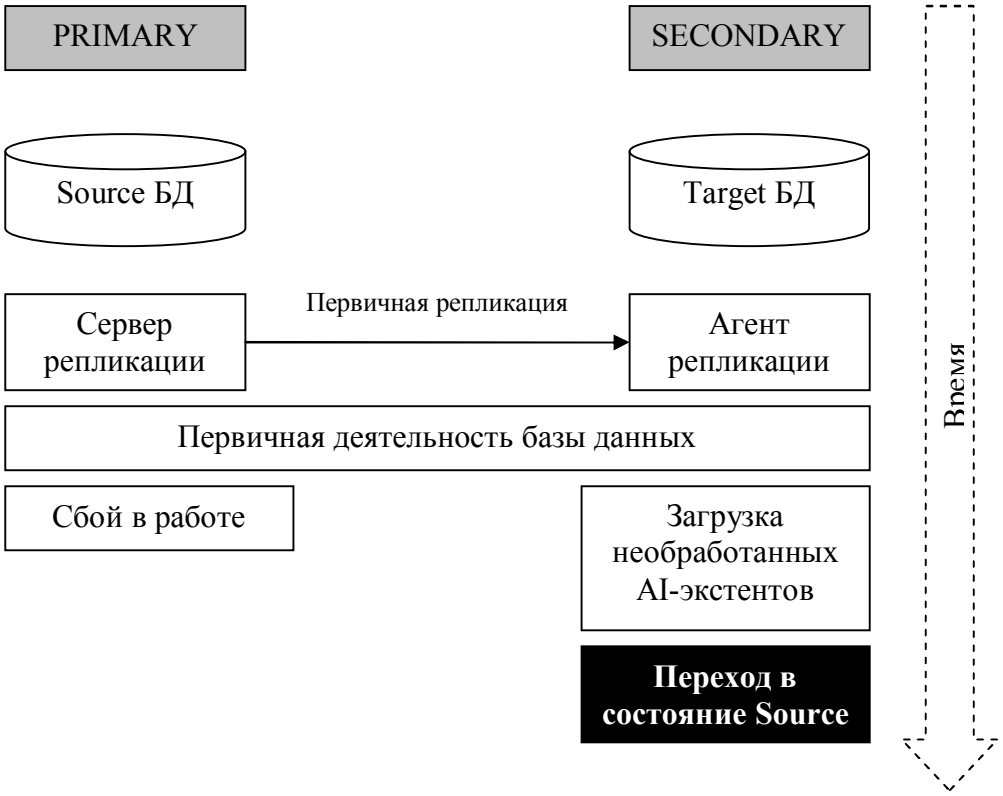
2. Начиная с этого экстента необходимо выполнить накат каждого AI-экстента до полного завершения наката изменений:

```
dsrutil target-db-name -C applyExtent qualified-extent-name
```

Если AI-экстенты не могут быть применены к target-базе, то она не будет содержать последних выполненных транзакций. В этом случае, если транзакции не могут быть восстановлены, то вы однозначно потеряете их после перехода.

Шаг 4: Переход target-базы в source.

Следующий шаг процесса восстановления - это переход target-базы в source-базу.



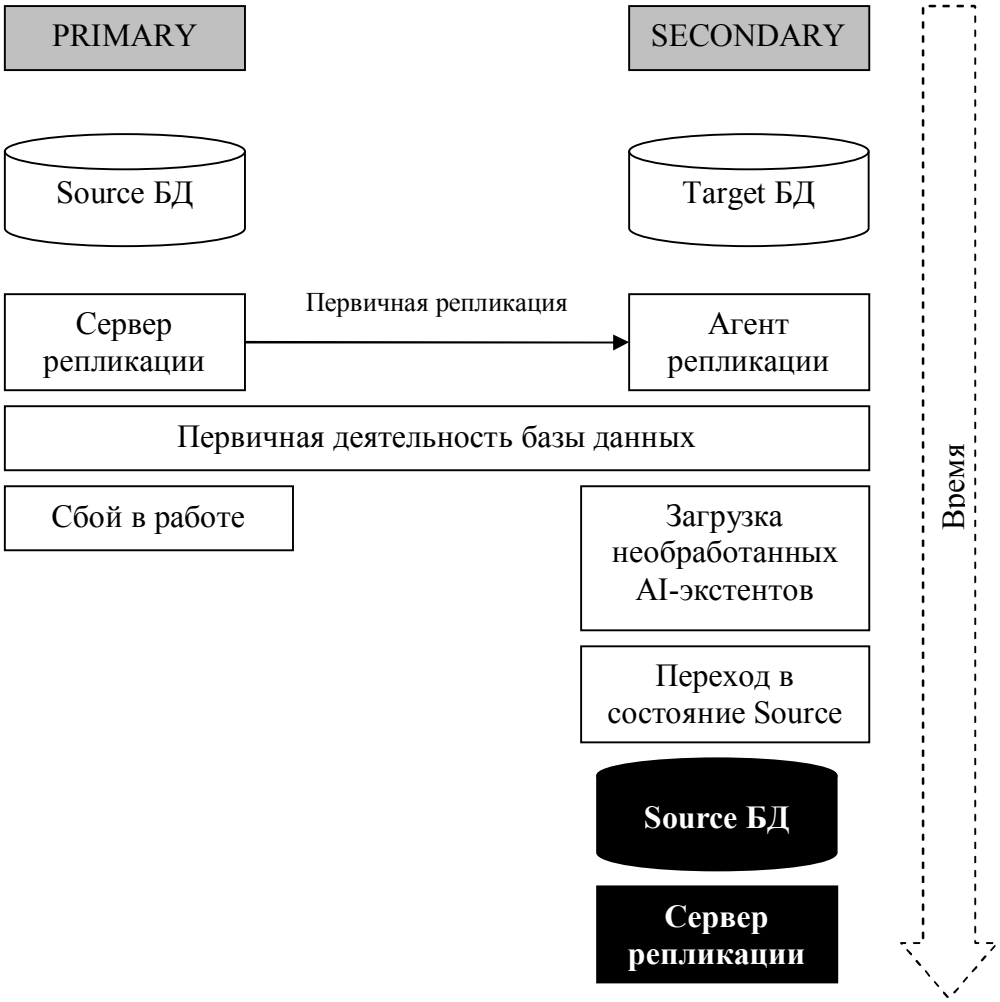
Для выполнения перехода необходимо правильно настроить свойства transition в файле свойств и выполнить следующую команду:

```
dsrutil target-db-name -C transition
```

Для того чтобы администратор не вводил вручную эту команду, можно настроить выполнение перехода в автоматическом режиме. Но в автоматическом режиме необработанные AI-экстенты не могут быть загружены в target-базу до выполнения перехода. Поэтому пользователи должны после завершения перехода вручную восстановить все потерянные изменения.

Шаг 5: Failover

После перевода secondary database в source-базу вся деятельность, которая выполнялась на первичной машине, теперь может выполняться и на вторичной машине.

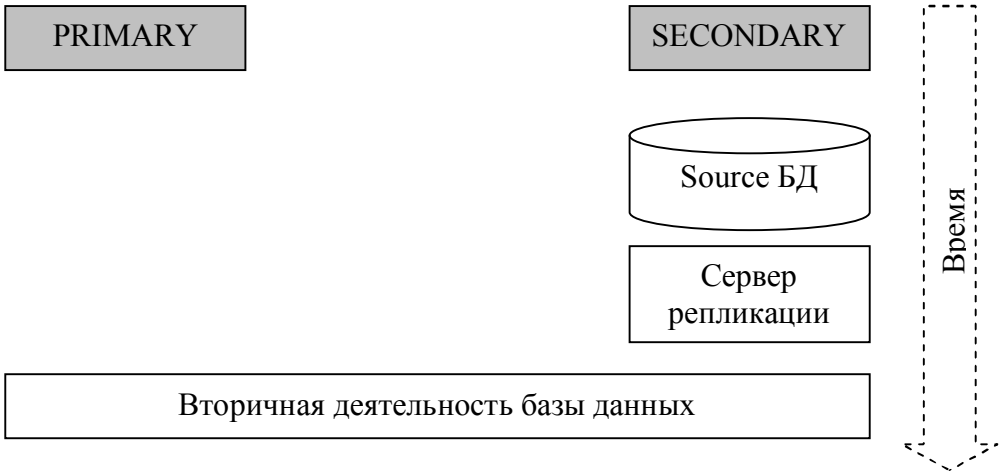


Переход можно настроить так, чтобы после его завершения secondary database была автоматически стартована. По умолчанию ее старт должен быть выполнен вручную.

Сервер репликации начнет работать сразу после запуска secondary database, ставшей source, и будет пытаться подключиться ко всем своим агентам. Для увеличения количества попыток подключения используйте параметр defer-agent-startup.

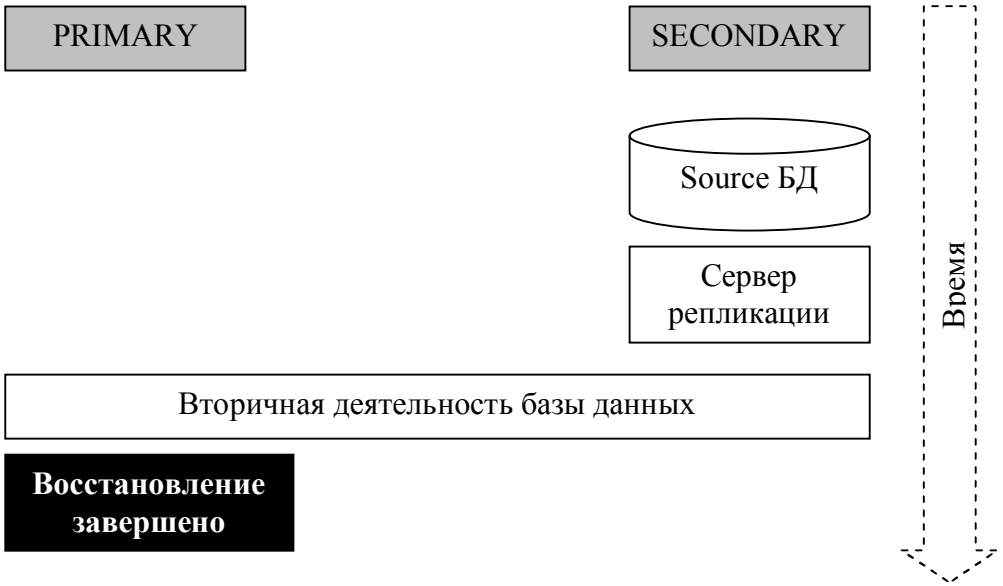
Если возможна ситуация, при которой первичная машина будет недоступна еще длительный период времени, то можно допустить завершение работы сервера репликации, поскольку он не сможет установить связь со своими агентами. Сервер репликации может быть перезапущен позднее, когда первичная машина окажется доступной.

