



# Службы баз данных



---

# Содержание

## Глава 1. Управление базами данных . . . 1

Конфигурация и репликация баз данных . . . . .	1
Базы данных WebSphere Business Monitor . . . . .	2
Создание и развертывание баз данных . . . . .	4
Репликация баз данных . . . . .	11
Управление изменениями и создание артефактов . . . . .	31
Обслуживание баз данных. . . . .	33
Создание и настройка баз данных . . . . .	37
Управление базами данных во время выполнения . . . . .	38
Схема базы данных хронологии . . . . .	54

## Глава 2. Службы баз данных . . . . . 57

Схема базы данных хронологии . . . . .	59
Управляющая таблица службы перемещения данных . . . . .	60
Таблица метаданных и протокола службы перемещения данных . . . . .	63

## Глава 3. Устранение неполадок служб базы данных . . . . . 65

Вопросы развертывания . . . . .	65
Вопросы выполнения . . . . .	67
Остановка базы данных . . . . .	67



---

## Глава 1. Управление базами данных

Управление базами данных WebSphere Business Monitor имеет особое значение на этапах установки и развертывания WebSphere Business Monitor.

Управление базами данных включает следующие задачи:

- Создание баз данных
- Установка и конфигурация баз данных
- Создание статических и динамических таблиц и индексов базы данных
- Развертывание созданных сценариев репликации
- Развертывание определений метаданных cube view
- Обслуживание баз данных

**Примечание:** В документации WebSphere Business Monitor обсуждаются основные схемы различных баз данных. Некоторые элементы вывода генератора схемы знакомят с природой схем базы данных WebSphere Business Monitor. В то время как DBA, вероятно, использует эту информацию для обслуживания и тонкой настройки хранилища данных, ее не следует рассматривать как определение внешнего API. Возможно, что в следующих версиях WebSphere Business Monitor эта основная схема будет изменена. Заказчикам следует разработать инфраструктуру, в которой предполагается, что в следующих выпусках будет поддерживаться совместимость с предыдущими версиями этих схем. Пользовательский код, написанный для использования баз данных WebSphere Business Monitor, может быть несовместим со следующими выпусками продукта.

Ниже приведена информация, полезная при планировании и подготовке к управлению базами данных WebSphere Business Monitor.

---

## Конфигурация и репликация баз данных

Управление данными играет основную роль в WebSphere Business Monitor.

Архитектура базы данных WebSphere Business Monitor поддерживает следующие требования:

- Динамическая обработка хранилища данных должна быть отделена от доступа клиентов к хранилищу данных, для того чтобы обеспечить подходящий уровень обработки
- Должна существовать возможность выполнять клиентские запросы на изменение данных и достаточно быстро на них реагировать
- Должен быть оптимизирован доступ к хранилищу данных хронологии для многомерного анализа и составления отчетов

Форматы данных в базах данных WebSphere Business Monitor отличаются в зависимости от использующего их компонента. Данные используются двумя основными компонентами: процессором событий и клиентскими сводными панелями. Это различие в использовании делает необходимым отделение базы данных обработки событий от базы данных сводных панелей. Далее данные можно подразделить на информацию, связанную с моделью бизнес-величин, и информацию об обработке событий.

Сводные панели отображают два типа данных: новые и хронологические экземпляры данных. Число новых экземпляров мало по сравнению с числом хронологических экземпляров. Запросы к новым экземплярам должны выполняться очень быстро и на них не должны оказывать воздействие хронологические экземпляры. Эти два типа данных были разделены в две базы данных: рабочую и хронологию. Для увеличения производительности архитектура поддерживает все следующие функции:

- База данных, служащая контейнером определений для моделей бизнес-величин. Также хранится информация о других базах данных.

- База данных для поддержки транзакций, используемая процессором событий.
- База данных, работающая почти в реальном времени и поддерживающая очереди анализа без воздействия на сервер транзакций. Она используется сводными панелями.
- База данных, поддерживающая многомерный анализ хронологии транзакций. Она используется сводными панелями для просмотра данных хронологии.

Базы данных WebSphere Business Monitor разделены на четыре базы:

- **Хранилище:** содержит модели бизнес-величин и определения событий. В этой базе данных хранятся также схемы, имена и имена хостов базы данных состояний, рабочей базы данных и базы данных хронологии.
- **Состояния:** содержит текущее состояние выполняющихся экземпляров процесса и значения бизнес-величин, связанных с каждым экземпляром процесса. Она используется для обработки событий сервером WebSphere Business Monitor.
- **Рабочая:** в базе данных состояний и рабочей базе данных хранится в значительной степени одинаковая информация. Рабочая база данных отличается только способом хранения некоторых данных, насколько данные современны и как долго хранятся. Данные в рабочих базах данных хранятся по крайней мере на 24 дольше, чем в базе данных состояний. Назначение рабочей базы данных состоит в том, чтобы разрешить пользователю проводить анализ почти в реальном времени, не влияя на обработку событий сервером WebSphere Business Monitor. Рабочая база данных обслуживает клиентские очереди недавно использовавшихся экземпляров. В ней хранится информация о выполнении для группы бизнес-величин для целей составления отчетов. Она используется для просмотра информации в сводных панелях.
- **Хронология:** содержит информацию о завершенных экземплярах и текущее состояние выполняющихся экземпляров в виде звезды для целей составления многомерных хронологических отчетов. Она используется для просмотра информации в сводных панелях.

Имеется две базы данных, используемых для хранения отслеживаемых событий и данных диспетчера адаптивных действий. Эти базы данных предназначены для внутреннего использования WebSphere Business Monitor. В них не хранится никакая информация, относящаяся к экземплярам процессов или к показателям.

- **Генератор событий:** Хранит события, порожденные службами. Таблицы баз данных генератора событий расположены в базах данных служб.
- **Каталог действий:** Хранит события, определенные как ситуации и действия, которые диспетчер адаптивных действий должен выполнить над ними. Эта база данных создается во время установки.

## Базы данных WebSphere Business Monitor

WebSphere Business Monitor использует четыре базы данных для хранения данных событий и метаданных модели бизнес-величин. Используемые базы данных: рабочая база данных, базы данных состояний, хронологии и хранилища.

### База данных хранилища

База данных хранилища содержит метаданные, описывающие развернутые в данный момент модели бизнес-величин, а также информацию о других базах данных WebSphere Business Monitor. В базе данных хранилища содержится хронология развернутых моделей. Для одной установки WebSphere Business Monitor имеется только одна база данных хранилища.

Панель запуска заполняет базу данных хранилища следующими атрибутами базы данных состояний, рабочей базы данных и базы данных хронологии: имя базы данных, схема базы данных и имена хостов сервера базы данных. Эти атрибуты используются другими компонентами WebSphere Business Monitor для доступа к этим базам данных во время выполнения. База данных хранилища заполняется при импорте модели бизнес-величин.

База данных хранилища используется следующими компонентами:

- **Административная консоль**  
Модель бизнес-величин импортируется с помощью административной консоли WebSphere Business Monitor. В базу данных хранилища заносятся определения процессов и событий этой импортированной

модели. Когда импорт завершен, модель бизнес-величин считается развернутой. После импорта модели определения процессов и событий будут доступны другим компонентам WebSphere Business Monitor для извлечения.

База данных хранилища также используется генератором схемы. Генератору схемы требуется имя схемы, которое следует использовать при генерации артефактов баз данных. Когда пользователь изменяет предварительно развернутую модель бизнес-величин и пытается повторно создать схему для нее, генератор схемы проверяет наличие артефактов в базе данных хранилища перед генерацией артефактов управления изменениями.

- **Сводные панели**

Сводные панели имеют набор представлений, которые отображают данные в различных проекциях. Некоторые из этих представлений заполняются данными из рабочей базы данных, а некоторые - из базы данных хронологии. Для того чтобы разрешить пользователю настраивать и задавать параметры для этих представлений, представлениям необходимо получать метаданные WebSphere Business Monitor из базы данных хранилища. Некоторые представления требуют также создания запросов к DB2 Alphablox Cubes. Для составления таких запросов необходимы метаданные о размерностях, величинах и именах куба, которые получают из базы данных хранилища. Также все сводные панели отображают диаграммы бизнес-процессов, которые хранятся в базе данных хранилища.

- **WebSphere Business Monitor сервер**

Сервер WebSphere Business Monitor использует базу данных хранилища для извлечения определения процессов и событий.

## **База данных состояний**

В базе данных состояний хранится информация о выполняющихся экземплярах. Эта информация включает в себя значения показателей, бизнес-величин и ключевых индикаторов производительности (KPI). Она оптимизирована для интенсивной обработки транзакций. Для одной установки WebSphere Business Monitor имеется только одна база данных состояний.

Для каждого экземпляра процесса требуется две таблицы в базе данных состояний для хранения показателей, бизнес-величин и KPI. Эти таблицы имеют такую же структуру как и у экземпляров процесса. Каждая бизнес-величина представлена отдельным полем в одной из двух таблиц. В зависимости от выбранных во время компоновки моделей бизнес-величин опций, большая часть или вся информация из базы данных состояний реплицируется в рабочую базу данных.

База данных состояний используется сервером WebSphere Business Monitor. Во время выполнения сервер WebSphere Business Monitor добавляет, извлекает и изменяет информацию об экземплярах процессов, расположенных в базе данных состояний, в соответствии с обработанными событиями.

База данных состояний содержит следующую информацию:

- Информация о группе бизнес-величин, являющейся частью данных в импортированных моделях бизнес-величин.
- Выполняющийся экземпляр процесса, созданный во время выполнения WebSphere Business Monitor.
- Записи событий выполняющихся процессов. Запись события - это данные о событии, полученном для обновления определенной группы бизнес-величин.

## **Рабочая база данных**

Структура рабочей базы данных аналогична структуре базы данных состояний. Она получает информацию из базы данных состояний о текущем состоянии всех выполняющихся процессов, а также окончательное состояние завершенных или неудавшийся процессов. Эта информация используется сводными панелями WebSphere Business Monitor. Рабочая база данных также используется Диспетчер адаптивных действий для хранения предупреждений. Для одной установки WebSphere Business Monitor имеется только одна рабочая база данных.

В рабочей базе данных хранятся:

- Предупреждения, отправленные Диспетчер адаптивных действий сводным панелям

- Данные процессов
- Значения показателей

Информация реплицируется в рабочую базу данных из базы данных состояний.

Рабочая база данных используется сводными панелями WebSphere Business Monitor. Сводные панели извлекают данные выполняющихся или недавно завершенных экземпляров, необходимые для заполнения представлений. Представления сводной панели используют рабочую базу данных для анализа, поэтому они оптимизированы для обработки простых и агрегированных запросов.

## Базы данных хронологии

В базе данных хронологии хранятся все завершенные и выполняющиеся экземпляры процессов. Она используется сводными панелями для улучшения анализа данных с помощью DB2 Alphablox. Для одной установки WebSphere Business Monitor имеется только одна база данных хронологии. Данные в базе данных хронологии никогда не удаляются.

База данных хронологии должна содержать только данные за два года. Это одно из требований продукта WebSphere Business Monitor. Как было упомянуто ранее, данные хронологии никогда не удаляются автоматически. Таким образом, DBA отвечает за удаление данных, хранящихся дольше двух лет. В базе данных хронологии хранится информация, относящаяся как к экземплярам, которые выполняются длительное время, так и к завершенным экземплярам. Эта информация хранится в форме звезды, а не в плоских формах транзакций, применяемых в базе данных состояний и рабочей базе данных. База данных хронологии оптимизирована для агрегированных и длительных запросов. Она используется DB2 Alphablox в сводных панелях для предоставления расширенных многомерных отчетов.

Информация реплицируется в базу данных хронологии из рабочей базы данных.

В базе данных хронологии каждый экземпляр процесса имеет свой собственный набор таблиц. В отличие от базы данных состояний и рабочей базы данных каждый набор таблиц является схемой звезды, которая поддерживает составление многомерных отчетов.

База данных хронологии содержит динамические таблицы, созданные в соответствии с развернутой моделью бизнес-величин. Генератор схемы создает схему базы данных хронологии, которая используется для создания динамических таблиц и определений Cube Views .

База данных хронологии используется сводными панелями WebSphere Business Monitor. Сводные панели извлекают данные, необходимые для заполнения некоторых представлений базы данных хронологии. Например, представление Отчеты сфокусировано на анализе данных, извлеченных из базы данных хронологии.

База данных хронологии содержит следующую информацию:

- Данные различных версий выполняющихся или завершенных экземпляров процессов.
- Данные завершенных экземпляров процессов, хранящиеся в рабочей базе данных. Все завершенные экземпляры хранятся в рабочей базе данных в течение 24 часов. 24 часа - это стратегия хранения по умолчанию. Ее можно изменить как часть конфигурации службы перемещения данных. После репликации этих данных в базу данных хронологии, они удаляются из рабочей базы данных для повышения производительности.

## Создание и развертывание баз данных

Базы данных WebSphere Business Monitor создаются панелью запуска во время установки. Перед созданием баз данных следует спланировать их создание и развертывание.

Если базы данных WebSphere Business Monitor были удалены или повреждены после установки, администратор базы данных (DBA) может повторно создать их вручную выполнив сценарии создания, расположенные в <каталог\_установки\_монитора>\install\mondb\ . DBA также может удалить базы данных из



системы с помощью панели запуска. Для создания баз данных DBA следует сначала вручную удалить базы данных из DB2, а затем повторно создать эти базы данных с помощью панели запуска.

## Подготовка к развертыванию артефактов баз данных

Перед началом создания баз данных WebSphere Business Monitor с помощью панели запуска необходимо составить их план. Планирование включает в себя выделение памяти для баз данных, подготовка стратегий резервного копирования и указание параметров размеров таблиц и буферных пулов, а также определение параметров для экземпляров баз данных и отдельных баз данных.

Во время установки панель запуска создает рабочую базу данных и базы данных состояний и хронологии, а также объекты баз данных, которые будут применяться для целей администрирования. В дополнение к этим объектам генератор схемы создает набор объектов баз данных (например, таблицы), предназначенных для модели бизнес-величин. Панель запуска создает набор табличных пространств и буферных пулов по умолчанию для рабочей базы данных и баз данных состояний и хронологии. Ссылки на эти табличные пространства по умолчанию находятся в файле конфигурации табличного пространства. Эти табличные пространства разработаны для того, чтобы позволить пользователям быстро подготовиться к тестированию и проверки концептуальных сценариев. Для того чтобы избежать снижения быстродействия и ограничений на ресурсы, важно спланировать распределение таблиц по табличным пространствам, а также применение табличными пространствами контейнеров и буферных пулов.

Во время установки создаются базы данных, а также только определения статических таблиц. В процессе генерации артефактов динамические таблицы назначаются табличным пространствам в рабочей базе данных и в базах данных состояний и хронологии на основании настраиваемого текстового файла конфигурации. WebSphere Business Monitor поставляется с файлом конфигурации по умолчанию, который расположен в каталоге <каталог\_установки\_монитора>\install\mondb. Этот файл конфигурации по умолчанию отображает каждую таблицу на одно табличное пространство подходящего размера. Для поддержки частного развертывания панель запуска создает во время установки набор табличных пространств (с размером страниц 4Кб, 8Кб, 16Кб и 32Кб), который соответствует записям в файле конфигурации по умолчанию. Приведенный ниже пример содержит выдержку из файла конфигурации табличного пространства по умолчанию, поставляемого вместе с WebSphere Business Monitor:

```
#
# База данных состояний
#
db2.state.Default.TABLE.4K.0=DSDFLTTS4
db2.state.Default.TABLE.8K.0=DSDFLTTS8
db2.state.Default.TABLE.16K.0=DSDFLTTS16
db2.state.Default.TABLE.32K.0=DSDFLTTS32
```

Эта простая конфигурация применяется во время генерации артефактов; все таблицы, которые следует создать в базе данных состояний и соответствующие табличному пространству с размером страницы 4Кб, назначаются DSDFLTTS4. Таблицы с размером страниц 8 Кб будут храниться в табличном пространстве с именем DSDFLTTS8, таблицы с размером страниц 16 Кб - в DSDFLTTS16. Поскольку требования к табличному пространству могут меняться (в зависимости от сложности модели и объема хранящихся данных), рекомендуется не использовать параметры конфигурации табличного пространства по умолчанию в тестовой или рабочей среде. Планирование и определение подходящей стратегии хранения позволяет обеспечить высокое быстродействие.