После успешного перехода вся деятельность, выполнявшаяся ранее на первичной машине, может быть восстановлена на вторичной машине. В таком состоянии система может находиться сколько угодно времени. Тем не менее, необходимо позаботиться о скорейшем возвращении ее работы в нормальный режим.



Шаг 6: Завершение восстановления первичной машины

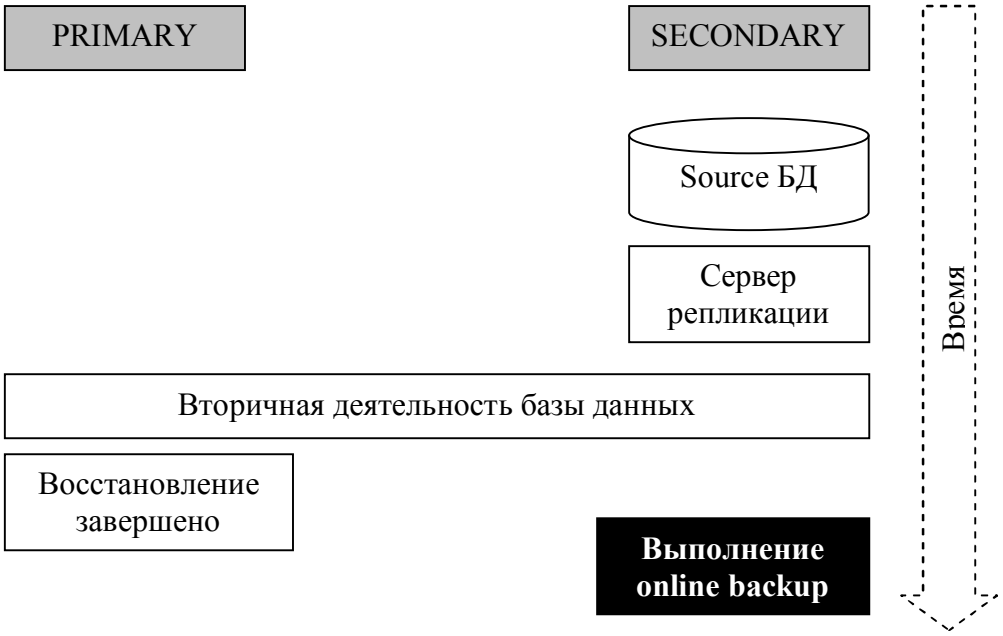
После восстановления и запуска первичной машины необходимо выполнить процесс failback.



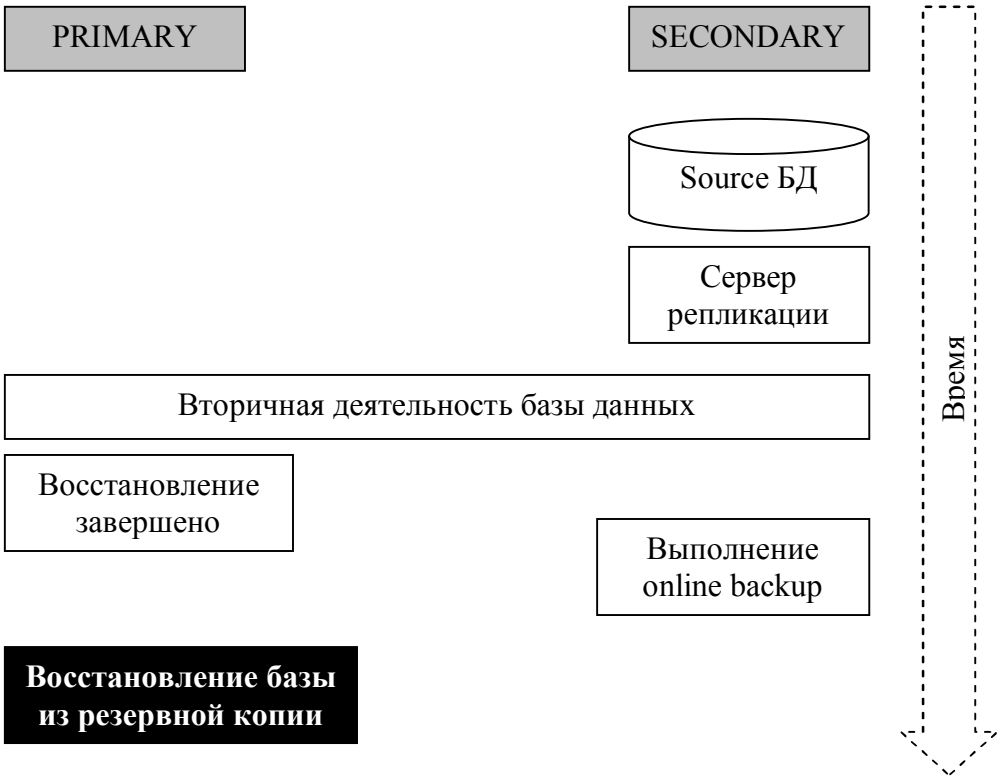
Шаг 7: Начало перехода primary database

Первым шагом к началу процесса failback является запуск репликации из secondary database в primary database. Этот процесс известен как secondary replication.

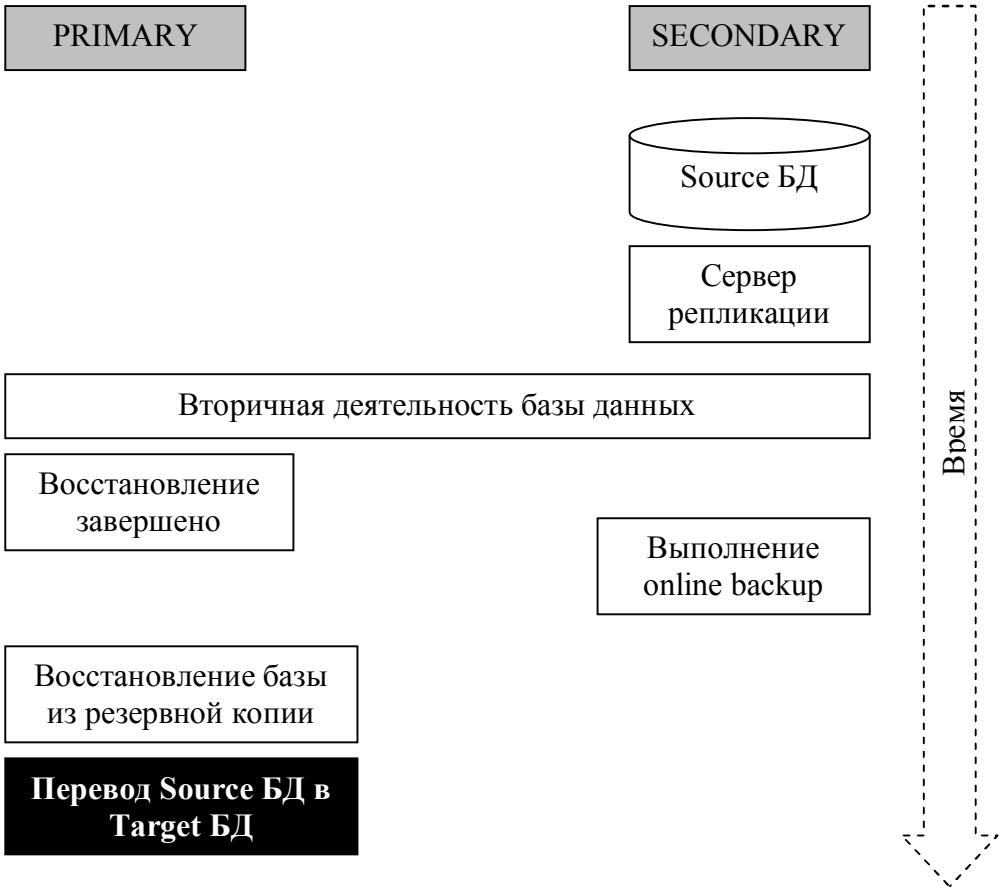
Вторичная репликация начинается с выполнения резервной копии вторичной базы данных, с использованием online резервного копирования в целях уменьшения времени простоя.



Файлы резервной копии должны быть скопированы на первичную машину или сохранены на общем носителе, а также они быть доступными для первичной и вторичной машины. После этого можно восстановить первичную базу данных.



Перед восстановлением базы данных следует убедиться в том, что PROREST будет использовать правильный структурный файл. Для этого на secondary database требуется обновить структурный файл командой PROSTRCT LIST и скопировать его на первичную машину. Теперь необходимо выполнить переход первичной базы из source в target.