Расширенные параметры файла конфигурации можно использовать для отображения таблиц в табличные пространства не только на основании размера страниц, но также на основании типа данных, которые будут сохранены. За дополнительной информацией обратитесь к примеру файла конфигурации.

Для определения того, какому табличному пространству присвоить таблицу, генератор схемы выполняет следующие задачи:

- Определяет тип таблицы.
- Вычисляет минимальный размер страницы необходимый для хранения одной записи данных.
- Устанавливает доступные для этого типа табличные пространства. Если такое табличное пространство найдено, он определяет его расположение и использует. В ином случае продолжает работу.
- Устанавливает доступные табличные пространства, заданные по умолчанию для генератора схемы. Если такое табличное пространство находится, он определяет его расположение и использует. Если доступное табличное пространство данного типа не найдено, продолжайте выполнение следующих действий.
- Назначьте таблицу табличному пространству базы данных по умолчанию (не указывая блок табличного пространства во время создания таблицы).

**Примечание:** Развертывание не будет удачным, если для требуемого размера страницы в базах данных не задано табличное пространство по умолчанию.

Можно отредактировать файл конфигурации табличного пространства с помощью любого текстового редактора либо создайте новый файл. Используйте вкладку общей конфигурации в административной консоли генератора схемы для того, чтобы заставить генератор схемы использовать другой файл конфигурации.

**Примечание:** Генератор схемы в действительности не создает никакие табличные пространства для записей в этом файле конфигурации. Это необходимо выполнить вручную до развертывания созданных артефактов баз данных. Развертывание артефактов будет неудачным, если таблица назначается несуществующему табличному пространству.

## Развертывание артефактов базы данных

Таблицы базы данных развертываются после создания баз данных WebSphere Business Monitor с помощью панели запуска. На этапе развертывания генератор схемы настраивается для создания артефактов, которые будут развернуты и завершат установку базы данных. После этого базы данных можно заполнять данными.

**Примечание:** Базы данных создаются один раз. Для каждой модели бизнес-величин в базу данных добавляются таблицы.

### Генерация схем баз данных:

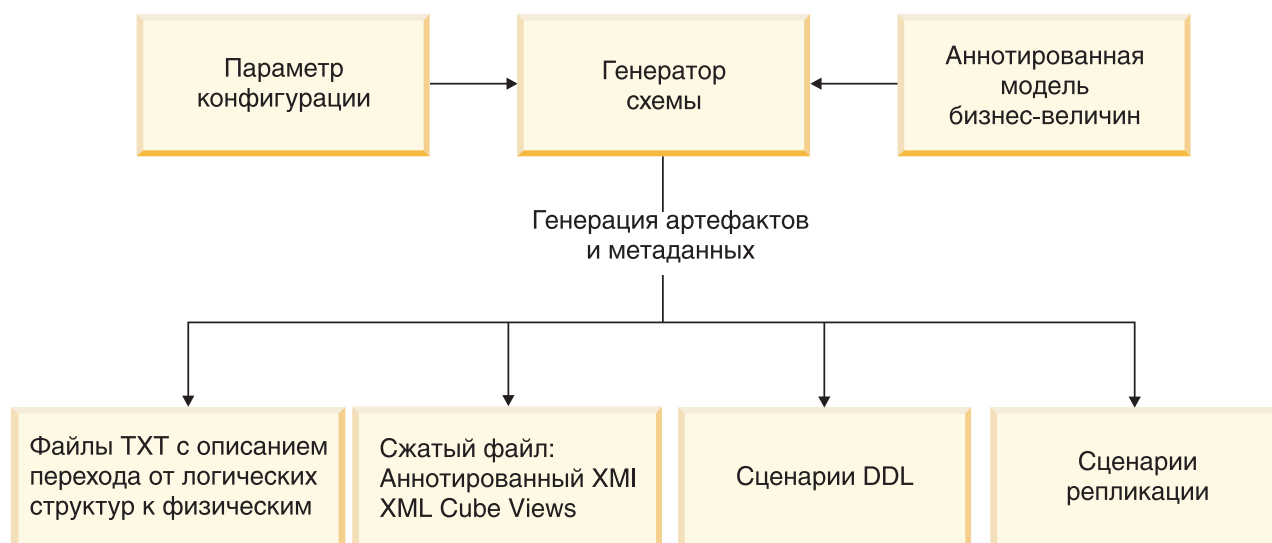
Схемы баз данных WebSphere Business Monitor основаны на моделях бизнес-величин. Эти схемы создаются генератором схемы.

Модель бизнес-величин создается с помощью редактора бизнес-величин. Это модель с комментариями, содержащая метаданных бизнес-модели. С помощью редактора бизнес-величин пользователь может определить, что необходимо отслеживать: контекст, ключевые индикаторы производительности (KPI), показатели и бизнес-ситуации. Модель бизнес-величин создает информацию об отслеживании событий. Когда модель бизнес-величин завершена, она экспортируется в WebSphere Business Monitor как файл XML с комментариями в формате zip для использования генератором схемы.

Генератор схемы является частью административной консоли WebSphere Business Monitor. Обычно это средство настраивается и применяется администратором базы данных (DBA). Генератор схемы использует модель бизнес-величин в качестве входных данных, а затем генерирует артефакты базы данных.

Ниже приведены создаваемые артефакты:

- Сценарии на языке определения данных (DDL), предназначенные для создания таблиц баз данных. Для каждого созданного файла DDL (state.ddl, runtime.ddl, datamart.ddl) создается соответствующий текстовый файл (stateMapping.txt, runtimeMapping.txt, datamartMapping.txt). Эти файлы содержат описание того, какие показатели и процессы представлены физическими артефактами базы данных (таблицами, полями).
- Определения DB2 Cube Views, описывают данные хронологии в формате звезды. Определения cube views можно импортировать в центр DB2 OLAP.
- Сценарии репликации, которые включают репликацию между базой данных состояний, рабочей базой данных и базой данных хронологии. Для каждой из этих баз данных генератор схемы создает сжатый файл, содержащий все необходимые артефакты репликации. Обычно эти артефакты распространяются и развертываются DBA в соответствии с инструкциями в “Развертывание служб перемещения данных” на стр. 40.



Таблицы компонентов базы данных WebSphere Business Monitor бывают двух типов:

- Таблицы базы данных состояний, которые создаются во время установки. Эти таблицы являются общими для всех моделей бизнес-величин и не зависят от отдельных моделей бизнес-величин.
- Существуют динамические таблицы баз данных, которые зависят от модели бизнес-величин, импортируемой в административную консоль WebSphere Business Monitor. Схемы динамических таблиц базы данных являются уникальными для каждой модели бизнес-величин. Все изменения в модели бизнес-величин, связанные с динамическими таблицами, приводят к изменению сценария управления изменениями. Более подробная информация о сценариях управления изменениями приведена в “Управление изменениями и создание артефактов” на стр. 31.

### Определения Cube views:

Генератор схемы создает XML-файл DB2 Cube Views. Администратор базы данных (DBA) импортирует этот файл XML в DB2 OLAP Center.

Генератор схемы создает XML-файл Cube Views на основании модели бизнес-величин. Модель бизнес-величин содержит информацию, полезную при описании величин и размерностей. Можно также описать, какие агрегирования можно применить к величинам.

Для каждого процесса в модели бизнес-величин создаются куб и модель куба. Для действий, связанных с процессом, также создаются куб и модель куба. Каждый куб и модель куба содержат некоторые предопределенные величины и размерности, создаваемые автоматически.

Каждый куб и модель куба содержат три встроенных величины:

- **Прошедшее время:** для этой величины определена функция агрегирования **avg**.
- **Время работы:** для этой величины определена функция агрегирования **avg**.
- **Количество экземпляров:** для этой величины определена функция агрегирования **count**.

Автоматически создаются следующие размерности:

- **CreationTime:** Момент создания экземпляра процесса
- **StartTime:** Момент запуска экземпляра процесса
- **State:** Содержит возможные состояния (строковые значения) экземпляра процесса, такие как запущен, выполняется или завершен
- **TerminationTime:** Момент завершения экземпляра процесса

Для этих основанных на времени размерностей применяется определенная общим образом размерность (DIM\_TIME). Она имеет три предопределенных уровня: год, месяц и день.

При моделировании можно определить свои собственные бизнес-величины. Созданные бизнес-величины могут быть или величинами, или размерностями. В документации WebSphere Business Modeler подробно описывается применение WebSphere Business Modeler для создания величин и размерностей.

Более подробная информация о DB2 Cube Views содержится в документации DB2.

### **Создание и развертывание артефактов:**

Генератор схемы создает артефакты представлений базы данных и куба на основании каждой импортированной модели бизнес-величин.

Артефакты должны выполняться администратором базы данных на этапе развертывания моделей бизнес-величин. Перед развертыванием любой модели бизнес-величин необходимо выполнить следующие задачи:

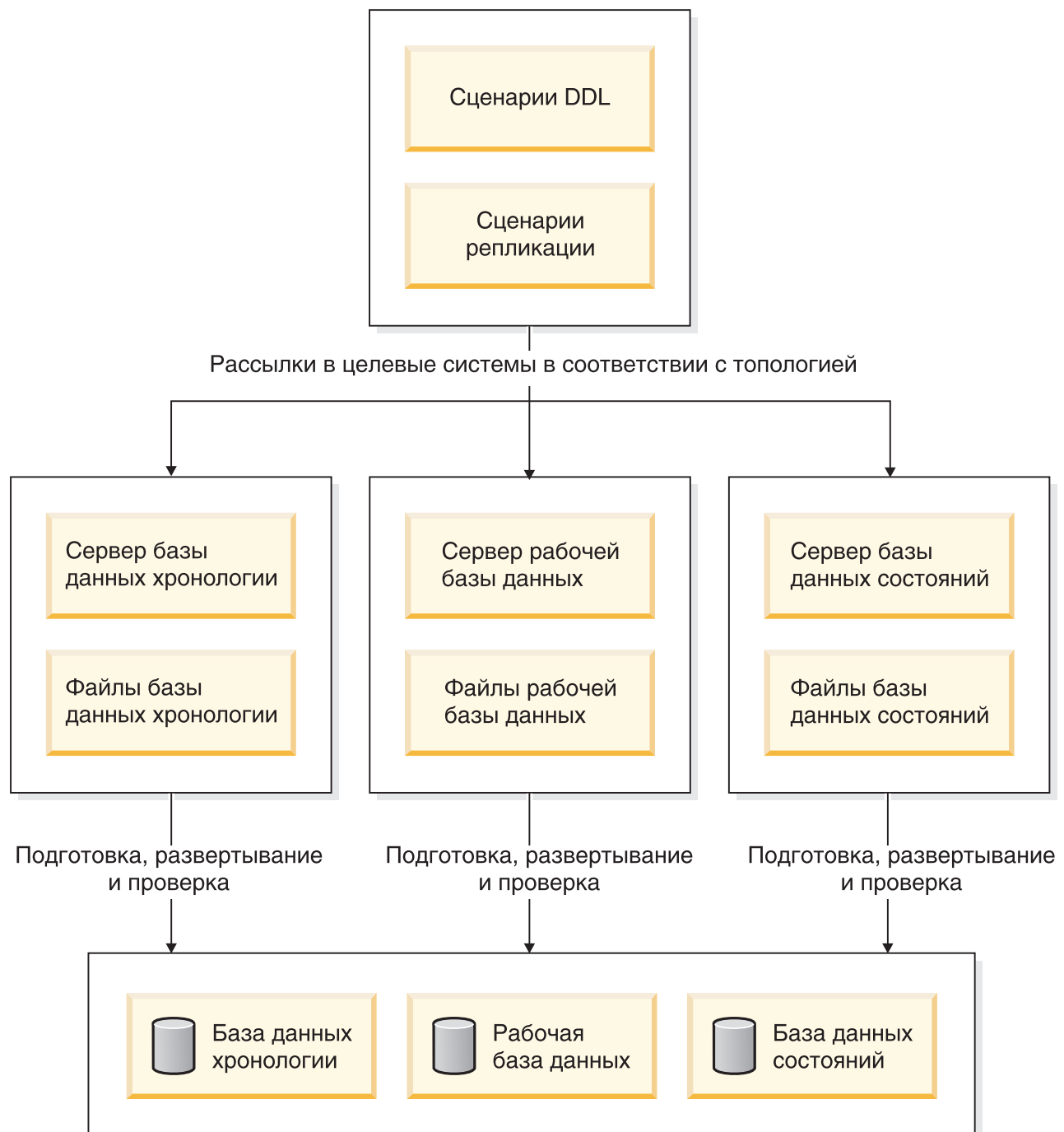
- Выполнить задачи предварительного планирования. В их число входит принятие решения о топологии, определение частоты событий, поддерживаемого числа пользователей, параметров базы данных и экземпляра базы данных, буферных пулов и табличных пространств, а также стратегии резервного копирования и восстановления. Также важно разработать стратегию определения способа и места хранения созданных артефактов. Также рекомендуется иметь возможность поиска по набору созданных ранее артефактов в соответствии с моделью и номером версии, а также были ли они развернуты ранее. Это может оказаться полезным для управления изменениями и поддержки сценариев.
- Создать рабочую базу данных, базы данных состояний, хранилища и хронологии с помощью панели запуска WebSphere Business Monitor. Создание базы данных включает создание набора статических таблиц базы данных, настройка базы данных состояний и рабочей базы данных в качестве источников для репликации, а также создание других объектов базы данных, например, хранящихся процедур и udf, относящихся к различным компонентам WebSphere Business Monitor.
- С помощью генератора схемы создать артефакты, относящиеся к модели бизнес-величин.

**Примечание:** Перед развертыванием созданных артефактов рекомендуется создать резервные копии всех баз данных.

Для развертывания созданных артефактов выполните следующие действия:

- Разверните сценарии DDL для создания динамических таблиц модели бизнес-величин в рабочей базе данных и базах данных состояний и хронологии.
- Разверните сценарии репликации для включения репликации баз данных.
- Импортируйте файл определения представления куба в DB2 Cube Views.

Следующий рисунок описывает этап развертывания артефактов:



### Настройка артефактов:

В определенных обстоятельствах для улучшения производительности может оказаться полезным изменение созданных артефактов базы данных. Существует два основных подхода к решению данной задачи: итерационное усовершенствование и частное усовершенствование.

### Итерационное усовершенствование преобразования табличного пространства

Число таблиц, создаваемых генератором схемы, зависит (среди всего прочего) от сложности модели бизнес-величин. Изначально сложно определить, каким образом можно оптимально присвоить эти таблицы пространствам таблиц. Следующий простой подход может оказаться полезным при постепенном

усовершенствовании преобразования таблицы в пространство таблиц. Дополнительная информация о планировании базы данных приведена в разделе “Подготовка к развертыванию артефактов баз данных” на стр. 5. Для усовершенствования преобразования таблицы в пространство таблиц следует выполнить следующие действия:

1. Запустите генератор схем с помощью файла конфигурации пространства таблицы (по умолчанию или какого-либо другого специального файла).
2. Извлеките созданные артефакты во временный каталог и определите номер и тип таблиц, которые будут созданы в базе данных. Для каждой модели бизнес-величин будут созданы таблицы следующих типов:
  - **Контекст и действие:** существует только в базах данных состояний и рабочих базах данных, и доступ к ним осуществляется с помощью сервера и сводных панелей.
  - **CD:** существует только в базах данных состояний и рабочих базах данных. Таблицы CD обычно содержат больше столбцов, чем связанные с ними таблицы контекста и действий. Размер конкретной таблицы CD в значительной степени зависит от числа транзакций относительно соответствующего контекста или таблицы действий, от отношения числа транзакций обновления и вставки, от того, с какой частотой связанный компонент применения изменений службы перемещения данных считывает транзакции из таблицы CD и включает их в соответствующую таблицу CCD, а также от того, как часто записи таблицы CD удаляются связанным компонентом жизненного цикла источника.
  - **CCD:** существует только в рабочих базах данных и базах данных хронологии. Структура таблицы CCD полностью совпадает со структурой соответствующей таблицы CD, и ее размер в значительной степени определяется теми же факторами. Единственное отличие состоит в том, что вместо считывания компонентом применения изменений транзакции считываются компонентом ETL, а записи удаляются компонентом жизненного цикла приемника.
  - **Внутренние таблицы RM:** существуют только в рабочих базах данных и базах данных хронологии. Эти таблицы используют страницы с максимальным размером 4 Кб.
  - **Факт и размерность:** существуют только в базах данных хронологии.
3. Изменяют пространство таблиц таким образом, что оно содержит:
  - Запись преобразования для каждого типа таблицы
  - Записи преобразования таблиц нескольких типов в том случае, если большое число таблиц может быть присвоено одному и тому же табличному пространству

#### **Примечание:**

- Не задавайте описания табличного пространства для типов таблиц, которые не будут созданы в определенной базе данных, потому что эти табличные пространства не будут использоваться.
  - Показатели соответствуют полям таблицы. Чем больше показателей определено, тем больше будет таблица, и, таким образом, большее табличное пространство будет требоваться.
4. Создайте эти табличные пространства (и буферные пулы) перед развертыванием созданных артефактов. Генератор схемы не проверяет, существуют ли табличные пространства, объявленные в файле конфигурации, поскольку соединение с базой данных не установлено. Однако, развертывание не будет удачным, если табличные пространства не существуют.
  5. Перезапустите генератор схемы с помощью измененного файла конфигурации табличного пространства.

**Примечание:** Измените имя файла конфигурации в конфигурации административной консоли генератора схемы при создании нового файла конфигурации табличного пространства.

#### **Частные усовершенствования**

Можно изменять созданные артефакты, изменив следующее:

- Любой индекс (добавление, изменение, удаление, назначение отдельному табличному пространству (только DMS)) для улучшения производительности базы данных.

**Примечание:** Следует внимательно относиться к созданию уникальных индексов, поскольку это может привести к неожиданным сбоям.



- Любое присваивание табличного пространства (можно назначить другое табличное пространство, добавить присваивание табличного пространства для индексов или больших объектов, если применяются табличные пространства DMS (управляемые базой данных пространства)).
- Любые комментарии к таблицам (не рекомендуется, поскольку комментарии определяют, что представляет каждая таблица и поле)

Вообще, разрешенные изменения не затрагивают основную схему или структуру таблиц.

При изменении созданных сценариев следует иметь в виду следующие ограничения:

- Нельзя изменять имена таблиц.
- Нельзя изменять имена полей.
- Нельзя изменять типы данных полей.
- Нельзя удалять поля или таблицы.
- Нельзя добавлять поля или таблицы.
- Нельзя изменять первичные ключи таблиц.
- Нельзя изменять свойство способности полей принимать неопределенное значение (nullability).
- Нельзя изменять присваивание схемы для таблиц.
- Нельзя добавлять новые ограничения, такие как ограничение уникальности или ограничение по внешнему ключу.

**Важное замечание:** Изменения в созданных артефактах базы данных не учитываются при следующем запуске генератора схемы для той же модели бизнес-величин. Например, пользователь изменяет созданные артефакты для модели бизнес-величин "Финансовая модель" перед их развертыванием. Впоследствии пользователь изменяет модель бизнес-величин и повторно создает все артефакты. В этом случае пользователю потребуется изменить заново созданные артефакты, поскольку генератору схемы не известно об изменениях в ранее созданных артефактах.

#### Примечание:

- Для развертывания артефактов требуется Java версии 1.4.2 или более поздних версий.
- Перед выполнением сценариев репликации в путь к системе необходимо добавить каталог Java bin.

## Репликация баз данных

Технология репликации баз данных применяется для перемещения относящихся к модели бизнес-величин данных из базы данных состояний в рабочую базу данных и из рабочей базы данных в базу данных хронологии.

Для настройки репликации для модели бизнес-величин между базой данных состояний, рабочей базой данных и базой данных хронологии необходимо выполнить следующие задачи высокого уровня:

1. **Генерация сценариев настройки репликации.** Генератор схемы анализирует модель бизнес-величин, для которой следует настроить репликацию, и создает набор файлов настройки.
2. **Распространение сценариев настройки репликации.** Эти файлы настройки необходимо вручную переместить в системы, в которых размещаются рабочая база данных и базы данных состояний и хронологии.
3. **Выполнение сценариев настройки репликации.** Каждый из файлов настройки при выполнении создаст необходимые объекты базы данных и настроит утилиты, перемещающие данные из одной базы данных в другую.

После успешного завершения можно запускать и выполнять утилиты репликации. В следующих разделах приводится обзор архитектуры и даются объяснения некоторых применяемых фундаментальных понятий.

## Службы перемещения данных

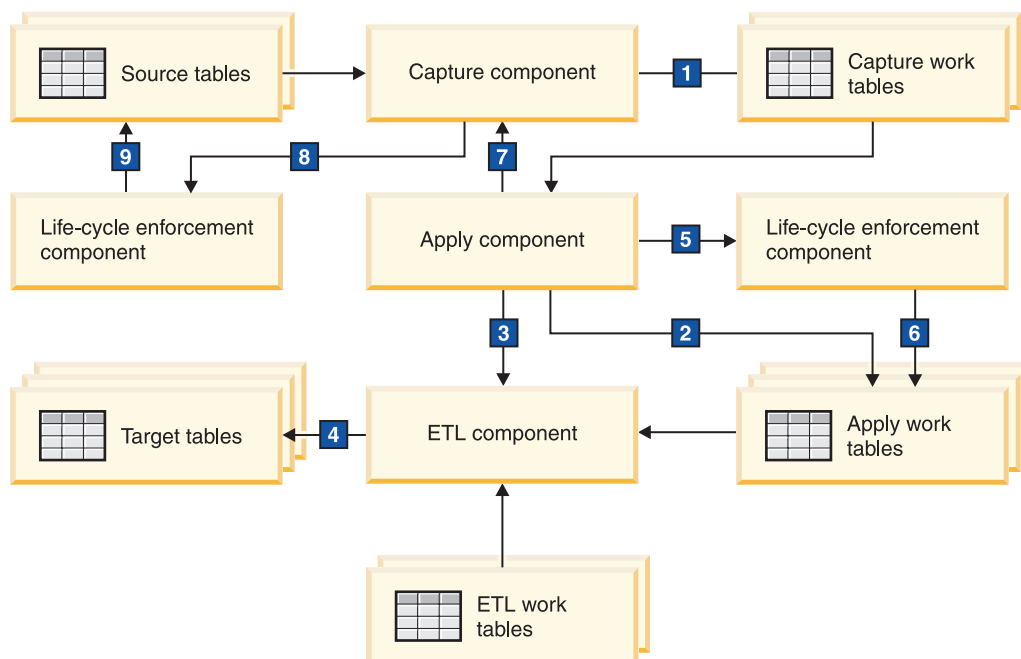
Служба перемещения данных осуществляет перенос данных из исходной базы данных в целевую. Исходная и целевая базы данных могут быть однородными или неоднородными, и могут располагаться в одной системе

или быть распределенными между несколькими системами. Кроме перемещения данных служба также может осуществлять их преобразование и предоставлять основные функциональные возможности жизненного цикла данных по требованию приложения.

Службы перемещения данных реализованы в виде пяти главных компонентов:

1. Компонент сбора данных (источника)
2. Компонент применения изменений (приемника)
3. Компонент ETL (извлечь, преобразовать, загрузить)
4. Компонент жизненного цикла источника
5. Компонент жизненного цикла приемника

Компоненты сбора данных и применения изменений работают вместе, перемещая данные из исходной базы данных в целевую. Компонент ETL выполняет необходимое преобразование данных, если структуры данных исходной базы отличаются от структур данных целевой. На следующей диаграмме представлен процесс перемещения данных:



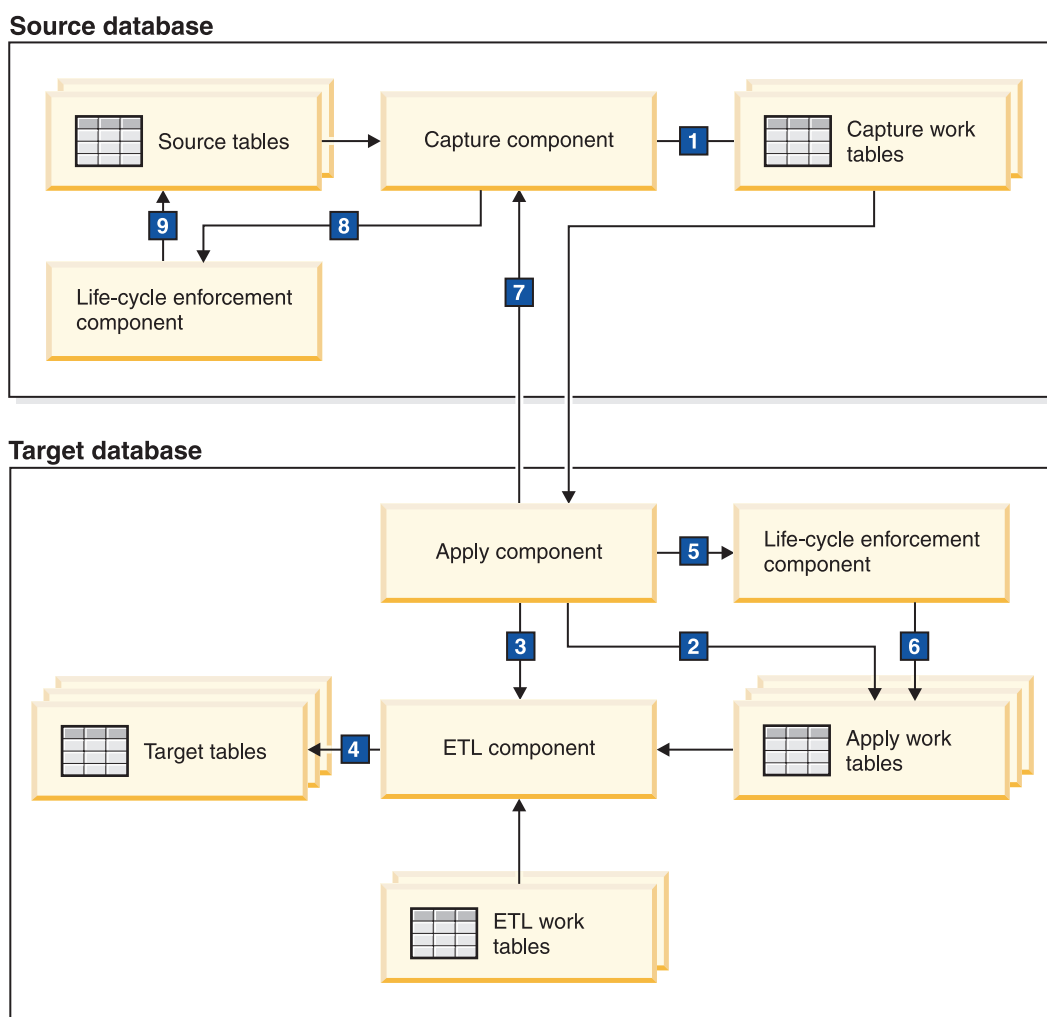
Процесс перемещения данных включает следующие шаги:

1. Данные сохраняются в исходных таблицах и часто обновляются, например, сервером монитора. Компонент сбора данных записывает в рабочие таблицы все изменения данных, внесенные в исходные таблицы.
2. Через predetermined intervals of time these changes are recognized by the change application component and recorded in the work tables.
3. После успешной записи изменений вызывается компонент ETL.
4. Этот компонент выполняет необходимые преобразования, используя сохраненные в рабочих таблицах компонента применения изменений данные и predetermined rules. Successfully transformed data is recorded in the target tables. Any incomplete or incorrect data remains in the work tables for further processing.
5. После завершения обработки данных компонентом ETL вызывается компонент жизненного цикла приемника.



6. Со временем в рабочих таблицах компонента применения изменений может накопиться большой объем данных. Все успешно обработанные компонентом ETL данные удаляются из этих таблиц компонентом жизненного цикла приемника.
7. После того как данные успешно скопированы в целевую базу данных, они больше не нужны и могут быть удалены из рабочих таблиц компонента сбора данных. Компонент сбора данных периодически сокращает рабочие таблицы с целью уменьшения риска непредвиденных сбоев ресурсов.
8. Удаление данных из рабочих таблиц компонента сбора данных инициирует вызов компонента жизненного цикла источника.
9. Все успешно обработанные данные помечаются как готовые к удалению и в соответствии со стратегией хранения жизненного цикла источника удаляются из исходной базы данных.

Компоненты сбора данных и жизненного цикла источника обычно расположены в исходной системе; тогда как компоненты применения изменений, ETL и жизненного цикла приемника расположены в целевой системе, как показано на следующем рисунке:



В службе перемещения данных может использоваться несколько экземпляров компонентов в зависимости от структур данных в исходной и целевой базах. Количество экземпляров компонентов непосредственно зависит от числа групп бизнес-величин и числа исходных и целевых таблиц в модели бизнес-величин. Каждый экземпляр однозначно идентифицируется. В WebSphere Business Monitor применяются следующие правила:

- Один экземпляр компонента сбора данных назначается одному проекту модели бизнес-величин и собирает изменения во всех исходных таблицах, которые принадлежат этому проекту модели бизнес-величин.

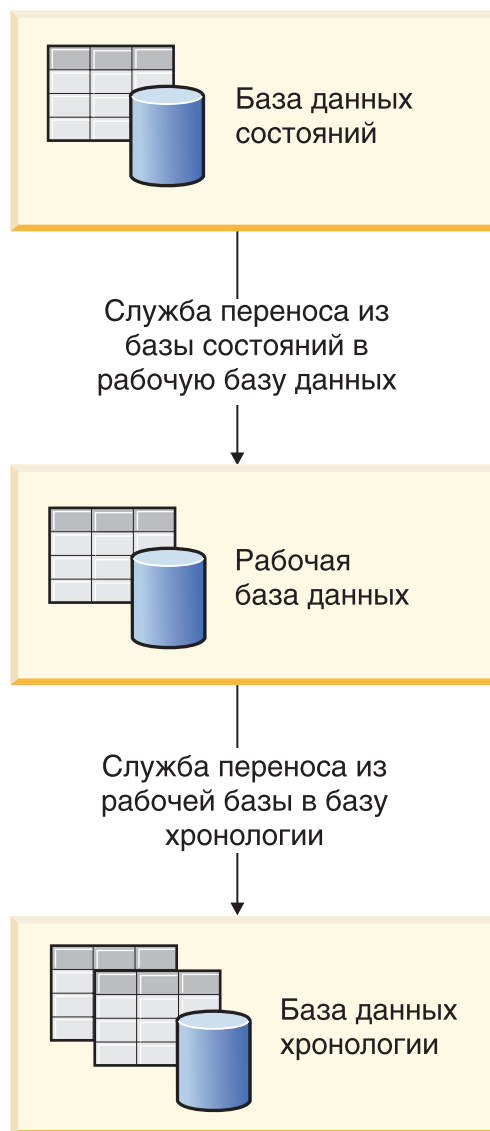
- Один экземпляр компонента применения изменений назначается одному проекту модели бизнес-величин и собирает изменения во всех целевых таблицах, которые принадлежат этому проекту модели бизнес-величин.
- Один экземпляр компонента ETL назначается одной целевой таблице.
- Один экземпляр компонента жизненного цикла источника назначается одной исходной таблице.
- Один экземпляр компонента жизненного цикла приемника назначается одной целевой таблице.