Для этого можно использовать один из следующих методов:

- Переход в момент восстановления:

```
prorest primary-db-name backup-file -REPLTransition
```

- Переход после восстановления:

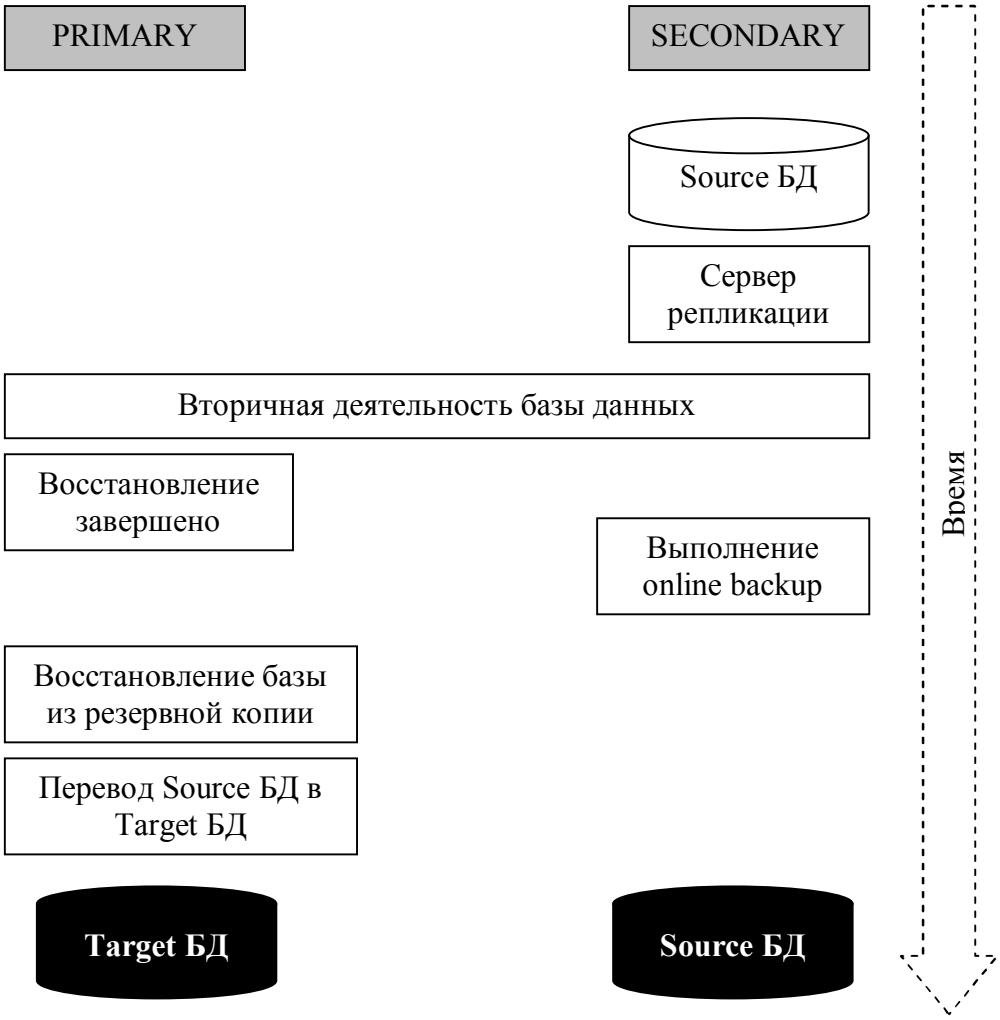
```
prorest primary-db-name backup-file
```

После перехода воспользуйтесь следующей командой:

```
proutil primary-db-name -C enableSiteReplication target
```

Независимо от выбранного метода первичная база станет target-базой, т.е. точной копией промышленной базы данных на вторичной машине.

В этой точке вторичная база является source-базой, а первичная база - это target-база.



Теперь можно начать выполнение secondary replication.

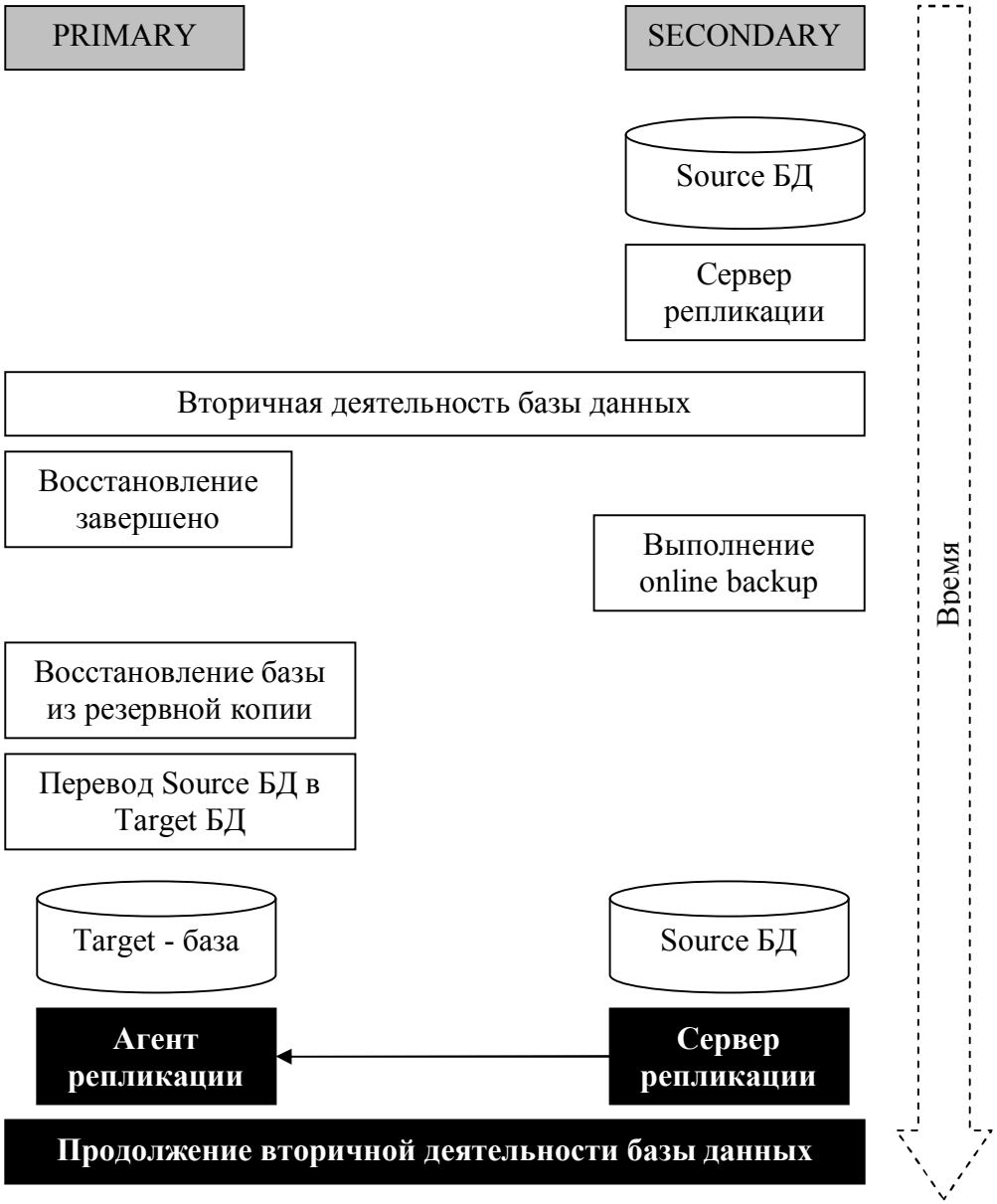
Шаг 8: Выполнение secondary replication.

Одновременно с запуском primary database будет запущен и агент репликации. Если сервер репликации был настроен с использованием параметра defer-agent-startup, и указанное время ожидания агента еще не завершено, то сервер репликации подключится к агенту при следующей попытке установления связи. Следует помнить, что сервер репликации может завершить свою работу, если истекло время, установленное параметром defer-agent-startup или параметром connect-timeout. Если сервер репликации завершит свою работу, то его нужно перезапустить командой:

```
dsrutil secondary-db-name -C restart server
```

После установления связи между сервером и агентом репликации начинается процесс вторичной репликации.

На следующем рисунке представлен процесс репликации на primary database всех изменений (она теперь является target-базой), сделанных на secondary database с момента формирования online копии.



Шаг 9: Failback

Выполнить failback промышленных процессов на первичный компьютер можно одним из следующих методов:

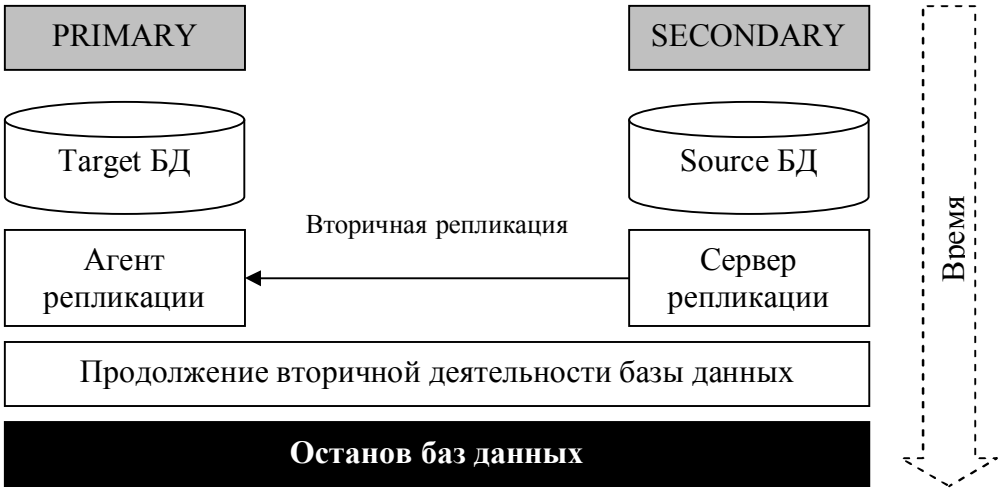
- Переход вторичной базы данных с использованием модификатора failover
- Выполнение контролируемого перехода, в котором сначала осуществляется переход первичной (primary) базы, и потом выполняется отдельный переход вторичной (secondary) базы.

Выполнение failback с использованием transition failover

При выполнении вторичной репликации обе базы данных запущены и выполняют репликацию всех транзакций вторичной базы в первичную базу.

Пока вторичная репликация работает, и сервер репликации с агентом активно выполняют копирование, необходимо определить наилучшее время для failback промышленного процесса на первичный компьютер.