Экземпляром компонента может быть, например, программа, хранимая процедура или триггер базы данных.

В WebSphere Business Monitor используется два экземпляра службы перемещения данных:

- Служба перемещения данных из базы данных состояний в рабочую базу данных
- Служба перемещения данных из рабочей базы данных в базу данных хронологии.

Служба перемещения данных из базы состояний в рабочую базу обрабатывает данные, сохраненные сервером монитора в базе состояний, и перемещает их в рабочую базу, где к ним можно получить доступ через сводные панели. Другая служба перемещает данные из рабочей базы данных в базу данных хронологии. Это перемещение показано на следующей диаграмме:



Далее описываются конфигурации по умолчанию для этих служб, и способы их настройки, запуска, остановки и отслеживания.

#### **Служба перемещения данных из базы данных состояний в рабочую базу данных:**

Служба перемещения данных из базы состояний в рабочую базу обрабатывает данные, сохраненные сервером монитора в базе состояний, и перемещает их в рабочую базу, где к ним могут получить доступ другие компоненты WebSphere Business Monitor и службы перемещения данных из рабочей базы данных в базу данных хронологии.

Для этой службы перемещения данных применяется следующая конфигурация по умолчанию:

- Изменения в исходных таблицах сервера монитора (базы данных состояний) непрерывно собираются и записываются в рабочие таблицы.
- Записанные в рабочие таблицы изменения непрерывно распространяются компонентом применения изменений и вносятся в рабочие таблицы рабочей базы данных. Эти рабочие таблицы недоступны для других компонентов WebSphere Business Monitor и существуют для внутреннего использования.
- Компонент применения изменений вызывает компонент ETL как только появляются новые данные, требующие обработки. В соответствии с расписанием, в котором первоначально установлен период,

равный 5 минутам, компонент ETL обрабатывает данные, хранящиеся в рабочих таблицах применения изменений, или ожидает следующего выполнения по расписанию. Увеличение задержки между запланированными выполнениями приводит к тому, что данные дольше остаются в базе данных состояний после перемещения туда сервером монитора и перед перемещением их в целевые таблицы рабочей базы данных. После размещения в рабочей базе данных хронологии данные становятся доступными для других компонентов WebSphere Business Monitor.

- Все успешно обработанные компонентом ETL данные удаляются из рабочих таблиц применения изменений компонентом жизненного цикла приемника в соответствии с расписанием. По умолчанию этот компонент выполняется каждые 24 часа. Увеличение запланированной задержки приводит к увеличению размера рабочих таблиц. Уменьшение задержки также может вызвать непредвиденные проблемы из-за возможности одновременной попытки нескольких компонентов служб данных получить доступ к рабочим таблицам.
- Данные, которые были успешно перенесены из рабочих таблиц компонента сбора данных в рабочие таблицы компонента применения изменений, автоматически удаляются из рабочих таблиц компонента сбора данных по умолчанию каждые 5 минут.
- При каждом сокращении рабочих таблиц компонента сбора данных вызывается компонент жизненного цикла источника. Работа этого компонента также определяется расписанием. Он удаляет только те данные из исходных таблиц, которые помечены сервером монитора как готовые к удалению, если после последнего сокращения прошло не меньше пяти минут. Если для интервала сокращения компонента жизненного цикла задано значение меньше, чем для интервала сокращения компонента сбора данных, то сокращение будет производиться в соответствии с интервалом сокращения компонента сбора данных.

Например: для интервала сокращения рабочих таблиц компонента сбора данных задано значение 5 минут, а выполнение компонента жизненного цикла источника запланировано каждую минуту. Программа сбора данных сможет начать сокращение не раньше, чем через 5 минут. Поскольку процедура сокращения компонента сбора данных не активируется в течение 5 минут, компонент жизненного цикла не вызывается. По истечении 5 минут данные удаляются из рабочих таблиц, вызывается компонент жизненного цикла источника и удаляет данные из исходных таблиц базы данных состояний.

Конфигурация по умолчанию может быть изменена.

#### **Служба перемещения данных из рабочей базы данных в базу данных хронологии:**

Данные, перемещенные службой перемещения данных из рабочей базы данных в базу данных хронологии, сохраняются до явного удаления администратором базы данных (DBA). Успешно перемещенные в базу данных хронологии данные доступны для извлечения и анализа другими компонентами WebSphere Business Monitor.

Для этой службы перемещения данных применяется следующая конфигурация по умолчанию:

- Изменения в рабочей базе данных непрерывно собираются и записываются в рабочие таблицы. Отслеживаемые таблицы рабочей базы данных заполняются службой перемещения данных из базы данных состояний в рабочую базу данных.
- Записанные в рабочие таблицы изменения непрерывно распространяются компонентом применения изменений и вносятся в рабочие таблицы базы данных хронологии. Эти рабочие таблицы недоступны для других компонентов WebSphere Business Monitor и существуют для внутреннего использования.
- Компонент применения изменений вызывает компонент ETL как только появляются новые данные, требующие обработки. В соответствии с расписанием, в котором первоначально установлен период, равный 24 часам, компонент ETL обрабатывает данные, хранящиеся в рабочих таблицах применения изменений, или ожидает следующего выполнения по расписанию. Увеличение задержки между запланированными выполнениями приводит к тому, что данные дольше остаются в рабочей базе данных после перемещения туда и перед перемещением их в базу данных хронологии. После размещения в базе данных хронологии данные становятся доступными для других компонентов WebSphere Business Monitor.

**Примечание:** По причине зависимости от вызова компонентом применения изменений и конфигурации компонента применения изменений, компонент ETL может оказаться неспособным обрабатывать новые

данные каждые 24 часа (или в соответствии с установленным периодом времени). Поэтому расписание следует интерпретировать следующим образом: "не обрабатывать новые данные раньше, чем через 23 часа 59 минут после завершения последнего цикла обработки."

- Все успешно обработанные компонентом ETL данные удаляются из рабочих таблиц применения изменений компонентом жизненного цикла приемника в соответствии с расписанием. По умолчанию этот компонент выполняется каждые 24 часа. Увеличение запланированной задержки приводит к увеличению размера рабочих таблиц. Уменьшение задержки также может вызвать непредвиденные проблемы из-за возможности одновременной попытки нескольких компонентов служб данных получить доступ к рабочим таблицам.
- Данные, которые были успешно перенесены из рабочих таблиц компонента сбора данных в рабочие таблицы компонента применения изменений, автоматически удаляются из рабочих таблиц компонента сбора данных каждые 5 минут.
- При каждом сокращении рабочих таблиц компонента сбора данных вызывается компонент жизненного цикла источника. Работа этого компонента также определяется расписанием. Он удаляет только те данные из исходных таблиц в рабочей базе данных, которые помечены сервером монитора как готовые к удалению, и которые находятся в рабочей базе данных как минимум 24 часа. Интервал сокращения по умолчанию установлен равным 5 минутам. Если для интервала сокращения компонента жизненного цикла задано значение меньшее, чем для интервала сокращения компонента сбора данных, то сокращение будет производиться в соответствии с интервалом сокращения компонента сбора данных.

Например: для интервала сокращения рабочих таблиц компонента сбора данных задано значение 5 минут, а выполнение компонента жизненного цикла источника запланировано каждую минуту. Программа сбора данных сможет начать сокращение не раньше, чем через 5 минут. Поскольку процедура сокращения компонента сбора данных не активируется в течение 5 минут, компонент жизненного цикла не вызывается. По истечении 5 минут данные удаляются из рабочих таблиц, вызывается компонент жизненного цикла источника и удаляет данные из исходных таблиц рабочей базы данных.

Конфигурация по умолчанию может быть изменена.

## **Администрирование служб перемещения данных**

Для запуска или остановки службы перемещения данных для модели бизнес-величин необходимо определить связанные экземпляры главного компонента и включить или выключить их.

Все экземпляры компонента сбора данных (источника) и компонента применения изменений (приемника) считаются экземплярами главного компонента. Экземпляры компонента ETL и компонента жизненного цикла приемника зависят от экземпляров компонента применения изменений (приемника). Экземпляры компонента жизненного цикла источника зависят от компонента сбора данных (приемника), и их не нужно запускать или останавливать явным образом. В общем, все экземпляры компонента сбора данных (источника) в исходной базе данных и все экземпляры компонента применения изменений (приемника) должен запускать или останавливать пользователь. Только после того, как все эти экземпляры будут запущены или остановлены, служба перемещения данных считается запущенной или остановленной.

Экземпляр компонента сбора данных является эквивалентом программы сбора данных DB2, а компонент применения изменений является эквивалентом программы применения изменений DB2. Обе программы могут быть запущены (остановлены) или вручную с помощью сценариев, или с помощью расписания, в зависимости от того, в какой операционной системе установлена база данных. Готовые к использованию сценарии запуска и остановки создаются при развертывании. В системах Windows эти сценарии являются пакетными файлами. В системах UNIX они являются сценариями оболочки.

Каждая из программ сбора данных должна быть запущена в системе, в которой расположена DB2, содержащая исходную базу данных (базу данных состояний или рабочую базу данных в зависимости от соответствующей службы перемещения данных). Компонент сбора данных должен иметь локальный доступ к файлам протокола DB2.

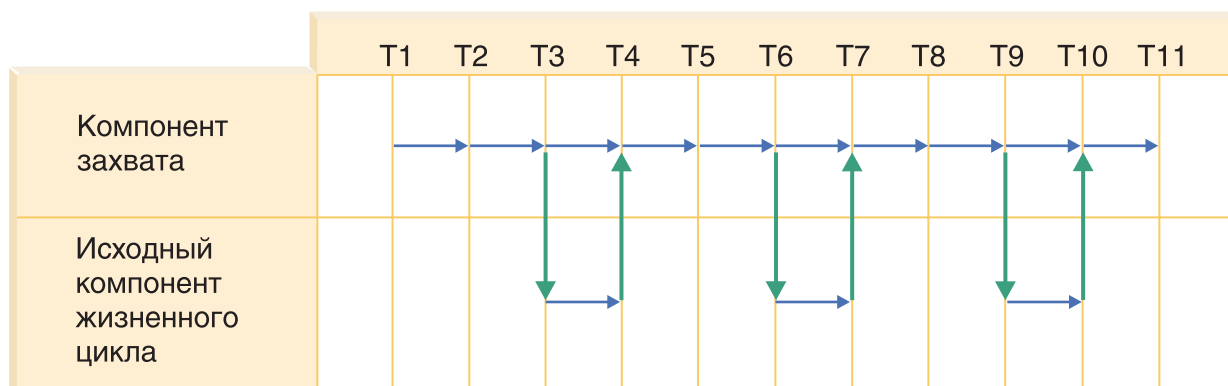
**Примечание:** Все изменения в исходной базе данных собираются, даже если экземпляры сбора данных не выполняются. Однако эти собранные изменения не обрабатываются и не применяются к целевой базе данных до тех пор, пока все экземпляры не будут функционировать.

## Конфигурация службы перемещения данных

Поведение и планирование каждого компонента Службы перемещения данных может быть по-разному настроено в зависимости от потребностей среды разработки и тестирования продуктов. Изменение конфигурации одного компонента может непосредственно повлиять на поведение другого компонента.

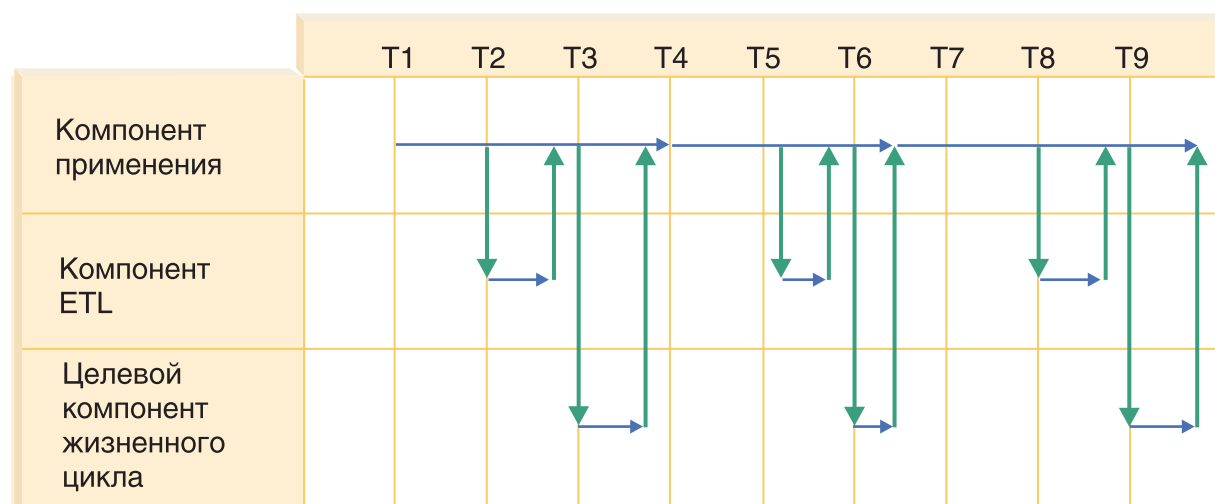
В общем, существует две зависимости:

- Компонент сбора данных периодически запускает компонент жизненного цикла источника. Если компонент сбора данных не работает, активизация компонента жизненного цикла источника не происходит. Можно настроить задержку между вызовами компонента жизненного цикла. На приведенном выше рисунке компонент жизненного цикла источника вызывается каждые 3 единицы



времени, выполняет некоторые действия и возвращает управление компоненту сбора данных, который продолжает обработку.

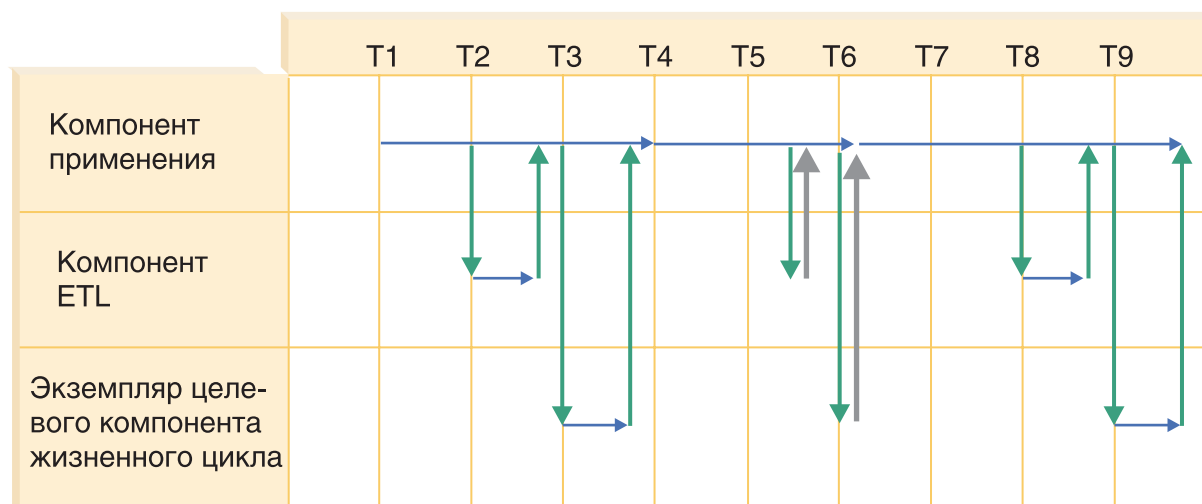
- Компонент ETL и компонент жизненного цикла приемника вызываются компонентом применения изменений после успешного перемещения данных из исходной базы данных в целевую базу данных. Компоненты ETL и жизненного цикла приемника будут вызваны, только если выполняется компонент применения изменений.



Поскольку существует необходимость выполнения зависящих друг от друга компонентов по различному расписанию, вызов не обязательно приводит к выполнению. Вместо этого каждый зависимый компонент при вызове проверяет свое расписание, и если время оказывается неподходящим для выполнения каких-либо

задач, то он возвращает управление вызвавшему его компоненту. В приведенном выше примере компоненты жизненного цикла приемника и ETL могут быть выполнены только дважды, если расписание для обоих компонентов ограничивает их вызов одним разом каждые 5 единиц времени.

Компонент ETL (и компонент жизненного цикла приемника) вызывается и выполняется в момент T2



(соответственно, T3). Следующий вызов происходит приблизительно в момент T6. Поскольку с момента их последнего выполнения прошло менее 5 единиц времени, управление немедленно возвращается к компоненту применения изменений. Последующий вызов приблизительно в момент T8 (соответственно T9) приводит к выполнению, поскольку прошло более 5 единиц времени. Каждый компонент реализован в виде одного или нескольких экземпляров. Экземпляры можно настраивать по отдельности для более гибкого управления.

**Примечание:** Изменения вступают в силу немедленно, если не указано иначе.

По умолчанию конфигурации компонентов сбора данных и применения изменений могут быть изменены или путем соответствующего изменения управляющих таблиц, или с помощью параметров команд в сценариях запуска. Компоненты вызова жизненного цикла и ETL можно настроить с помощью одной из управляющих таблиц.

Для настройки компонентов Службы перемещения данных в зависимости от потребностей среды разработки, тестирования и выполнения необходимо выполнить следующие действия.

#### Настройка экземпляров компонента сбора данных (Исходного)

Экземпляр компонента сбора данных является эквивалентом утилиты репликации сбора данных DB2. По умолчанию эта утилита настроена на постоянный сбор измененных данных в исходных таблицах и запись этих изменений во внутренние рабочие таблицы. В общем случае, нет необходимости менять настройки по умолчанию для экземпляров компонента сбора данных.

##### • Идентификация экземпляров компонента сбора данных.

Для сбора данных, связанных с моделью бизнес-величин, используется несколько экземпляров компонента сбора данных (или утилит сбора данных DB2). Для того чтобы выяснить, какие утилиты сбора данных назначены для обслуживания модели бизнес-величин, выполните следующие действия:

- Определите, для какой службы перемещения данных вы хотите изменить конфигурацию утилиты сбора данных.
- Проверьте содержимое таблицы метаданных WBIRMADM.RMMETADATA в базе данных состояний (для службы перемещения данных из базы данных состояний в рабочую базу данных) или в рабочей базе данных (для службы перемещения данных из рабочей базы данных в базу данных хронологии) и определите имена всех утилит сбора данных (поле SRC\_RM\_CAP\_SVR\_NAME).

**Пример:** По запросу "SELECT OM\_NAME, SRC\_TAB\_NAME, SERVICE\_NAME, SRC\_RM\_CAP\_SVR\_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE\_NAME='State to Runtime' " может быть выдано:

В приведенном примере утилита сбора данных CAPTURE\_1 назначена для сбора всех измененных

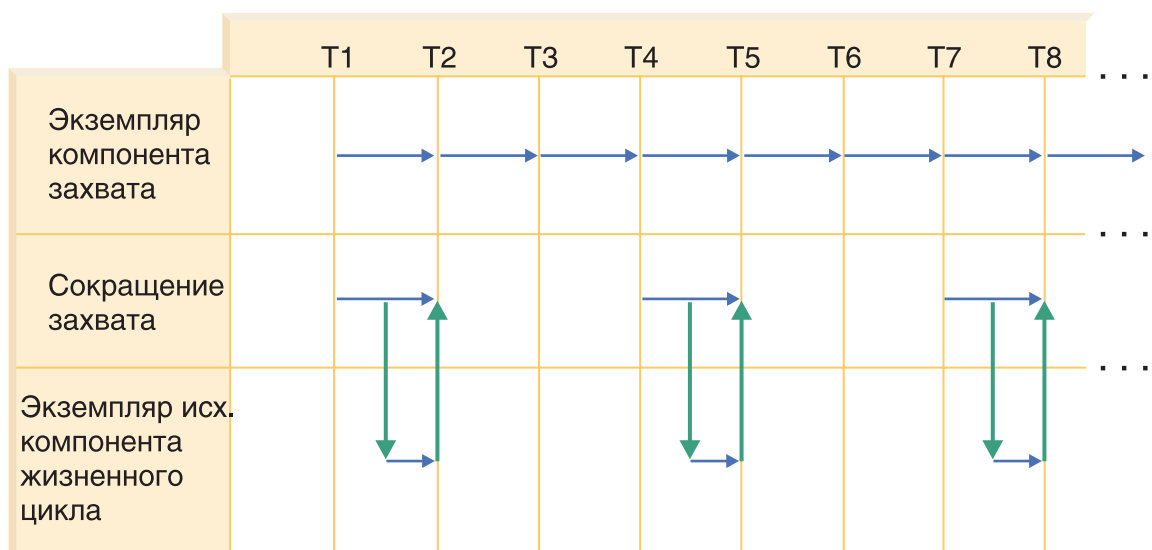
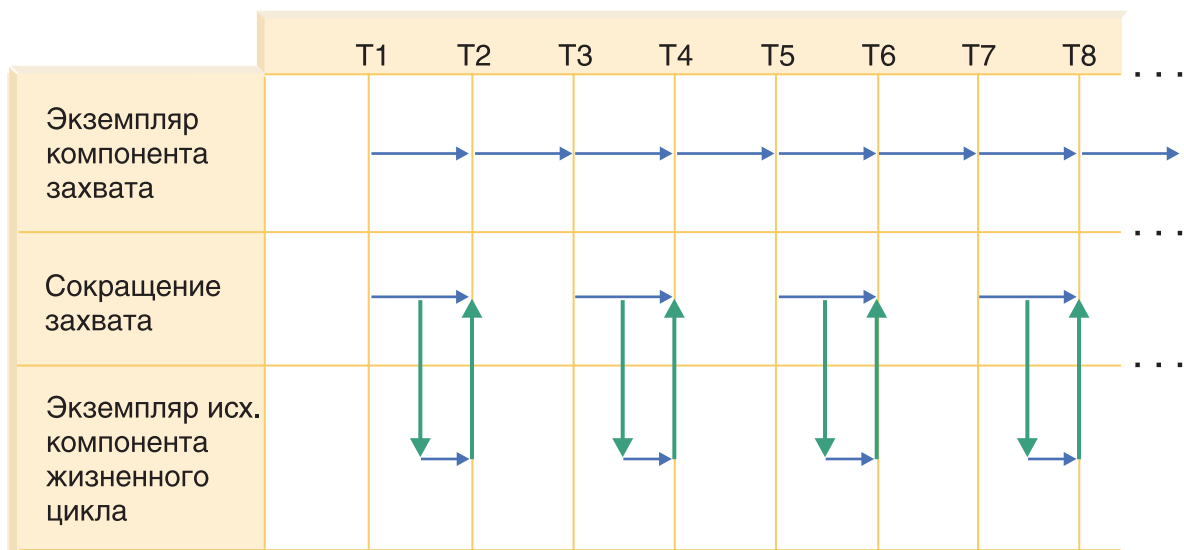
OM_NAME	SRC_TAB_NAME	SERVICE_NAME	SRC_RM_CAP_SVR_NAME
STEW_S	wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	State to Runtime	CAPTURE_1
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJR5HFQQGE	State to Runtime	CAPTURE_1

данных в двух исходных таблицах, связанных с моделью бизнес-величин STEW\_S в базе данных состояний.

- **Изменение интервала сокращения рабочей таблицы сбора данных.**

Утилиты сбора данных автоматически сокращают свои рабочие таблицы каждые 300 секунд (значение по умолчанию параметра `prune_interval`), если включено автоматическое сокращение (параметр `autoprune` равен "y"). Каждая операция сокращения автоматически вызывает экземпляр компонента жизненного цикла источника с помощью триггера базы данных. Поэтому изменение значения параметра интервала сокращения для утилиты сбора данных непосредственно влияет на частоту сокращения исходной таблицы компонентом жизненного цикла источника. Ниже приведен пример, который иллюстрирует, как изменение интервала сокращения для утилиты сбора данных может повлиять на вызов экземпляра компонента жизненного цикла источника.





Увеличение значения параметра `prune_interval` экземпляра утилиты сбора данных от 2 единиц времени (например, 300 секунд) до 3 единиц времени (например, 450 секунд) приводит к тому, что:

- Подлежащие удалению записи рабочих таблиц сбора данных продолжают дольше оставаться в рабочих таблицах. Следовательно, увеличиваются возможные требования к объему памяти. Размеры рабочих таблиц возрастают, но загрузка системы и риск возникновения аварийных ситуаций уменьшается.
- Записи исходных таблиц, удаленные на основании стратегии сохранения жизненного цикла, остаются в исходных таблицах дольше, чем ожидалось.

Вообще, если параметр `prune_interval` утилиты сбора данных имеет большее значение, чем параметр `prune_interval` компонента жизненного цикла, то более высокий приоритет имеет значение параметра утилиты сбора данных. Если утилита сбора данных не выполняется, или если выключена функция автоматического сокращения, то не выполняется никаких активаций жизненного цикла источника.

### Настройка компонента жизненного цикла источника

В каждой исходной базе данных (состояний и рабочей) используются множественные экземпляры компонента жизненного цикла. Каждый экземпляр, реализованный с помощью триггера, осуществляет стратегию сохранения, определенную в таблице `WBIRMADM.RMPRUNECTRL`, расположенной в исходной

базе данных для службы перемещения данных. Стратегии хранения жизненного цикла задаются на основе таблиц. Таким образом, одна запись в WBIRMADM.RMPRUNECTRL соответствует одной таблице, для которой необходимо сокращение.

TABLE_NAME	RETEN...	LAST_PRUNED	PRUNE_IN...	PRUNE_EN...	LOG...	ROW...
wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI	1440	Oct 11, 2005 4:40:...	5	1	0	0
wbi.AIR_BVSOYAP1DRWFD5HNQJR5HFQQQE	1440	Oct 11, 2005 4:40:...	5	1	0	0

- **Идентификация экземпляров компонента жизненного цикла источника.**

Для того чтобы определить, какие триггеры назначены для осуществления стратегий хранения для данной модели бизнес-величин, выполните следующие действия:

- Определите, для какой службы перемещения данных вы хотите изменить конфигурацию ETL.
- Проверьте содержимое таблицы метаданных WBIRMADM.RMMETADATA в базе данных состояний (для службы перемещения данных из базы данных состояний в рабочую базу данных) или в рабочей базе данных (для службы перемещения данных из рабочей базы данных в базу данных хронологии) и определите имена всех связанных триггеров в поле SRC\_RM\_PRUNE\_TRG\_NAME.

**Пример:** По запросу "SELECT OM\_NAME, SRC\_TAB\_NAME, SERVICE\_NAME, SRC\_RM\_PRUNE\_TRG\_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE\_NAME='State to Runtime'" может быть выдано:

В этом примере два триггера (WBIRMADM.MCPruneTrig\_8 и WBIRMADM.MCPruneTrig\_9)

OM_NAME	SRC_TAB_NAME	SERVICE_NAME	SRC_RM_PRUNE_TRG_NAME
STEW_S	wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	State to Runtime	WBIRMADM.MCPruneTrig_8
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE	State to Runtime	WBIRMADM.MCPruneTrig_9

осуществляют стратегию хранения жизненного цикла для исходных таблиц модели бизнес-величин STEW\_S в базе данных состояний. Поскольку стратегии хранения определены с помощью таблицы, а не именами экземпляров компонента жизненного цикла, отслеживайте значения поля SRC\_TAB\_NAME при планировании изменения режима вызовов жизненного цикла.

- **Изменение конфигурации экземпляра компонента жизненного цикла источника .**

- Включение и выключение экземпляров компонента жизненного цикла:

Сокращение может значительно повысить производительность системы. Когда сокращение включено, это уменьшает объем информации, которую должны обрабатывать сервера транзакций (статические) и сервера отчетов (динамические). Это также немного увеличивает нагрузку на те же самые таблицы при каждом вызове в соответствии со значениями параметров компонента жизненного цикла. Когда сокращение выключено, объем исходных таблиц будет некоторое время увеличиваться, что может привести к снижению быстродействия.

По умолчанию исходные таблицы автоматически сокращаются в соответствии со стратегией хранения жизненного цикла. Для временного выключения сокращения измените соответствующие записи в WBIRMADM.RMPRUNECTRL: установите значение поля PRUNE\_ENABLED равным 1 для включения сокращения или равным любому другому численному значению (предпочтительно нулю) для его выключения.

Записи исходной таблицы wbi.CTX\_TQ4MUFT42JOT5F6R3KSDQDE2UI будут очищены, но никаких записей таблицы wbi.AI\_BVSOYAP1DRWFD5HNQJR5HFQQQE очищено не будет, если используется следующая конфигурация. По запросу: "SELECT TABLE\_NAME, PRUNE\_ENABLED FROM WBIRMADM.RMPRUNECTRL" может быть выдано:

TABLE_NAME	PRUNE_ENABLED
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	1
wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE	0

- Изменение стратегии хранения:

Стратегии хранения могут быть изменены только для исходных таблиц, расположенных в рабочей базе данных. Для всех таблиц, расположенных в базе данных состояний, время хранения принудительно

устанавливается равным 0, независимо от значений параметров в WBIRMADM.RMPRUNECTRL. Период хранения определяется как минимальное время, в течение которого запись должна сохраняться в исходной таблице до ее удаления при наличии двух признаков. Только один из этих двух признаков можно настроить с помощью управляющих таблиц: время хранения, заданное в минутах. Каждая запись, помеченная как готовая для удаления, которая остается в исходной таблице не менее времени RETENTION\_IN\_MINUTES, подлежит удалению.

Для исходных таблиц рабочей базы данных значение времени хранения по умолчанию составляет один день (1440 минут), в течение которого записи, помеченные сервером как готовые к удалению, должны оставаться в таблицах. По запросу: "SELECT TABLE\_NAME, RETENTION\_IN\_MINUTES FROM WBIRMADM.RMPRUNECTRL" может быть выдано:

Измененные записи управляющей таблицы WBIRMADM.RMPRUNECTRL собираются при каждом

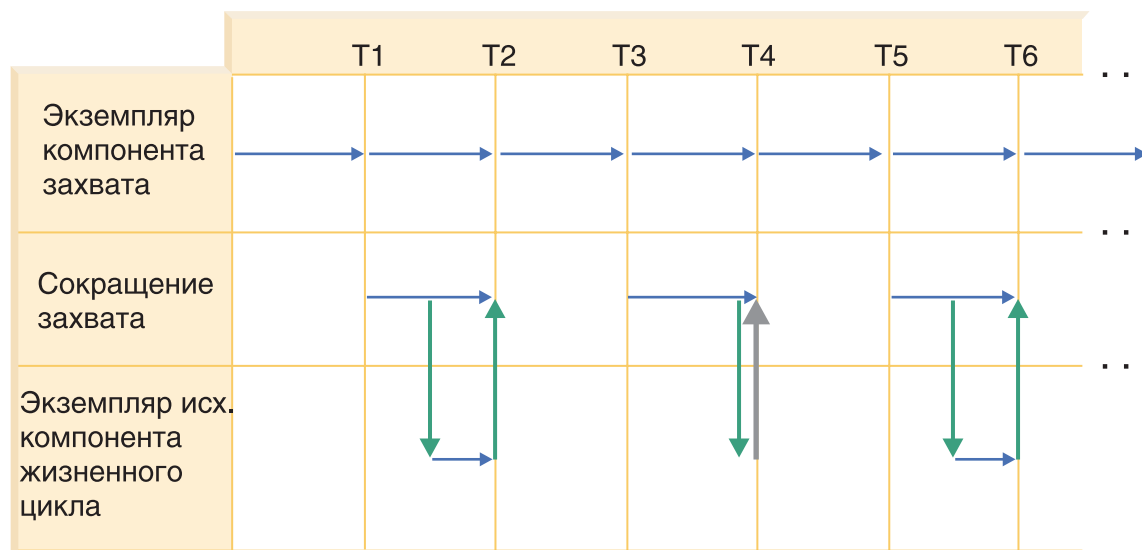
TABLE_NAME	RETENTION_IN_MINUTES
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	1440
wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE	1440

вызове компонента жизненного цикла источника.