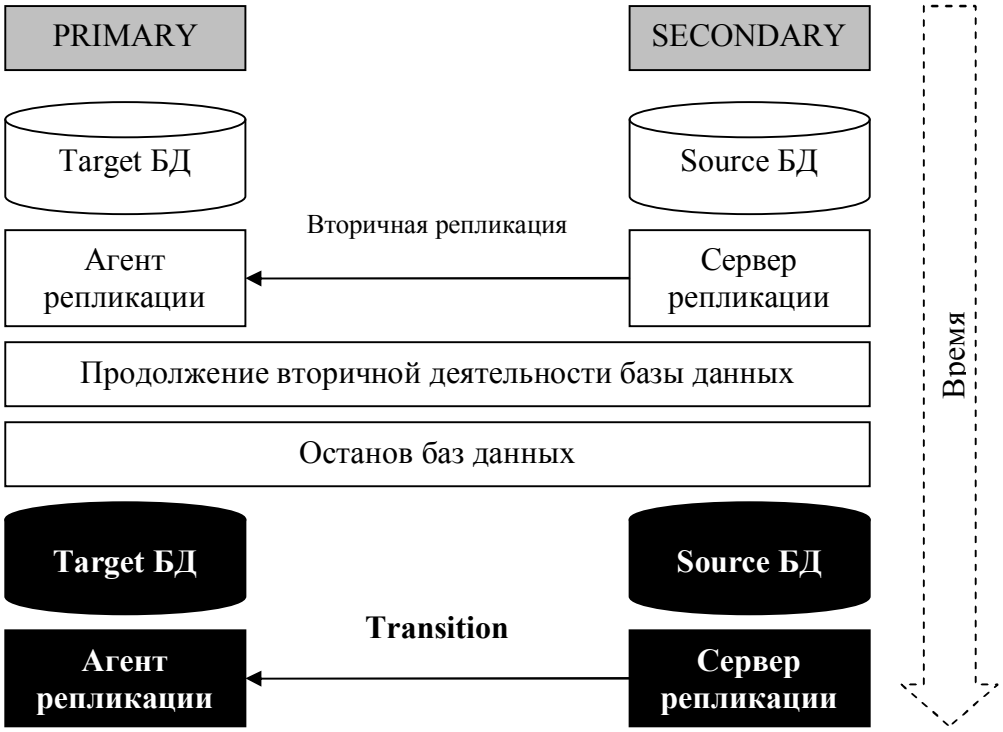
При запуске процесса failback ни один из пользователей не должен быть подключен к первичной или ко вторичной базе. Это условие можно обеспечить путем перезапуска обеих баз данных. Убедившись, что ни одного подключенного пользователя нет, процесс failback можно начинать. На рисунке представлен процесс останова баз данных.



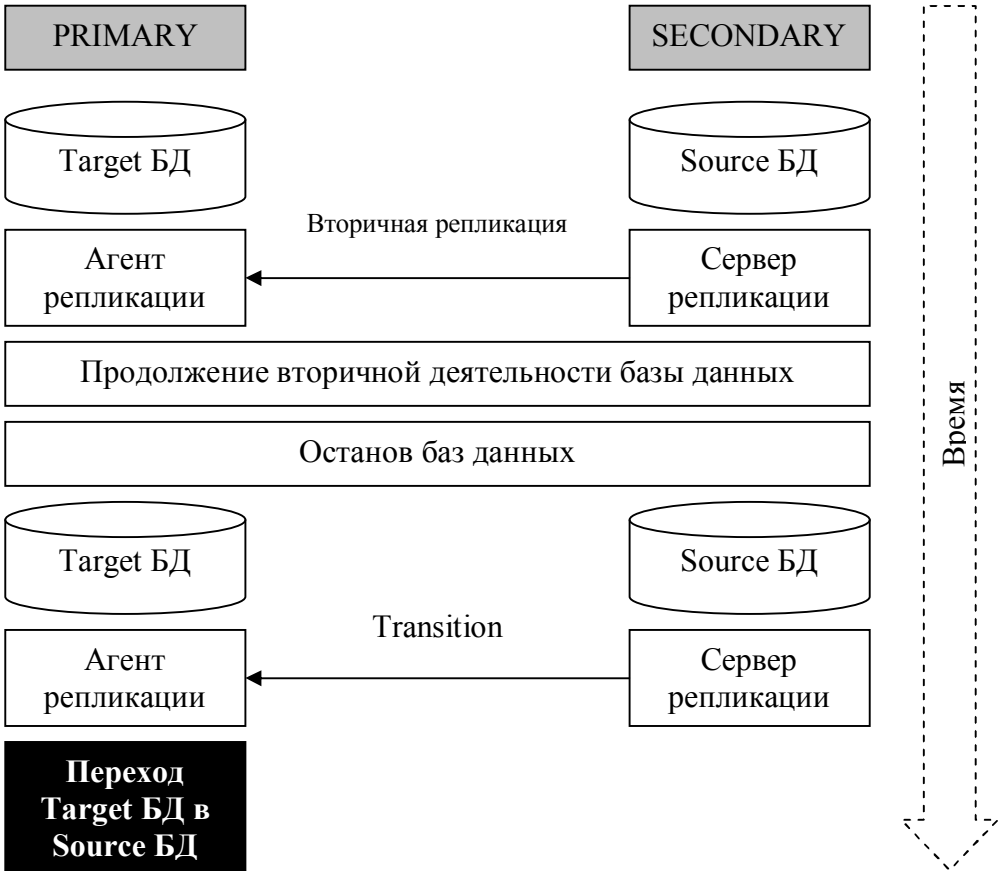
Для запуска процесса failback необходимо выполнить команду:

```
dsrutil secondary-db-name -C transition failover
```

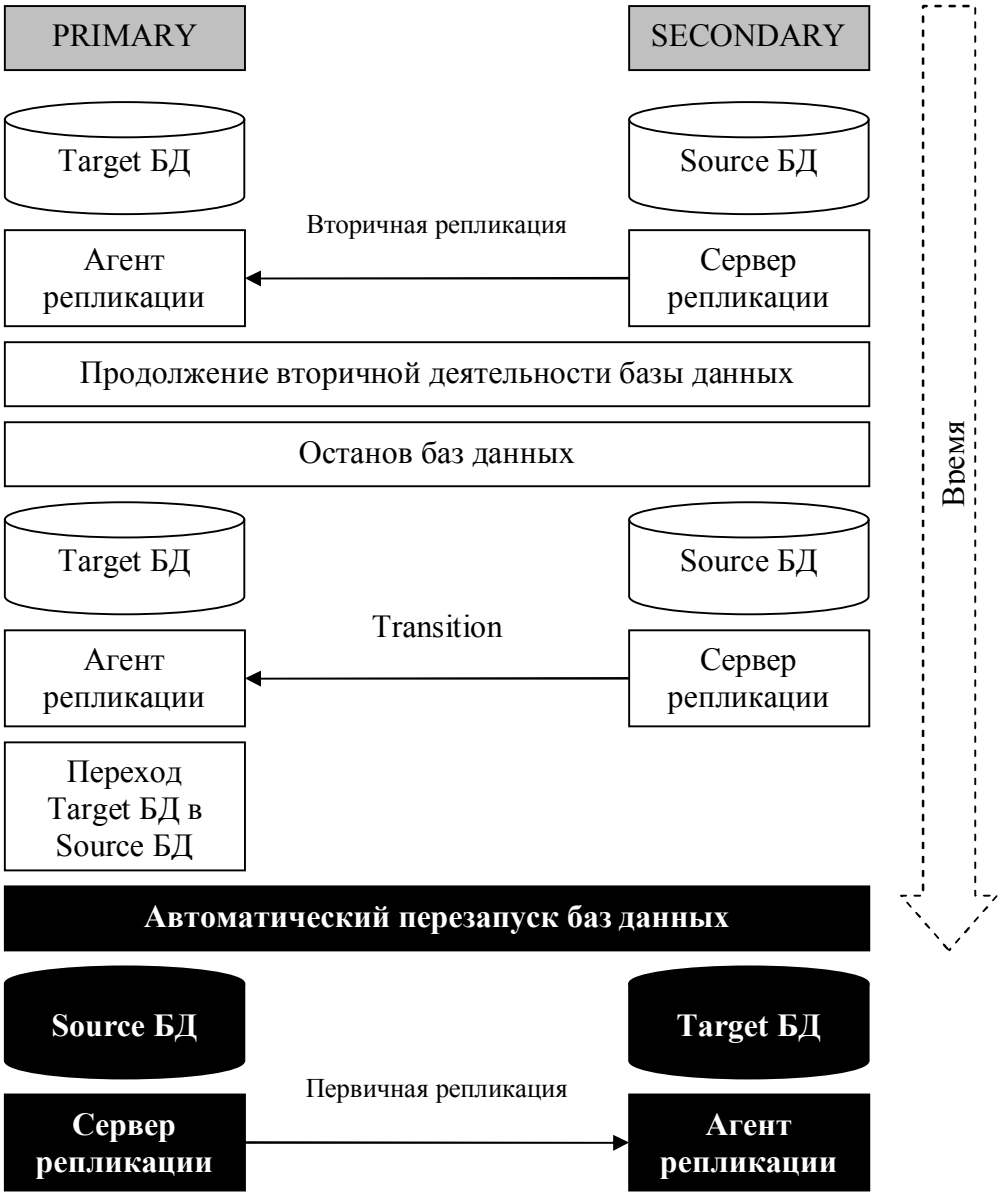
Модификатор failover указывает OE Replication, что это - failover transition, в результате чего процесс transition начнется на обеих базах (primary и secondary). Когда процесс transition начат, сервер репликации сообщает агенту репликации на первичной машине о том, чтобы началась подготовка первичной базу данных к началу transition на ней. В этой точке вторичная база данных будет остановлена и перезапущена. Процесс transition будет выполнен, как только завершится процесс синхронизации.



Переход первичной базы данных по завершению процесса синхронизации будет выполнен в соответствии с настройкой.



После завершения процесса перехода обе базы данных будут перезапущены, им будут присвоены новые роли.



Базы данных поменялись ролями. Первичная база данных теперь снова стала промышленной, а вторичная база снова стала её копией. Первичная репликация теперь выполняется так же, как и выполнялась до возникновения сбоя, а сервер репликации вновь копирует транзакции во вторичную базу данных.

Преимущества и недостатки использования transition failover для выполнения failback

В использовании transition failover в процессе failback существуют как преимущества, так и недостатки.

Преимущества	Недостатки
Если transition настроен должным образом, то можно инициализировать весь процесс failback, выполнив только одну команду. В этом случае не понадобится выполнять дополнительные ручные команды, и весь процесс можно запустить с одной машины.	Процесс перехода рассматривается как «всё или ничего». Это означает, что если произойдет какой-либо сбой во время перехода на любой из баз, то базы будут восстановлены в их оригинальное состояние, если резервное восстановление было должным образом настроено. В результате первичная база данных останется target, а вторичная - source.

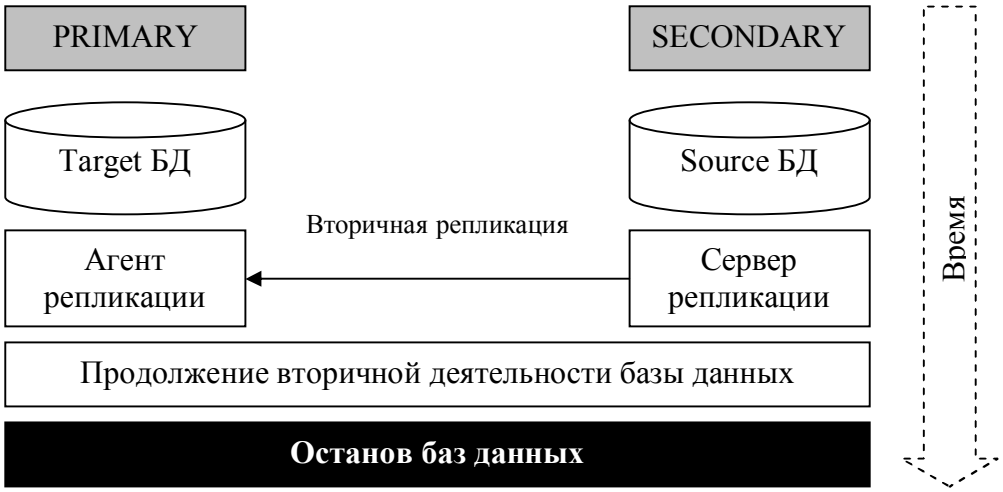
Преимущества	Недостатки
После завершения процесса перехода роли баз данных будут восстановлены, базы будут перезапущены и продолжится их обычная работа.	Обе базы должны находиться в offline в течение всего процесса перехода. Если произойдет сбой во время перехода на первичной базе, то эта база должна быть восстановлена для возврата к начальному состоянию, а может занять определенное время, в течение которого обе базы будут недоступны.

Для уменьшения времени простоя в случае сбоя при переходе можно использовать, так называемый управляемый переход (controlled transition). Но следует помнить, что управляемый переход требует большего вмешательства администратора базы данных, а так же, что необходим доступ к первичной и вторичной машинам.

Выполнение failback с использованием управляемого перехода

При выполнении вторичной репликации обе базы данных запущены, и все транзакции копируются в первичную базу данных.

Как и ранее для начала процесса failback необходимо, чтобы все пользователи были отключены, поэтому обе базы должны быть остановлены. Рекомендуется проверить все настройки transition для первичной и вторичной базы данных.



Нельзя запускать базы данных после перехода, т.к. необходимо будет выполнить дополнительные действия на случай сбоя во время перехода. Убедившись, что подключенных пользователей нет, можно начать процесс failback.

Для начала процесса failback с контролируемым переходом необходимо:

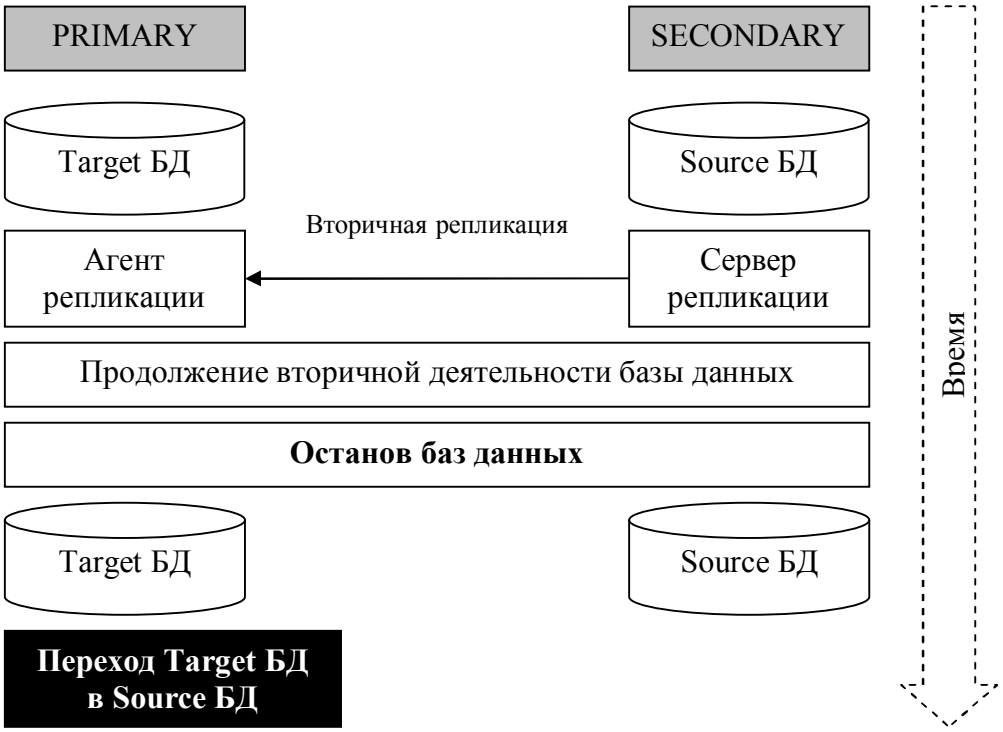
1. Остановить и перезапустить обе базы данных. Этим будет гарантирован сброс всех изменений в source-базе на диск и репликация их на target-базу.
2. Проверить завершение процесса синхронизации баз данных, для этого:
   a. Можно использовать лог файл баз
   b. Можно использовать команду:

      `dsrutil` *`source-db-name`* `-C status –detail`

      Если будет возвращено число 3049 – обе базы синхронизированы.
3. Снова остановить базы данных.
4. Для первичной базы запустить команду:

   `dsrutil` *`primary-db-name`* `-C transition`

   Эта команда переведет первичную базу данных в состояние source.



Если в процессе перехода произойдет сбой, можно перезапустить вторичную базу данных для продолжения промышленной работы. Позже можно попытаться запустить процесс failback повторно.

Если переход первичной базы в состояние source был успешно завершен, следует выполнить переход вторичной базы данных.

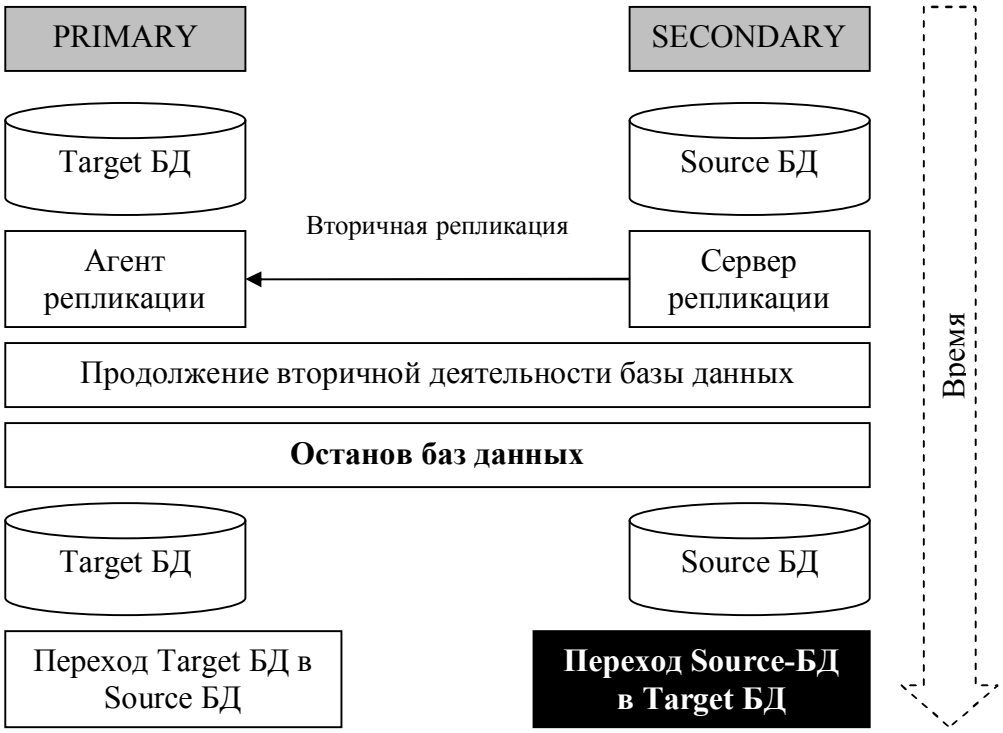
5. На вторичной машине выполнить переход с помощью команды:

```
dsrutil secondary-db-name -C transition
```

Если произойдет сбой во время перехода на вторичной базе, то можно запустить уже имеющуюся source-базу в качестве промышленной, что позволит исключить время простоя и продолжить работу системы.