- Планирование сокращения исходных данных:

Существует зависимость между интервалом сокращения рабочей таблицы сбора данных и вызовом компонента жизненного цикла источника. Вызов не приводит к выполнению компонента, если прошло недостаточно времени после последнего вызова экземпляра компонента жизненного цикла источника, как показано на следующем рисунке:

Допустим, что запланировано выполнение компонента жизненного цикла источника каждые 4 единицы



времени, но компонент сбора данных настроен на сокращение своих рабочих таблиц каждые 2 единицы времени. Тогда вызов в момент T4 не приведет к выполнению.

Для изменения расписания, установленного по умолчанию, найдите соответствующие записи в WBIRMADM.RMPRUNECTRL и измените значение поля PRUNE\_INTERVAL, в котором указана минимальная задержка между выполнениями в минутах.

Увеличение этого значения приведет к меньшей частоте выполнений (но число вызовов останется

TABLE_NAME	LAST_PRUNED	PRUNE_INTERVAL
wbi.CTX_TQ4MUFT42JOT5F6R3KSDQDE2UI	Oct 11, 2005 5:16:44 PM ...	5
wbi.AI_BVSOYAP1DRWFD5HNQJR5HFQQQE		5

прежним). При каждом выполнении определяется, какие записи исходной таблицы подлежат удалению,

и они удаляются. Регулярно производите отслеживание исходных баз данных для обнаружения и устранения потенциальных причин снижения производительности, связанных с блокировками, возникшими в результате этих удалений.

## Настройка компонента применения изменений (приемника)

Экземпляр компонента применения изменений - это утилита репликации применения изменений DB2. Изменения, собранные утилитами сбора данных, непрерывно применяются к промежуточным таблицам в целевой базе данных по умолчанию. Значения по умолчанию параметров настройки утилиты должны подходить для большинства сред, и их не следует изменять.

- **Идентификация экземпляров компонента применения изменений.**

Для внесения изменений данных во внутренние промежуточные таблицы, связанные с моделью бизнес-величин, используется несколько экземпляров компонента применения изменений (утилит применения изменений DB2). Для определения того, какие утилиты применения изменений назначены для обслуживания модели бизнес-величин, выполните следующие действия:

- Определите, для какой службы перемещения данных вы хотите изменить конфигурацию утилиты применения изменений.
- Проверьте содержимое таблицы метаданных WBIRMADM.RMMETADATA в базе данных состояний (для службы перемещения данных из базы данных состояний в рабочую базу данных) или в рабочей базе данных (для службы перемещения данных из рабочей базы данных в базу данных хронологии) и определите имена всех утилит применения изменений (поле TGT\_RM\_APP\_SVR\_NAME). По запросу: "SELECT OM\_NAME, SRC\_TAB\_NAME, SERVICE\_NAME, TGT\_RM\_APP\_SVR\_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE\_NAME='State to Runtime'" может быть выдано:

В этом примере все измененные данные в модели бизнес-величин STEW\_S, собранные в базе данных

OM_NAME	SRC_TAB_NAME	SERVICE_NAME	TGT_RM_APP_SVR_NAME
STEW_S	wbi.CTX_TQ4MUF142JOT5F6R3KSDODE2UI	State to Runtime	APPLY_4
STEW_S	wbi.AI_BVS0YAP1DRWFD5HNGJR5HFQGGQE	State to Runtime	APPLY_4

состояний, вносятся в промежуточные таблицы рабочей базы данных утилитой применения изменений APPLY\_4.

Каждый раз, когда утилита применения изменений завершает обработку всех (подтвержденных) изменений, записанных к этому моменту утилитой сбора данных, вызываются один или несколько экземпляров компонента ETL и экземпляры компонента жизненного цикла приемника.

## Настройка компонента ETL

Компоненты ETL реализованы в WebSphere Business Monitor в виде хранимых процедур базы данных. Эти хранимые процедуры всегда расположены в целевой базе данных для всех служб перемещения данных. Поэтому все хранимые процедуры ETL, назначенные для службы перемещения данных из базы данных состояний в рабочую базу данных, расположены в рабочей базе данных, а хранимые процедуры ETL, назначенные для службы перемещения данных из рабочей базы данных в базу данных хронологии, расположены в базе данных хронологии.

- **Идентификация экземпляров компонента ETL.**

Для обработки данных, добавленных во внутренние промежуточные таблицы, связанные с моделью бизнес-величин, используется несколько экземпляров компонента ETL. Для того чтобы выяснить, какие хранимые процедуры назначены для обслуживания заданной модели бизнес-величин, выполните следующие действия:

- Определите, для какой службы перемещения данных вы хотите изменить конфигурацию ETL.
- Проверьте содержимое таблицы метаданных WBIRMADM.RMMETADATA в базе данных состояний (для службы перемещения данных из базы данных состояний в рабочую базу данных) или в рабочей базе данных (для службы перемещения данных из рабочей базы данных в базу данных хронологии) и определите имена всех хранимых процедур ETL (поле TGT\_RM\_SPETL\_NAME). По следующему запросу: "SELECT OM\_NAME, SRC\_TAB\_NAME, TGT\_TAB\_NAME, SERVICE\_NAME,

TGT\_RM\_SPETL\_NAME FROM WBIRMADM.RMMETADATA WHERE SERVICE\_NAME='State to Runtime'" может быть выдано:

В этом примере, все измененные данные в модели бизнес-величин STEW\_S, собранные в базе данных

OM_NAME	SRC_TAB_NAME	TGT_TAB_NAME	SERVICE_NAME	TGT_RM_SPETL_NAME
STEW_S	wbi.CTX_TG4MUFT42JOT5F6R3KSDQDE2UI	wbi.CTR_TG4MUFT42JOT5F6R3KSDQDE2UI	State to Runtime	WBIRMADM.WBIRMSP_10
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJR5HFQGGQE	wbi.AIR_BVSOYAP1DRWFD5HNGJR5HFQGGQE	State to Runtime	WBIRMADM.WBIRMSP_14

состояний и внесенные в промежуточные таблицы в рабочей базе данных, будут обрабатываться хранимыми процедурами с именами WBIRMADM.WBIRMSP\_10 и WBIRMADM.WBIRMSP\_14.

Успешно обработанные данные сохраняются в целевых таблицах (определяемых в поле TGT\_TAB\_NAME) рабочей базы данных.

#### • Изменение конфигурации экземпляра компонента ETL.

Конфигурации экземпляров компонента ETL хранятся в управляющей таблице WBIRMADM.RMCONTROL. Для экземпляров, назначенных для службы перемещения данных из базы данных состояний в рабочую базу данных, конфигурации хранятся в рабочей базе данных; а для всех остальных - в базе данных хронологии. Все изменения, внесенные в конфигурацию, будут собраны хранимыми процедурами при следующем запуске. С помощью управляемой таблицы можно настроить три опции:

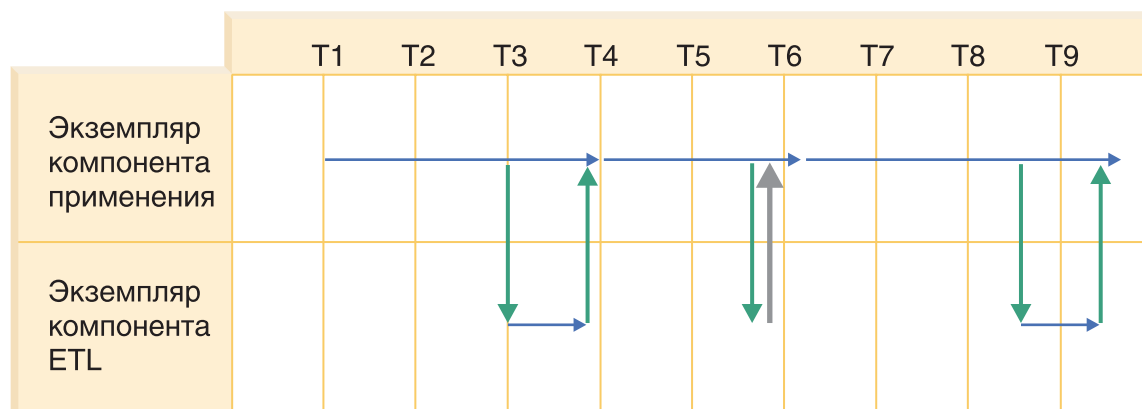
- Минимальное время между двумя выполнениями ETL (ETLSCHEDMETHOD, ETL\_0\_MINUTES)
- Уровень дискретности вывода протокола (LOGLEVEL)
- Продолжительность транзакции (COMMITINTERVAL)

Каждая запись этой таблицы соответствует одному экземпляру компонента ETL, который, в свою очередь, соответствует одной целевой таблице, требующей заполнения. Следующий пример иллюстрирует влияние изменений конфигурации на поведение экземпляра.

TARGETTABLE	TGT_RM_SPETL_NAME	ETLSCHEDMETHOD	ETL_0_MINUTES	LOGLEVEL	COMMITINTERVAL
wbi.CTR_TG4MUFT42JOT5F6R3KSDQDE2UI	WBIRMADM.WBIRMSP_10	1	5	0	1000
wbi.AIR_BVSOYAP1DRWFD5HNGJR5HFQGGQE	WBIRMADM.WBIRMSP_14	1	5	0	1000

#### • Изменение расписания ETL.

Экземпляры компонента ETL вызываются при каждом завершении обработки набора подписки экземпляром компонента применения изменений. При вызове экземпляр ETL проверяет свое расписание, а затем или начинает обработку, или возвращает управление непосредственно экземпляру компонента применения изменений. Он использует информацию из управляющей таблицы WBIRMADM.RMCONTROL для определения того, есть ли необходимость в выполнении или нет. На следующем рисунке показана разница между вызовом и выполнением: при первом и третьем вызовах экземпляр компонента ETL выполняется в соответствии с расписанием. Второй вызов происходит не по расписанию и не приводит ни к каким действиям.



На частоту выполнения экземпляров компонента ETL в Службе перемещения данных из базы данных состояний в рабочую базу данных и в Службе перемещения данных из рабочей базы данных в базу данных хронологии влияет ряд факторов:

- Коэффициент готовности: определяет, насколько быстро данные должны стать доступны в целевой таблице. Выбор меньшего интервала приведет к тому, что данные станут доступны быстрее, но при этом увеличится загрузка системы.
- Объем данных: Утилиты репликации постоянно (или в соответствии с настройками) поставляют данные в промежуточные таблицы, независимо от того, обработаны ли они экземплярами компонента ETL или нет. Чем больший объем данных необходимо обрабатывать, тем больше используется ресурсов базы данных. Более частая обработка данных может уменьшить максимальное использование ресурсов.
- Время обработки: Обработка компонентом ETL рабочей базы данных занимает меньше времени, чем обработка базы данных хронологии. Составляйте расписания соответственно. Применение небольшой задержки между выполнениями не повысит производительность, если выполнение происходит дольше указанной в расписании задержки. Например, если экземпляр компонента ETL совершает обработку за 60 секунд, то установленный в расписании 30-секундный интервал в действительности превратится в 60-секундный, поскольку экземпляры компонентов ETL выполняются последовательно.

В настоящее время поддерживаются два режима планирования:

- Гибкое расписание:

Экземпляр ETL выполняется, только если как минимум ETL\_0\_MINUTES прошло после его последнего выполнения (LASTUPDATED). Например, предположим, что управляющая таблица содержит следующую информацию.

Хранимая процедура WBIRMADM.WBIRMSP\_10 выполнится не раньше, чем 11 октября 2005 в

TGT_RM_SPETL_NAME	TARGETTABLE	LASTUPDATED	ETL_0_MINUTES
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI	Oct 11, 2005 5:20:20 PM	60
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HINQJR5HFQQQE	Oct 11, 2005 5:20:20 PM	60

18:20:20 (11 октября 2005 17:20:20 + 60 минут). Расписания могут проскальзывать, если хранимая процедура будет вызвана позже, чем 11 октября 2005 года в 18:20:20. Предположим, что текущее время 19 часов, и хранимая процедура не была выполнена, как ожидалось, в 18:20. Хранимая процедура вызывается и выполняется (примерно на 40 минут позже). Она не будет выполняться повторно до, по крайней мере, 19:00+ 60 минут = 20:00. Запланированное выполнение пропущено, поскольку выполнение процедур ETL было запланировано на 20 минут после начала каждого часа, но теперь происходит в начале каждого часа. При необходимости можно сбросить расписание, изменив значение системного времени в поле LASTUPDATED.

Используйте этот способ планирования, если не требуется выполнение в фиксированное время. Для включения этого вида планирования установите значение поля ETLSCHEDMETHOD в таблице WBIRMADM.RMCONTROL равным 0 для всех процедур, которые присвоены одной группе бизнес-величин:

TGT_RM_SPETL_NAME	TARGETTABLE	ETLSCHEDMETHOD
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI	0
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HINQJR5HFQQQE	0

- Фиксированное расписание:

Это расписание установлено по умолчанию для всех компонентов ETL. Экземпляры компонента ETL выполняются, если текущее время больше NEXTSTARTTIME. Для того чтобы избежать пропусков, при каждом выполнении хранимой процедуры вычисляется время следующего выполнения на основе текущего времени и предыдущего запланированного времени выполнения. Следующий пример иллюстрирует подобное расписание:



TGT_RM_SPETL_NAME ⇅	TARGETTABLE ⇅	LASTUPDATED ⇅	ETL_O_MINUTES ⇅	NEXTSTARTTIME ⇅
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	Oct 11, 2005 5:20:20 PM ...	60	Oct 11, 2005 6:20:20 PM ...
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HNQJR5HFQQQE ...	Oct 11, 2005 5:20:20 PM ...	60	Oct 11, 2005 6:20:20 PM ...

Предположим, что текущее время равно 19:00 и по некоторым причинам хранимая процедура не была выполнена, как ожидалось, в 18:20. Хранимая процедура выполняется, поскольку текущее время больше NEXTSTARTTIME (18:20), а дата та же. Следующее выполнение будет запланировано на 19:20, в соответствии с первоначальным расписанием, а не на 20:00, как в случае гибкого расписания. Используйте этот метод планирования, если хранимые процедуры должны выполняться в пределах установленного промежутка времени. Для включения этого вида планирования установите значение поля ETLSCHEDMETHOD в таблице WBIRMADM.RMCONTROL равным 1 для всех процедур, которые присвоены одной группе бизнес-величин:

TGT_RM_SPETL_NAME ⇅	TARGETTABLE ⇅	ETLSCHEDMETHOD ⇅
WBIRMADM.WBIRMSP_10	wbi.CTR_TQ4MUFT42JOT5F6R3KSDQDE2UI ...	1
WBIRMADM.WBIRMSP_14	wbi.AIR_BVSOYAP1DRWFD5HNQJR5HFQQQE ...	1

Настоятельно рекомендуется использовать одинаковый метод планирования для экземпляров компонента ETL, принадлежащих одной модели бизнес-величин, по причине их взаимозависимости. Это особенно важно для базы данных хронологии и для расписаний с большими интервалами (несколько часов или более). Если для ETLSCHEDMETHOD указано значение, отличное от 0 или 1, то экземпляр компонента ETL будет выключен.