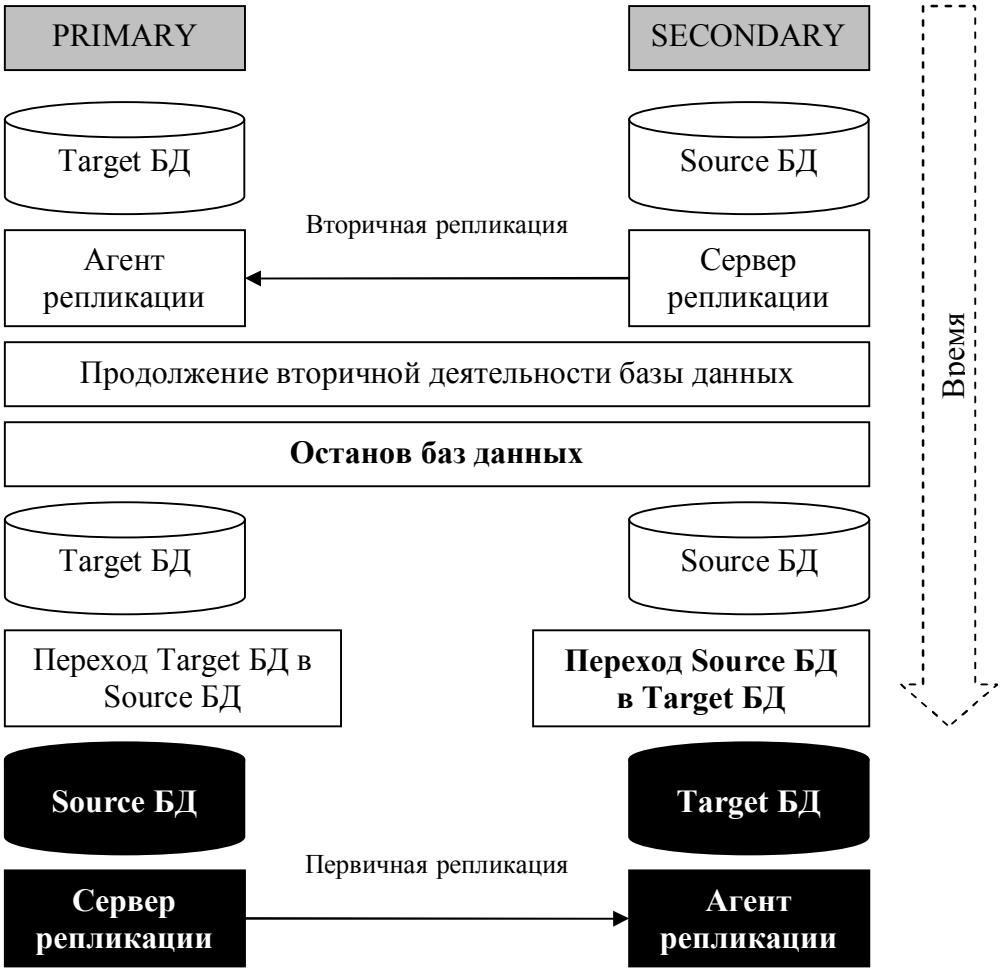
Если произошел сбой во время перехода, и первичная база уже запущена, то необходимо попытаться завершить процесс перехода вторичной базы путем повторного выполнения команды, указанной выше.

После того как команда успешно отработает, переход вторичной базы в состояние target завершен.



В этой точке роли обеих баз меняются: первичная база стала снова промышленной базой, а вторичная стала снова ее копией.

Первичная репликация теперь снова работает, т.е. как и до сбоя сервер репликации копирует транзакции во вторичную базу.



Достоинства и недостатки выполнения failback с использованием управляемого перехода

Достоинства	Недостатки
Время простоя в случае сбоя минимально.	Требуется выполнение дополнительных действий для завершения процесса failback
Полный контроль над процессом transition, т.е. когда и как будет выполнен переход обеих баз.	Необходимо иметь доступ, как к первичной, так и к вторичной машине.
Возможность обеспечения работы системы за счет вторичной базы в случае сбоя перехода первичной базы.	Требуется выполнение многочисленных остановок баз
Возможность использования первичной базы в качестве промышленной в случае сбоя на вторичной базе при переходе.	Необходим перезапуск обеих баз после завершения перехода.

Восстановление после сбоя при переходе

В большинстве случаев, если произойдет сбой во время перехода (transition failure) должен быть выполнен так называемый recovery backup, который обеспечит приведение базы данных в исходное состояние. Чтобы иметь возможность выполнить такое восстановление, процесс transition должен немедленно выполнить recovery backup до того, как база данных будет необратимо изменена. Например, во время выполнения перехода target-базы все незавершенные к этому времени транзакции должны откатиться (rolled back). Если не было сделано резервной копии базы до того, как транзакции откатились, то такой откат транзакций называется необратимым изменением.

Параметр перехода recovery-backup-arguments из файла свойств target-базы, позволяет определить параметры резервного копирования, используемые для recovery backup во время процесса transition. Но этот параметр не только устанавливает параметры резервного копирования, он также указывает процессу transition на необходимость выполнения recovery backup. Если параметр recovery-backup-arguments не используется, то transition не выполнит recovery backup. Процесс recovery backup будет выполнен до того, как будет выполнена первая необратимая операция изменения базы.

Процесс transition готовится к восстановлению следующим образом:

1. В каталоге базы данных создается новый файл с именем database.transition.recovery. Этот файл будет содержать всю необходимую информацию на случай, если понадобится выполнение восстановления.

2. Если установлены параметры резервного копирования, transition выполняет его с учетом следующих условий:

   - Если база данных является source, то прежде чем будут выполнены другие операции на базе, будет создана offline резервная копия. Если база находится в online, то перед началом резервного копирования она будет остановлена.

   - Если база является target и она находится в online, то выполняется online-копия, если база находится в offline, то, соответственно, выполняется offline-копия. В любом случае recovery backup будет выполнен прежде, чем иные операции процесса transition будут запущены.

Если ошибка произойдет во время выполнения перехода source-базы, вся контрольная и критичная информация базы данных будет перезаписана информацией, сохраненной до начала перехода. После чего база окажется в исходном состоянии, но она будет находиться в offline. Восстановление базы из recovery backup в этом случае не потребуется.

Если ошибка произойдет во время перехода target-базы, то для восстановления будут выполнены следующие действия:

- Если в базе были сделаны необратимые изменения и был выполнен recovery backup:

  - Текущему логу базы dbname.lg будет изменено имя на dbname.lg.save.

  - База данных будет удалена.

  - База будет восстановлена из recovery backup. Убедитесь, что в каталоге базы данных находится последний актуальный структурный файл .st.

  - Лог базы dbname.lg.save будет переименован обратно в dbname.lg.

  - Контрольная и критичная информация о базе данных будет перезаписана информацией, сохраненной до запуска процесса transition.

    База данных теперь находится в том же состоянии, в каком она была перед запуском процесса transition, но только будет остановлена.

- Если в базе были выполнены необратимые изменения, и при этом не создавался recovery backup, то процесс восстановления будет завершен с ошибкой, указывающей на невозможность продолжения восстановления базы данных.

Журнал выполнения transition

Стандартная информация о работе процесса transition отображается на экране в следующем виде, как представлено ниже:

```
Transitioning database /dir/srcdb
----------------------------------------------------------------
13:28:51 Opening database :                    Succeeded
13:28:51 Setting up transition :               Succeeded
13:28:53 Shutting down database :              Succeeded
13:29:25 Truncating BI :                S      ucceeded
13:29:28 Starting database in Cur Role :       Succeeded
13:29:44 Synchronization in process :          Succeeded
13:29:49 Replication Server processing :       Succeeded
13:29:54 Preparing to transition Target DB :   Succeeded
13:30:05 Shutting down database :              Succeeded
13:30:32 Target transition being performed :   Succeeded
13:31:14 Switching AI Extents :                Succeeded
13:31:20 Switching database role :             Succeeded
13:31:20 Updating database master block :      Succeeded
13:31:20 Comparing databases :                 Succeeded
13:31:20 Completing Target transition :        Succeeded
13:31:47 Backing up database :                 Succeeded
13:31:52 Starting database in New Role :       Succeeded


The Transition of this database has completed normally.
```

Эта информация отправляется в stdout программой rprep. В данном случае это пример успешного перехода.

Если же переход завершился ненормально, то журнал будет выглядеть так, как приведено ниже:

```
Transitioning database /dir/srcdb
----------------------------------------------------------------
08:12:00 Opening database :                    Succeeded
08:12:00 Setting up transition :               Succeeded
08:12:03 Shutting down database :              Succeeded
08:12:37 Truncating BI :                       Succeeded
08:12:40 Starting database in Cur Role :       Succeeded
08:12:56 Synchronization in process :          Succeeded
08:13:01 Replication Server processing :       Succeeded
08:13:06 Preparing to transition Target DB :   Succeeded
08:13:16 Shutting down database :              Succeeded
08:13:44 Target transition being performed :   Failed with -241

The transition of this database failed. Attempting recovery.

08:14:22 Retrieving prior Recovery Control :   Succeeded
08:14:22 Opening database :                    Succeeded
08:14:22 Restoring prior Recovery Control :    Succeeded
08:14:22 Updating Replication Control Info :   Succeeded

The Transition of this database failed, but recovery was successful.
```

Дополнительно к этим данным записывается информация в лог-файл базы данных (.lg) и, если указан параметр *–logging* утилиты DSRUTIL, то и в отдельный файл с именем database.repl.util.lg:

```
dsrutil database -C transition [failover] [-logging 2]
```

Этот лог имеет тот же формат, что и лог базы данных, но только с расширенными возможностями для диагностики. В данном случае предложен уровень логирования 2.

Свойства transition

Все свойства перехода могут быть сгруппированы в секции transition файла *db-name.repl.properties*, который должен находиться в том же каталоге, где и файл *db-name.db*.