- **Изменение уровня ведения протоколов.**

Хранимые процедуры поддерживают два уровня ведения протоколов: минимальный (0) и максимальный (1). По умолчанию установлено минимальное значение. При необходимости его можно изменить в поле LOGLEVEL таблицы WBIRMADM.CONTROL для некоторых хранимых процедур (TGT\_RM\_SPETL\_NAME). Весь вывод протокола добавляется к таблице WBIRMADM.RMLOG. В следующем примере обе хранимые процедуры WBIRMADM.WBIRMSP\_10 и WBIRMADM.WBIRMSP\_14 используют минимальный уровень ведения протокола:

Поскольку таблица протокола не сокращается автоматически, ее необходимо регулярно отслеживать с

ENTRYSTMP ⇅	NAME ⇅	OPERATION ⇅	ISTRACEENTRY ⇅	ID
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	SP_START	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	DEL_TEMP	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	INS_TEMP	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	FETCH_TARGET_...	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_14	SP_END	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_10	SP_START	0	
Oct 11, 2005 4:40:20 PM ...	WBIRMADM.WBIRMSP_10	DEL_TEMP	0	

помощью DBA. Используйте минимальный уровень ведения протокола, если не указано иначе.

- **Изменение продолжительности транзакций.**

Успешно обработанные хранимыми процедурами данные немедленно записываются в целевые таблицы. Однако, в зависимости от значения интервала фиксации (поле COMMITINTERVAL таблицы WBIRMADM.RMCONTROL) изменения в целевой таблице не фиксируются до тех пор, пока не будет обработано заданное количество записей, или пока не останется записей для обработки. Увеличение значения COMMITINTERVAL (например, до 1500) приведет к тому, что хранимая процедура будет обрабатывать большее количество данных перед выполнением фиксации изменений. Блокировки целевой таблицы будут более длительными, и это может негативно повлиять на другие компоненты, которые пытаются получить доступ к этой таблице. При уменьшении продолжительности (например, до 500) меньшее количество записей обрабатывается до фиксации изменений в целевой таблице, и блокировки освобождаются раньше.

TARGETTABLE	TGT_RM_SPETL_NAME	ETL_0_MINUTES	LOGLEVEL	COMMITINTERVAL
wbi.CTR_TQ4MFT42JOTSF6R3KSDGDE2UI ...	WBIRMADM.WBIRMSP_10	5	0	1500
wbi.AIR_BVSOYAP1DRWFD5HNGJRSHFQGGQE...	WBIRMADM.WBIRMSP_14	5	0	500

## Настройка компонента жизненного цикла приемника

Объем рабочих таблиц ETL постоянно растет по мере внесения или изменения данных экземплярами компонента применения изменений. Один экземпляр компонента жизненного цикла приемника, реализованный с помощью хранимой процедуры, назначается одной рабочей таблице в каждой целевой базе данных (рабочей или хронологии). Каждый экземпляр осуществляет внутренние стратегии хранения, определенные в управляющей таблице WBIRMADM.RMPRUNECTRL. Так же, как и в случае исходных таблиц, стратегии хранения жизненного цикла для рабочих таблиц ETL задаются на основе таблиц. Таким образом, одна запись в WBIRMADM.RMPRUNECTRL соответствует одной таблице, для которой необходимо сокращение.

### • Идентификация экземпляров компонента жизненного цикла приемника.

Для определения того, какие хранимые процедуры назначены для осуществления стратегий хранения рабочей таблицы ETL для данной модели бизнес-величин:

- Определите, для какой службы перемещения данных вы хотите изменить конфигурацию ETL.
- Проверьте содержимое таблицы метаданных WBIRMADM.RMMETADATA в рабочей базе данных (для службы перемещения данных из базы данных состояний в рабочую базу данных) или в базе данных хронологии (для службы перемещения данных из рабочей базы данных в базу данных хронологии) и определите имена всех связанных хранимых процедур в поле TGT\_RM\_APP\_PRUNE\_SP\_NAME следующей таблицы.

В этом примере две хранимых процедуры (WBIRMADM.WBIRMSP\_13 и

OM_NAME	SRC_TAB_NAME	TGT_RM_APP_STG_TAB_NAME	TGT_RM_APP_PRUNE_SP_NAME	SERVICE_NAME
STEW_S	wbi.CTX_TQ4MFT42JOTSF6R3KSDGDE2UI	APP.CCD_6	WBIRMADM.WBIRMSP_P_13	State to Runtime
STEW_S	wbi.AI_BVSOYAP1DRWFD5HNGJRSHFQGGQE	APP.CCD_7	WBIRMADM.WBIRMSP_P_17	State to Runtime

WBIRMADM.WBIRMSP\_P\_17) осуществляют стратегию хранения жизненного цикла для рабочих таблиц ETL модели бизнес-величин STEW\_S в рабочей базе данных. Поскольку стратегии хранения определены с помощью таблицы, а не именами экземпляров компонента жизненного цикла, отслеживайте значения поля TGT\_RM\_APP\_STG\_TAB\_NAME при планировании изменения режима вызовов жизненного цикла.

### • Изменение конфигурации экземпляра компонента жизненного цикла приемника

Конфигурации по умолчанию должны подходить для большей части развертываний, однако их можно настроить, как это описано в следующих разделах.

- Включение и выключение экземпляров компонента жизненного цикла приемника.

По умолчанию рабочие таблицы ETL автоматически сокращаются в соответствии со стратегией хранения жизненного цикла. Для временного выключения сокращения измените соответствующие записи в WBIRMADM.RMPRUNECTRL: установите значение поля PRUNE\_ENABLED равным 1 для включения сокращения или равным любому другому численному значению (предпочтительно нулю) для его выключения. Рабочие таблицы ETL автоматически сокращаются, если рабочая таблица WBIRMADM.RMPRUNECTRL содержит следующие записи в рабочей базе данных:

Перед выключением экземпляров компонента жизненного цикла приемника, убедитесь в доступности

TABLE_NAME	PRUNE_ENABLED
APP.CCD_6	1
APP.CCD_7	1

достаточного объема памяти в контейнере табличного пространства для связанной таблицы. При каждом изменении сервером монитора записей в исходных таблицах в рабочие таблицы ETL добавляется одна запись. Таким образом, одна запись в исходной таблице может быть временно представлена несколькими записями в рабочих таблицах, что приводит к более быстрому росту объема рабочих таблиц по сравнению с исходными. Изменения в WBIRMADM.RMPRUNECTRL будут собраны при следующем вызове экземпляра компонента жизненного цикла приемника.



— Изменение стратегии хранения.

Все записи, успешно обработанные экземплярами компонента ETL, могут быть удалены из рабочих таблиц. И для рабочей базы данных, и для базы данных хронологии, по умолчанию установлен период хранения 0 минут:

Все подходящие записи будут удалены при следующем вызове экземпляра компонента жизненного

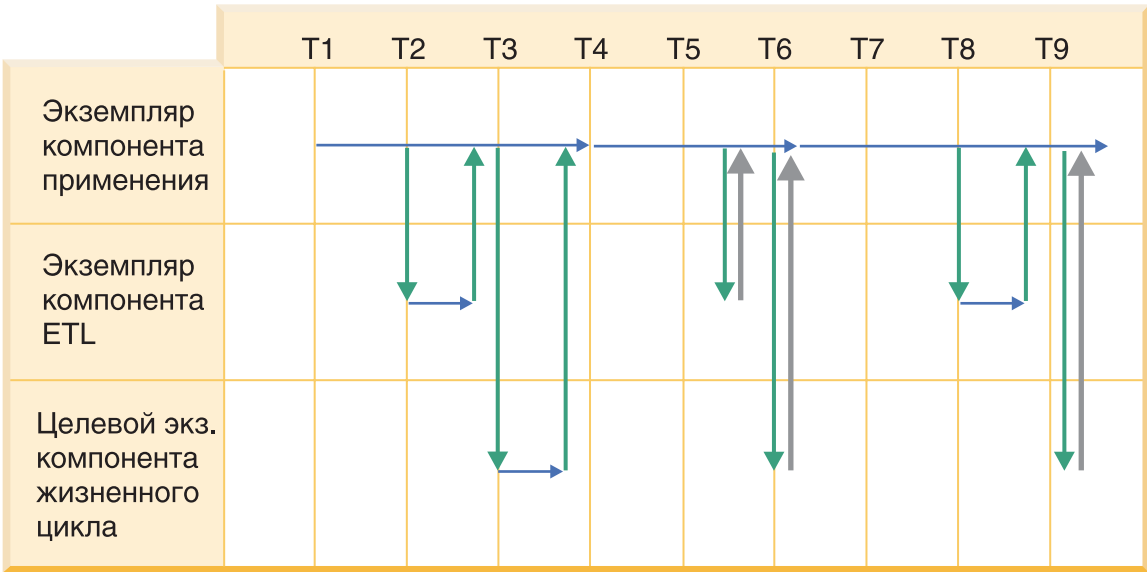
TABLE_NAME	RETENTION IN MINUTES
APP.CCD_6	0
APP.CCD_7	0

цикла приемника. Установка периода хранения равным нулю не гарантирует немедленного удаления, поскольку время выполнения компонента жизненного цикла определяется расписанием.

Пользователь может хранить данные в этой таблице, изменив значение поля RETENTION\_IN\_MINUTES в WBIRMADM.RMPRUNCTRL на желаемую продолжительность в минутах.

— Планирование сокращения рабочих таблиц ETL.

Планирование компонента жизненного цикла приемника подобно планированию компонента ETL. После завершения цикла применения изменений и завершения всех связанных экземпляров компонента ETL последовательно вызываются экземпляры компонента жизненного цикла приемника. Вызов приводит к выполнению, если включено сокращение рабочей таблицы, и если выполнение



запланировано. В приведенном выше примере экземпляр компонента жизненного цикла приемника выполняется только один раз - в момент времени T3. Есть несколько причин, по которым в моменты T6 и T9 не происходит выполнения:

- Конфигурация экземпляра компонента жизненного цикла приемника была изменена между моментами T4 и T6 и сокращение выключено.
- Время, прошедшее между моментами T3 и T9, меньше интервала, указанного для этого экземпляра компонента жизненного цикла приемника.

**Примечание:** В этом примере, расписания для экземпляра компонента ETL и экземпляра компонента жизненного цикла приемника различны (предполагая, что сокращение не выключено).

В общем, либо используйте одинаковое расписание для всех соответствующих экземпляров, либо используйте расписание с более длительными циклами для экземпляров жизненного цикла. Причина этого состоит в том, что данные не могут быть сокращены, если они не еще были успешно обработаны экземплярами компонента ETL. Для изменения расписания, установленного по

умолчанию, найдите соответствующие записи в WBIRMADM.RMPRUNECTRL и измените значение поля PRUNE\_INTERVAL, в котором указана минимальная задержка между выполнениями в минутах: Увеличение интервала сокращения приведет к меньшей частоте выполнений, но число вызовов

TABLE_NAME	LAST PRUNED	PRUNE_INTERVAL
APP.CCD_6	Oct 11, 2005 4:40:20 PM ...	1440
APP.CCD_7	Oct 11, 2005 4:40:20 PM ...	1440

останется прежним. При каждом выполнении определяется, какие записи рабочей таблицы подлежат удалению, и они удаляются. Рабочие таблицы APP.CCD\_6 и APP.CCD\_7 будут сокращаться приблизительно один раз в день (1440 минут), если используется описанная выше конфигурация. Последний раз сокращение было выполнено 11 октября в 16:40, и оно будет выполнено снова 12 октября около 16:40.

### Обзор параметров конфигурации служб перемещения данных

В следующей таблице перечислены наиболее часто используемые параметры, предусмотренные для всех компонентов службы перемещения данных. Дополнительная информация о параметрах конфигурации приведена в руководстве по репликации DB2.

Компонент	Имя параметра	Значения по умолчанию	Допустимые значения	Расположение параметра
Сбора данных	autoprune	Y		
Сбора данных	prune_interval (секунд)	300		
Жизненного цикла источника	PRUNE_ENABLED	1	0 - Выключен 1 - Включен	Служба перемещения данных Исходная DB: WBIRMADM.RMPRUNECTRL
Жизненного цикла источника	RETENTION_IN_MINUTES	0 - Из базы данных состояний в рабочую  1440 - Из рабочей базы данных в базу данных хронологии	От 0 до предельного значения BIGINT, установленного в DB2	Служба перемещения данных Исходная DB: WBIRMADM.RMPRUNECTRL
Жизненного цикла источника	PRUNE_INTERVAL (минут)	5	От 0 до предельного значения BIGINT, установленного в DB2	Служба перемещения данных Исходная DB: WBIRMADM.RMPRUNECTRL
ETL	ETLSCHEDMETHOD	1	0 - Гибкое расписание  1 - Расписание с фиксированными интервалами  Другое - ETL выключен	Служба перемещения данных Целевая DB: WBIRMADM.RMCONTROL
ETL	ETL_0_MINUTES	5 - Из базы данных состояний в рабочую  1440 - Из рабочей базы данных в базу данных хронологии	От 0 до предельного значения INTEGER, установленного в DB2	Служба перемещения данных Целевая DB: WBIRMADM.RMCONTROL
ETL	LOGLEVEL	0	0 - Для нормального протокола  1 - Для протокола трассировки	Служба перемещения данных Целевая DB: WBIRMADM.RMCONTROL
ETL	COMMITINTERVAL (количество записей)	1000	0 - Выключить фиксации до завершения  1 - Фиксировать каждую запись  n - предельное значение BIGINT, установленное в DB2	Служба перемещения данных Целевая DB: WBIRMADM.RMCONTROL

В следующей таблице перечислены наиболее часто используемые параметры, предусмотренные для всех компонентов службы перемещения данных. Дополнительная информация о параметрах конфигурации приведена в руководстве по репликации DB2.

Компонент	Имя параметра	Значения по умолчанию	Допустимые значения	Расположение параметра
Жизненного цикла приемника	PRUNE_ENABLED	1	0 - Выключен 1 - Включен	Служба перемещения данных Целевая DB: WBIRMADM.RMPRUNECTRL
Жизненного цикла приемника	RETENTION_IN_MINUTES	0	От 0 до предельного значения BIGINT, установленного в DB2	Служба перемещения данных Целевая DB: WBIRMADM.RMPRUNECTRL
Жизненного цикла приемника	PRUNE_INTERVAL (минут)	1440	От 0 до предельного значения BIGINT, установленного в DB2	Служба перемещения данных Целевая DB: WBIRMADM.RMPRUNECTRL

**Примечание:** IBM оставляет за собой право вносить изменения в таблицы базы данных и поля, указанные ниже. Поэтому некоторые таблицы и поля могут быть изменены, удалены или добавлены в новом выпуске. Пользователь подвергает себя риску, ожидая увидеть в следующем выпуске те же содержание и структуру таблиц, которые описаны здесь. IBM документирует все подобные изменения по мере их появления.

## Управление изменениями и создание артефактов

Модель бизнес-величин содержит множество определений событий и процессов. На основе этих определений генератор схемы создает соответствующие артефакты, необходимые для создания таблиц базы данных, определений Cube Views и сценариев репликации. Изменения в модели бизнес-величин вызывают изменения в генерируемых артефактах.

Если такое изменение происходит, необходимо перезапустить генератор схемы для создания новых сценариев модели бизнес-величин. Это называется управлением изменениями.

Управление изменениями необходимо в следующих случаях:

- Добавлен новый процесс, что привело к появлению новой таблицы.
- Добавлен новый показатель, не являющийся частью размерности или созданный в новой размерности, и в соответствующие таблицы процессов добавлен новый столбец.
- Изменена длина показателя типа строка, что привело к изменению длины соответствующего столбца.

Любые изменения в модели бизнес-величин требуют повторного выполнения следующих действий:

- Импорт измененной модели бизнес-величин в генератор схемы административной консоли WebSphere Business Monitor для создания соответствующих артефактов.
- Запустите вновь созданные сценарии на языке определения данных (DDL) для внесения изменений в базу данных.
- Разверните сценарии репликации для синхронизации базы данных состояний, рабочей базы данных и базы данных хронологии после внесения изменений.
- Развертывание вновь созданных определений Cube Views.
- Развертывание модели бизнес-величин в административной консоли WebSphere Business Monitor.