Пример файла свойств primary database:

```
[server]
        control-agents=agent1
        database=ks1
        transition=manual
        transition-timeout=600
        agent-shutdown-action=recovery
        repl-keep-alive=0
[control-agent.agent1]
        name=agent1
        database=target
        host=localhost
        port=6931
        connect-timeout=120
        replication-method=async
        critical=0
[agent]
        name=agent1
        database=ks1
        listener-minport=4387
        listener-maxport=4500
        repl-keep-alive=0
[transition]
        database-role=reverse
        responsibility=primary
        auto-begin-ai=1
        auto-add-ai-areas=1
        transition-to-agents=agent1
        ai-structure-file=ks1.addai.st
        restart-after-transition=1
        source-startup-arguments=-DBService replserv
        target-startup-arguments=-S 6931 -DBService replagent
        recovery-backup-arguments=primary.recovery.bak
```

Пример файла свойств secondary database:

```
[agent]
       name=agent1
       database=target
       listener-minport=4387
       listener-maxport=4500
       repl-keep-alive=0
[server]
       control-agents=agent1
       database=ks1
       transition=manual
       transition-timeout=600
       defer-agent-startup=60
       agent-shutdown-action=recovery
       repl-keep-alive=0
[control-agent.agent1]
       name=agent1
       database=target
       host=localhost
       port=6931
       connect-timeout=120
       replication-method=async
       critical=0
[transition]
       transition-to-agents=agent1
       responsibility=secondary
       database-role=reverse
       auto-begin-ai=1
       auto-add-ai-areas=1
       ai-structure-file=ks2.addai.st
```

```
restart-after-transition=1
source-startup-arguments=-DBService replserv
target-startup-arguments=-S 6931 -DBService replagent
backup-method=full-offline
backup-arguments=ks2.sav
incremental-backup-arguments=ks2.sav.inc
recovery-backup-arguments=!secondary.recovery.bak
```

Для настройки свойств перехода можно воспользоваться вопросами и ответами из следующей таблицы:

Вопросы для настройки	**Необходимо …**
Какая база primary, а какая secondary?	Изменить свойство *responsibility* для определения типа базы данных. Возможные значения: • Primary – первичная база данных • Secondary – вторичная база данных
Какой тип перехода будет выполняться?	Настроить свойство *database-role* для указания новой роли базы данных после перехода. Возможные значения: • Reverse – роль базы данных будет меняться обратно текущему ее состоянию, т.е. source-база данных становится target, а target становится source • Normal – после перехода репликация не будет активирована. Это значение по умолчанию.
Какой тип операций с after-imaging должен быть выполнен во время перехода?	Изменить свойства a*uto-begin-ai, auto-add-ai-areas* и *ai-structure-file*. Для auto-begin-ai: • 0 – не включать AI автоматически после перехода target в source • 1 – автоматически включить AI после перехода target в source Для auto-add-ai-areas: • 0 – не добавлять AI-экстенты автоматически (когда база данных переходит в source или когда AI-экстенты уже имеются для этой базы) • 1 – добавлять AI-экстенты автоматически, используя структурный файл, указанный свойством ai-structure-file. Имя структурного файла, содержащего список AI-экстентов для добавления, указать в свойстве ai-structure-file.
Какой тип резервной копии необходимо выполнить во время	Для определения типа резервной копии, которая должна быть выполнена во время перехода и до включения AI, использовать свойства backup-method, backup-arguments и

Вопросы для настройки	**Необходимо …**
перехода?	incremental-backup-arguments. backup-method: • mark - пометить базу данных как скопированную с использованием команды “rfutil db-name –C mark backedup”. Но этот метод в будущем не позволит использовать AI-экстенты для восстановления после сбоя. • full-offline – резервное копирование базы данных в offline с помощью утилиты PROBKUP. Резервное копирование будет выполнен в два шага. Первый – выполнение full backup до включения AI. Второй – incremental backup, после включения AI и смены роли базы данных. • full-online – выполнение online backup после перезапуска базы данных, который будет выполнен после перехода базы данных. Для указания параметров резервного копирования (offline/online) использовать свойство backup-arguments: • Как минимум нужно указать итоговый файл или устройство для сохранения копии. • Во избежание перезаписи, используйте различные имена файлов или устройств как для инкрементальной копии, так и для полной копии. • Нельзя использовать параметры проверки, такие как <-vp> и <-vf>. • Параметры должны начинаться с device-name. Свойство incremental-backup-arguments используется для параметров инкрементальной копии, выполняемой в offline после включения AI и после изменения роли базы данных. Первым параметром должен быть device-name.
Должна ли база данных быть автоматически перезапущена после перехода?	Чтобы указать, должна ли база данных автоматически быть перезапущена после перехода, измените следующие свойства перехода. Restart-after-transition: • 0 – не выполнять перезапуск базы данных. • 1 – автоматически перезапустить базу данных после перехода. Если вы определите это свойство, то должны так же указать все свойства *-startup-arguments, иначе произойдет сбой запуска базы данных.

Вопросы для настройки	Необходимо …
	Если база данных переходит в нормальное состояние, используйте свойство normal-startup-arguments. Это параметры, которые будут использоваться при запуске базы данных. Они будут добавлены к команде PROSERVE. В большинстве случаев единственным параметром, который здесь указывается, является –pf, который определяет файл параметров для запуска, например, так <-pf db-name.normal.pf> Если база данных переходит в состоянии source, необходимо использовать свойство source-startup-arguments. Эти параметры также будут использоваться для команды PROSERVE. И обычно это свойство тоже использует только один параметр –pf, <-pf db-name.source.pf>. Но, поскольку база переходит в source, необходимо также указать параметр <-DBService replserv> для запуска сервера репликации. Если база данных переходит в состояние target, используется свойство target-startup-arguments. Для него можно указать <-pf db-name.target.pf> и обязательно для обеспечения запуска агента репликации добавьте <-DBService replagent –S port-number или service-name>. Если база данных меняет свою роль (reverse), то необходимо указать два свойства: source-startup-arguments и target-startup-arguments.
Нужно ли в случае сбоя перехода автоматически выполнять восстановление?	Настроить свойство recovery-backup-arguments для указания процессу transition необходимости выполнить автоматическую попытку восстановления. Укажите device-name и прочие параметры для выполнения резервного копирования.

Пример стартового файла параметров

Переход может использовать стартовый файл параметров в следующих случаях:

- База данных перезапущена как source.
- База данных перезапущена как target.
- База данных перезапущена как normal.

При использовании перехода базы данных в состояние source следует указать имя файла параметров (.pf) в свойстве source-startup-arguments. Пример содержимого такого файла:

```
-DBService replserv [эта строка обязательна]
-B 10000 -n 1000
```

При использовании перехода базы данных в состояние target следует указать имя файла параметров (.pf) в свойстве target-startup-arguments. Пример содержимого такого файла:

```
-DBService replagent -S 6931 [эта строка обязательна]
-B 10000 -n 1000
```

При использовании перехода базы данных в состояние normal необходимо указать имя файла параметров (.pf) в свойстве normal-startup-arguments. Пример содержимого такого файла:

```
-B 10000 -n 1000
```

Мониторинг

Мониторинг работы OE Replication осуществляется с помощью утилиты DSRUTIL с параметрами monitor или status.

DSRUTIL MONITOR

При использовании параметра monitor можно визуально наблюдать за выполнением репликации. Для этого выполните следующую команду:

```
dsrutil db-name -C monitor
```

После выполнения команды на экране будет отображено следующее меню:

```
OpenEdge Replication Monitor                  Page 1
Database:  /testdb/sports

S.  Replication server status
R.  Replication server remote agents
A.  Replication agent status

M.  Modify display defaults
Q.  Quit

Enter your selection:
```

Здесь для использования меню необходимо в «Enter your selection» ввести соответствующую букву английского алфавита и нажать клавишу <Enter>.

Экран S (Replication server status) предоставляет информацию о статусе сервера репликации. Этот экран доступен только на стороне сервера. Например:

```
    Database:  /testdb/sports

    Database is enabled as OpenEdge Replication:  Source

    Server is:                                    In Normal Processing
    Number of configured agents:        2
    Delay Interval (current / min / max):      500 / 5 / 500
    Recovery information:
        State:                                    No recovery being performed
        Agents needing recovery:        0
        Agents connected:               0
        Agents in synchronization:             0
    Transition information:
        Type:                                  Manual

RETURN - repeat, U - continue uninterrupted, Q - quit:
```

Назначение полей экрана «Replication server status»:

Поле	Описание
Database is enabled as OpenEdge Replication	Показывает тип базы данных. Возможные значения: - Source или Target.
Server is	Состояние процесса обработки информации сервером репликации. Возможные значения: **Normal processing** – обработка информации выполняется в обычном режиме; **Performing initialization** – сервер репликации выполняет инициализацию; **Performing startup synchronization** - сервер находится в процессе синхронизации Target-базы данных с Source-базой; **Connection** – сервер репликации выполняет подключение к агентам репликации; **Configuring connected agents** – сервером выполняется подтверждение связи с агентом; **Performing failure recovery** – выполняется попытка аварийного восстановления после обрыва соединения; **Unknown** - сервер репликации находится в неизвестном состоянии.
Number of configured agents	Показывает количество агентов, настроенных для работы с сервером репликации.
Delay Interval (current / min / max)	Показывает время в миллисекундах, которое проходит между последовательными опросами source-базы сервером репликации для получения данных, которые должны быть скопированы на target-базы. Здесь: Current – текущее значение; Min – минимальное значение; Max – максимальное значение. Последовательный опрос используется с целью увеличения производительности сервера, а время между опросами может изменяться для того, чтобы ограничить частоту опросов, когда активность в базе данных небольшая и, соответственно, данных для копирования меньше или совсем нет.
Recovery information	Отображаются детали, связанные с восстановлением после сбоя.
Agents needing recovery	Количество агентов, требующих восстановления после сбоя.
Agents connected	Количество агентов, на текущий момент подключенных к серверу репликации.
Agents in synchronization	Количество агентов, выполняющих процесс синхронизации.
Transition information	Отображаются детали, связанные с процессом Transition

Поле	Описание
Type	Тип процесса Transition: Manual – указывает на то, что процесс Transition должен быть завершен в ручном режиме. Automatic – указывает на то, что процесс Transition будет завершен автоматически.
Transition timeout limit	Отображается оставшееся время до начала перехода target-базы в состояния обычной базы данных. Это поле будет отображено, только если тип Transition имеет значение Automatic.

На экране R (Replication server remote agents) отображена информация, связанная с агентами репликации. Например:

```
Database is enabled as OpenEdge Replication:  Source

Remote Agents Configured:

ID   Name      Host Name         Database
1.   agent1    10.0.10.20        /testdb/target1/sports
2.   agent2    10.0.10.30        /testdb/target2/sports

Q. Quit

Enter your selection:
```

Экран A (Replication agent status) предоставляет информацию об агенте репликации, используемом для target-базы данных, для которой был запущен монитор. Это меню доступно только на стороне агента репликации. Пример:

```
Agent:
Name:                                          agent1
ID:                                            1
Host name:                                     10.0.10.30
State:                                         Normal Processing
Ready:                                         Yes
Critical:                                      No
Method:                                        Asynchronous
Agent is waiting for:                          Nothing
Maximum bytes in TCP/IP message:  8504
Server/Agent connection time:                  Fri May 21 12:18:44 2010
Delay Interval (current / min / max):          240 / 5 / 500
Transition information:
Type:                                          Manual
The last block received at:                    Fri May 21 12:18:45 2010
Activity information:
Blocks received:                               1
Blocks processed:                              1
Blocks acknowledged:                           0
Notes processed:                               0
Transactions started:                          0
Transactions ended:                            0
Synchronization points:                        1
AI Block Information:
Source RDBMS Block (Seq / Block):              3 / 191
Last Processed Block (Seq / Block):            3 / 191
Latency Information:
Repl Server behind Source DB by:               0 second(s)
Current Source Database Transaction:           4678144
Last Transaction Applied to Target:            4678144
Target Current as of (Target, Source):
             Fri May 14 17:53:37 2010, Fri May 14 17:53:37 2010 with delta of 000:00:00
```

Назначение полей экрана «Replication agent status»:

Поле	Описание
Database	Полное имя базы данных, относительно которой запущен монитор.
Database is enabled as OpenEdge Replication	Показывает тип базы данных. Возможные значения: - Source или Target.
Agent	Основная информация об агенте.
Name	Имя агента.
ID	Идентификационный номер агента.
Host name	IP-адрес машины, на которой размещена target-база и запущен агент репликации.
State	Информация о состоянии агента репликации. Возможные значения: **Normal processing** – обработка информации выполняется в обычном режиме; **Initial connection** – агент ожидает инициализации подключения сервером репликации; **Startup synchronization** – выполняется синхронизация сервера и агента репликации; **Initialization** – выполняется инициализация агента сервером; **Initial connection failed** – сервер не смог подключиться к агенту; **Invalid target database configuration** – ошибка в настройке target-базы данных; **Agent terminated** – target-база остановлена или работа агента была завершена командой DSRUTIL TERMINATE AGENT; **Online backup of Target Database** – выполняется резервное копирование target-базы данных в online; **Recovery synchronization** – выполняется синхронизация после восстановления; **Recovery failed** – произошел сбой в процессе восстановления; **Unknown** – агент находится в неизвестном состоянии.
Critical	Если агент имеет статус Critical, то отображается значение Yes, иначе No.
Method	Отображается тип репликации: **Asynchronous** – асинхронная репликация; **Synchronous** – синхронная репликация.
Agent is waiting for	Показывает, почему агент находится в ожидании. Возможные значения: **Nothing** – агент не находится в ожидании; **Schema lock request** – сервер ожидает блокировки схемы агентом для ее изменения; **Encryption Objects** – ожидает шифрования объектов (при включенном механизме OpenEdge Transparent Data Encryption).

Поле	Описание
Maximum bytes in TCP/IP message	Максимальное количество байт для TCP/IP сообщений.
Server/Agent connection time	Дата и время подключения агента и сервера.
The last block received at	Время получения последнего блока данных агентом.
Blocks received	Количество полученных блоков.
Blocks processed	Количество обработанных блоков.
Blocks acknowledged	Количество блоков, подтвержденных агентом.
Synchronization points	Количество выполненных точек синхронизации.
AI Block Information	Информация о задержке в процессе обновления target-базы данных. Это полезно знать, когда target-база переходит в обычное состояние из-за сбоя на стороне source-базы.
Source RDBMS Block (Seq / Block)	Номер текущего AI-блока на source-базе данных, обрабатываемый агентом репликации. Seq - это порядковый номер AI-экстента.
Last Processed Block (Seq / Block)	Последний обработанный агентом AI-блок.
Latency Information	Информация об отставании target-базы от source-базы.
Repl Server behind Source DB by	Время отставания сервера репликации от source-базы в секундах.
Current Source Database Transaction	Номер текущей транзакции на source-базе данных, которую обрабатывает агент репликации.
Last Transaction Applied to Target	Номер последней обработанной транзакции на target-базе.
Target Current as of (Target, Source)	Время завершения последней транзакции на target-базе относительно времени завершения последней транзакции на source-базе.

DSRUTIL STATUS

Команда DSRUTIL STATUS подходит для написания контролирующих сценариев. Синтаксис команды следующий:

```
dsrutil db-name -C status agentname [-detail]
```

Параметр status возвращает числовой код, по которому можно определить состояние сервера или агента репликации. Добавление параметра <-detail> позволяет получить код, более детально отображающий, что происходит с сервером или агентом репликации. Параметр *agentname* - это имя агента репликации, по которому нужно получить информацию. Этот параметр не является обязательным и может не использоваться, т.к. его применение имеет смысл только тогда, когда сервер репликации работает с двумя агентами, и когда команда выполняется на стороне сервера репликации. Если команда выполняется на стороне агента репликации или на стороне сервера репликации, когда он работает только с одним агентом, то использование имени агента не требуется.

Возвращаемый код командой DSRUTIL STATUS без параметра <-detail> состоит из четырех цифр. При этом первая цифра может принимать только три значения, как представлено в таблице:

Первые цифры кода	Значение
1xxx	Сервер репликации (Source-база данных)
2xxx	Контрольный (критичный) агент репликации (Target-база данных)

Первые цифры кода	Значение
3xxx	Обычный агент репликации (Target-база данных)

Значения оставшихся трех цифр описаны в следующей таблице. Таблица составлена так, чтобы показать соответствие возвращаемых кодов, когда команда DSRUTIL STATUS выполняется без параметра <-detail> и с этим параметром.

Обычный код	Значение	Детализированный код	Значение
100	Connecting	1001	Initial connection.
100	Connecting	1002	Initializing.
100	Connecting	6001	Server Initialization.
100	Connecting	6002	Connecting to Agents.
100	Connecting	6003	Configuring Agent(s).
101	Processing is taking place.	3048	Startup Synchronization.
101	Processing is taking place.	3049	Normal Processing.
101	Processing is taking place.	3050	Recovery Synchronization.
101	Processing is taking place.	6004	Recovery Processing.
101	Processing is taking place.	6005	Startup Synchronization.
101	Processing is taking place.	6021	Normal Processing.
102	Activity is halted	3051	Online backup of the Target Database.
102	Activity is halted	3052	Target Database in Quiet Point.
102	Activity is halted	3053	Target Database is in a BI stall.
102	Activity is halted	3054	Target Database is in an AI stall.
103	Pre-transition state.	2080	Pre-transition.
103	Pre-transition state.	2081	Applying After-image Extent.
104	Agent is listening.	2083	Listening.
105	Transition.	2082	Transitioning.
199	Inactive.	1032	Initial Connection Failed.
199	Inactive.	1033	Recovery Failed.
199	Inactive.	1034	Invalid Target Database Configuration.
199	Inactive.	1035	Agent Failed.
199	Inactive.	1036	Agent is Ignored.
199	Inactive.	1037	Agent is Stopped.
199	Inactive.	1038	Agent is Terminated.
199	Inactive.	1063	Agent is Ended.
199	Inactive.	6060	Server is ended.
255	Unknown.	0	Unknown.

Пример части кода из скрипта, использующего команду DSRUTIL STATUS для идентификации состояния процесса репликации:

```
RETCODE=`dsrutil $a -C status -detail`
MESS="Replication error!"
STS="High"
case $RETCODE in
0) MESS=" Unknown!" STS="High";;
1001) MESS=" Initial connection." STS="Low";;
1037) MESS=" Agent is Stopped!" STS="High";;
3048) MESS=" Startup Sychronization." STS="Low";;
3049) MESS=" Normal Processing." STS="Normal";;
2080) MESS=" Pre-transition!" STS="High";;
6005) MESS=" Startup Synchronization." STS="Low";;
6060) MESS=" Server is ended!" STS="High";;
esac
```

Используя этот небольшой пример можно написать простую программу слежения за процессом репликации, которая будет автоматически запускаться через некоторые интервалы времени и сообщать администратору базы данных о состоянии сервера и агентов.

Помимо возвращаемых кодов, относящихся к репликации, команда DSRUTIL STATUS имеет код завершения, значение которого можно получить после выполнения команды с помощью специальной переменной $?. Существует три кода завершения, означающие следующее:

- 0 – команда завершила работу корректно, результат ее работы отправлен в stdout[1];
- 2 – произошла ошибка при подключении к базе данных;
- 3 – подключение к базе данных было выполнено, но в этой базе не включен механизм OE Replication.

Следующий пример демонстрирует использование кода завершения ($?) команды DSRUTIL STATUS в скрипте repluptime.sh, который по результатам работы команды DSRUTIL MONITOR определяет отставание target-базы от source-базы данных в секундах и в блоках:

```
$DLC/bin/dsrutil $1 -C status &>/dev/null
case $? in
   0)
     echo "a" > $HOME/repl_zadan
     echo "" >> $HOME/repl_zadan
     echo "q" >> $HOME/repl_zadan
     echo "q" >> $HOME/repl_zadan
     cat $HOME/repl_zadan | $DLC/bin/dsrutil $1 -C monitor 2>/dev/null  1>$HOME/list2
     TEXT=`cat $HOME/list2 | grep second | grep -v seconds`
     SB=`cat $HOME/list2 | grep "Source RDBMS" |  awk '{print $9}'`
     TB=`cat $HOME/list2 | grep "Last Processed" |  awk '{print $9}'`
     DELTA=`expr $SB - $TB`
     echo $TEXT "," $DELTA "blok(s)"
     if [ -f $HOME/list2 ]
     then
        rm -f $HOME/list2
     fi
     if [ -f $HOME/repl_zadan ]
     then
        rm -f $HOME/repl_zadan
     fi
   ;;
   2) echo "База данных сейчас НЕДОСТУПНА!" ;;
   3) echo "В базе данных не включен OE Replication" ;;
esac
```

Результат работы этого скрипта выглядит следующим образом:

```
Repl Server behind Source DB by: 0 second(s) , 0 blok(s)
```

В данном случае это сообщение говорит об отсутствии отставания target-базы данных от source-базы как по времени, так и по количеству блоков.

[1] stdout - стандартный поток вывод данных в системах Unix, по умолчанию нацелен на запись на устройство отображения (монитор)