Генератор схемы проверяет предыдущую и новую версии модели бизнес-величин. Если эта модель не была развернута или не существует в хранилище, то генератор схемы создает артефакты, которые соответствуют новой модели. Если предыдущая версия модели существует, генератор схемы проверяет различия между развернутой и новой версиями модели. При обнаружении различий создаются соответствующие сценарии для изменения баз данных на основе этих различий. Дополнительную информацию о создании новой модели из существующей можно получить в документации WebSphere Business Modeler.

Некоторые изменения модели бизнес-величин недопустимы из-за ограничений на возможности изменения таблиц базы данных существующей модели бизнес-величин. В случае подобных изменений необходимо

повторно сгенерировать и развернуть всю модель. При этом весь набор артефактов генерируется и развертывается заново. Эти изменения перечислены ниже:

- Изменение атрибута **Использовать в WebSphere Business Monitor** бизнес-величины, например, изменение значения показателя с **Динамические данные о текущем процессе** на **Количественные данные в размерном анализе**.
- Изменение группы размерности, к которой принадлежит показатель.
- Изменение состояния переключателя **isPartOfDimensionKey** для показателя размерности в редакторе бизнес-величин.
- Изменение типа данных показателя. Изменить тип данных показателя можно, удалив текущий показатель и создав новый.
- Удаление показателя, являющегося частью ключа размерности.
- Создание нового показателя в качестве части ключа размерности для существующей размерности.
- Изменение таймера с накоплением на таймер без накопления или наоборот.
- Изменение модели процесса или группы действий.

**Примечание:** Удаление какого-либо процесса не требует повторной генерации модели, хотя и может привести к удалению показателя. Достаточно будет сгенерировать только изменения.

В соответствии с изменениями в редакторе бизнес-величин предусмотрено три сценария развертывания:

- Развертывание новой модели
- Развертывание новой версии модели
- Развертывание другой модели

## Добавлен новый процесс

Когда вы добавляете новый процесс в модель бизнес-величин, в рабочую базу данных и в базы данных состояний и хронологии добавляется новая таблица.

После добавления нового процесса следует с помощью генератора схемы сгенерировать соответствующие сценарии, необходимые для изменения созданных баз данных (рабочей базы данных, баз данных состояний и хронологии) и параметров репликации.

**Примечание:** Важно учесть это изменение при выполнении планирования. Дополнительная информация о планировании базы данных приведена в разделе “Подготовка к развертыванию артефактов баз данных” на стр. 5.

Для того чтобы синхронизировать репликации между базами данных, необходимо добавить в конфигурацию репликации новые таблицы, соответствующие недавно добавленным процессам. Необходимо развернуть эти сценарии, для того чтобы добавить новые таблицы в базы данных WebSphere Business Monitor и произвести соответствующие изменения в параметрах репликации.

При добавлении нового процесса необходимо выполнить следующее:

- Создайте резервные копии всех баз данных до развертывания какой-либо новой или обновленной модели бизнес-величин.
- С помощью генератора схемы сгенерируйте сценарии на языке определения данных (DDL), которые должны быть развернуты для добавления новых таблиц и изменения существующих баз данных.
- Разверните новые сценарии репликации в командном окне DB2 для включения репликации новых таблиц процессов.
- Повторно импортируйте модель куба в DB2 Cube Views и ALPHABLOX, так чтобы эта модель отражала все новые кубы, созданные в базе данных хронологии.

## В существующей группе бизнес-величин имеется новое поле

Добавление новой бизнес-величины (показателя) к существующей группе приводит к добавлению нового поля в некоторых таблицах экземпляров в базе данных состояний, рабочей базе данных и базе данных хронологии.

Новые поля добавляются в таблицы в соответствии с наличием атрибута **Использовать в WebSphere Business Monitor** у добавляемого показателя. Влияние на базы данных показано в следующей таблице:

*Применение показателя в базах данных WebSphere Business Monitor*

Применение в WebSphere Business Monitor	База данных состояний	Рабочая база данных	Базы данных хронологии
Временное вычисление	Да	Нет	Нет
Оперативные данные о выполняющихся процессах	Да	Да	Нет
Количественные данные в размерном анализе	Да	Да	Да
Группы агрегирования в размерном анализе	Да	Да	Да

После добавления новой бизнес-величины (показателя) в редакторе бизнес-величин следует с помощью генератора схемы сгенерировать соответствующие сценарии, необходимые для изменения созданных баз данных (рабочей базы данных, баз данных состояний и хронологии) и параметров репликации. Для изменения баз данных разверните сценарии на языке определения данных (DDL) в командном окне DB2 для добавления новых полей в таблицы баз данных.

Для того чтобы синхронизировать репликации между базами данных, необходимо добавить в DB2 Replication Center новые поля, соответствующие недавно добавленным показателям, развернув сценарии репликации в командном окне DB2.

При добавлении нового показателя к процессу необходимо выполнить следующее:

- Создайте резервные копии всех баз данных до развертывания какой-либо новой или обновленной модели бизнес-величин.
- Остановить службу монитора для этого процесса.
- Остановить службу репликации для этого процесса.
- С помощью генератора схемы сгенерируйте сценарии DDL, которые должны быть развернуты для добавления новых полей и изменения существующих баз данных.
- Разверните созданные сценарии репликации, которые отразят внесенные в топологию базы данных изменения.
- Повторно импортируйте модель куба в DB2 Cube Views и ALPHABLOX, так чтобы этот куб отражал все новые размерности, созданные в базе данных хронологии

## Обслуживание баз данных

Базы данных WebSphere Business Monitor требуют регулярного обслуживания. Некоторые средства DB2 могут быть полезными для улучшения использования и быстродействия баз данных.

Рекомендуется использовать следующие средства:

- Советник конфигурации.
- DB2 Web Health Center, который предупреждает администраторов баз данных (DBA) о потенциальных неполадках и дает рекомендации по их устранению. Администраторы баз данных (DBA) могут удаленно отслеживать экземпляр с помощью Web Health Center и видеть подробности предупреждений и рекомендации.
- Команды REORG и REORGCHK. REORG исключает избыточные записи и очищает память от удаленных записей таблиц и индексов. Это средство полезно применять в том случае, если выполняется большое

число удалений, обновлений либо вставок. REORGCHK обновляет статистику, которая используется средством оптимизации DB2. Это средство рекомендуется применять в том случае, если статистическая информация таблиц базы данных не обновлена вследствие обновлений базы данных.

- Команда RUNSTATS. Собирает статистическую информацию об объектах баз данных. Эту статистику можно применять при извлечении данных для выбора пути доступа к данным. Следовательно, DB2 будет обладать информацией, необходимой для выбора наиболее эффективного пути доступа. Быстродействие баз данных будет низким до тех пор, пока администратор не запустит команду DBA RUNSTATS для всех таблиц во всех базах данных после того, как был разрешен сбор данных в этих базах данных. Пример:  
RUNSTATS ON TABLE tablename WITH DISTRIBUTION AND DETAILED INDEXES ALL

**Примечание:** имя\_таблицы должно быть задано полностью вместе с именем схемы.

После выполнения этой команды. Выполните команду повторного связывания DB2:

```
db2rbind <псевдоним-базы-данных> -l logfile all
```

Выполнение затрагивает быстродействие обработки событий сервера монитора, а также быстродействие репликации. Кроме этого, интервал сокращения по умолчанию, установленный в таблице репликации RMPRUNECTL, должен быть равен 0 вместо текущего значения 1440 секунд (или 24 часа).

Более подробная информация об утилитах обслуживания DB2 находится в документации по IBM DB2.

Резервное копирование баз данных и их восстановление после ошибок развертывания также входит в обслуживание баз данных.

## Резервное копирование базы данных

Следует выполнить резервное копирование рабочей базы данных, баз данных состояний, хранилища и хронологии перед выполнением любых новых сценариев на языке определения данных (DDL) WebSphere Business Monitor независимо от того, развертывают ли они новую модель бизнес-величин или обновляют существующую.

Резервное копирование гарантирует, что в случае какой-нибудь неполадки будет возможен безопасный откат. Если вновь собранные данные не представляют интереса, то базу данных можно будет вернуть к предыдущему состоянию, в котором она не содержит недавно развернутой модели бизнес-величин.

Дополнительная информация о резервном копировании и восстановлении базы данных приведена в разделе Восстановление данных DB2.

## Восстановление после ошибок развертывания

Если в процессе развертывания артефактов репликации обнаружена ошибка. Следует отменить действия, выполненные в процессе развертывания отдельной модели бизнес-величин, чтобы отменить все изменения.

Каждое развертывание выполняется в несколько этапов. Далее описан типичный сценарий:

1. Развертывание DDL
  - a. Разверните state.ddl.
  - b. Разверните runtime.ddl.
  - c. Разверните datamart.ddl.
2. Развертывание служб перемещения данных
  - a. Разверните Источник\_установки\_службы\_Из\_базы\_данных\_состояний\_в\_рабочую\_базу\_данных.
  - b. Разверните Приемник\_установки\_службы\_Из\_базы\_данных\_состояний\_в\_рабочую\_базу\_данных.
  - c. Разверните  
Источник\_установки\_службы\_Из\_рабочей\_базы\_данных\_состояний\_в\_базу\_данных\_хронологии.
  - d. Разверните  
Приемник\_установки\_службы\_Из\_рабочей\_базы\_данных\_состояний\_в\_базу\_данных\_хронологии.

Для определения необходимого действия следует определить, в какой точке произошел сбой. Например, если произошел сбой state.ddl, тогда для возврата в исходного состояние достаточно простого отката



транзакции. Однако в случае сбоя `datamart.dll` откат `datamart.dll` вернет систему только к моменту после успешного выполнения `runtime.ddl`. Восстановление после сбоев в середине процесса развертывания служб перемещения данных выполняется сложнее всего, однако не является невозможным. Легче всего выполнить восстановление из первых развертываний, далее следуют развертывания новых моделей и, наконец, развертывания измененных моделей - самый сложный путь восстановления.

Восстановление после ошибок развертывания сценариев репликации включает в себя: определение, резервное копирование, восстановление или удаление и повторное развертывание.

### Определение

- Определите, какие произошли ошибки, и следует ли обращаться в службу поддержки IBM Support.
- Определите модель бизнес-величин, которая развертывалась на момент возникновения ошибки.
- Определите таблицы, которые создавались или изменялись генератором схемы при возникновении ошибки.
- Определите артефакты, которые создавались или изменялись генератором схемы при возникновении ошибки.
- Определите последнюю верную версию модели бизнес-величин. в базе данных хранилища.
- Определите, в случае развертывания управления изменениями, расположение артефактов, которые были развернуты для предыдущих версий модели. Это обеспечит структуры базы данных, их описания и взаимосвязи. Это может оказаться важным в том случае, если требуется выполнить резервное копирование данных и их восстановление в дальнейшем.
- Определите расположение текущих артефактов и файлов протоколов развертывания. Эта информация важна для определения неполадок и, возможно, потребуется для предоставления в службу поддержки IBM Support.
- Определите, в случае развертывания управления изменениями, имеются ли еще не обработанные данные в какой-либо из таблиц CCD. Для определения таблиц CCD (*TGT\_RM\_APP\_STG\_TAB\_NAME*) с именем проекта модели бизнес-величин (*OM\_NAME*) можно использовать развернутую таблицу *WBIRMDM.RMMETADATA* (доступна в рабочей базе данных и базе данных хронологии). Строки, помеченные **I** или **U**, могут еще остаться необработанными, и следует выполнить их резервное копирование. В поле *SERVICE\_NAME* содержится расположение таблицы CCD и целевой таблицы. На это указывает имя после слова *to*. Следует сохранить дорожку взаимосвязей к *TGT\_TAB\_NAME* в том случае, если принято решение о полном удалении всех артефактов и о создании совершенно нового набора. Это необходимо, генератор схемы не может сгенерировать те же имена для таблиц CCD, и потребуется восстановление этих данных в новые таблицы CCD после успешного развертывания.

### Резервное копирование

- На основании полученной на этапе определения информации установите, требуется ли сохранять какие-нибудь данные. Поскольку другие модели бизнес-величин могут быть запущены в то же самое время, когда выполняется развертывание, может потребоваться создать резервные копии объектов базы данных, связанных с другими моделями бизнес-величин.
- Может потребоваться создать резервные копии таблиц CCD (согласованное изменение данных). В этих таблицах генератор схемы может хранить информацию, которая еще отсутствует в таблицах базы данных источника и приемника.

**Примечание:** Завершенные события могут существовать только в базе данных хронологии.

### Восстановление или удаление

- Определите, будет ли проще восстановить предыдущую базу данных или удалить артефакты вручную.
- **Восстановление** Восстановление из резервной копии является хорошим решением в том случае, если не существует других моделей бизнес-величин либо когда другие модели бизнес-величин не активны. Восстановите предыдущий набор баз данных и для каждой базы данных повторно присвойте соответствующие приложения, а также повторно зарегистрируйте все основанные на Java хранимые процедуры и пользовательские функции.

**Примечание:**

- Дополнительная информация о резервном копировании и восстановлении базы данных приведена в разделе Восстановление данных в документации DB2.
- Для завершенных развернутых моделей в таблице *WBIRMADM.RMMETADATA* представлена информация о том, что не следует удалять. Однако при выполнении развертывания, для того чтобы определить некоторые артефакты и взаимосвязи, может потребоваться изучить протоколы развертывания для определения того, что можно безопасно удалить.

**• Удаление**

- Репликация: база данных хронологии и рабочая база данных
  - Остановите все серверы сбора данных, связанные с этой моделью бизнес-величин. (Серверы сбора данных на самом деле выполняются в рабочей базе данных и базе данных состояний.)
  - Остановите все серверы применения изменений, связанные с этой моделью бизнес-величин.
  - Удалите все хранимые процедуры ETL для модели бизнес-величин.
  - Удалите все промежуточные таблицы ETL, которые используются в модели бизнес-величин.
  - Удалите всю управляющую информацию ETL из таблицы *WBIRMADM.RMCONTROL* в соответствующей целевой базе данных для этой модели бизнес-величин.
  - Удалите все хранимые процедуры ETL сокращения и триггеры, используемые в модели бизнес-величин.
  - Удалите все таблицы, перечисленные в поле *TGT\_RM\_APP\_STG\_TAB\_NAME* таблицы *WBIRMADM.RMMETADATA*, имеющие суффикс *\_BKUP* и *\_M*, а также имеющие соответствующее *Имя\_службыRuntime\_to\_Historical* для хронологии и *State\_to\_Runtime* для рабочей базы данных этой модели бизнес-величин. Оставьте таблицу в *TGT\_RM\_APP\_STG\_TAB\_NAME*, поскольку она будет удалена позже.
  - С помощью DB2 Replication Center, удалите всех участников набора подписки применения изменений, обслуживающих эту модель бизнес-величин.
    - Если набор подписки применения изменений пуст, удалите его.
    - Если сервер применения изменений не имеет наборов подписки, удалите этот сервер.
  - Удалите из таблицы *WBIRMADM.RMMETADATA* все записи метаданных, связанные с моделью бизнес-величин. Также потребуется удалить эти же записи из таблицы *WBIRMADM.RMMETADATA* в рабочей базе данных при обработке базы данных хронологии, и из базы данных состояния при обработке рабочей базы данных. Следует удалить только строки для модели бизнес-величин и строки в имени службы *Runtime\_to\_Historical* при обработке базы данных хронологии и *State\_to\_Runtime* для рабочей базы данных.
- Репликация: база данных состояний и рабочая база данных
  - Остановите все сервера сбора данных, обслуживающие модель бизнес-величин.
  - Удалите все триггеры, связанные с таблицами CD сбора данных для модели бизнес-величин.
  - Удалите информацию об управлении сокращением таблиц из таблицы *WBIRMADM.RMPRUNECTRL* для триггеров сокращения таблиц, которые применяются для модели бизнес-величин.
  - С помощью DB2 Replication Center удалите все подписки для всех таблиц, связанных с моделью бизнес-величин.
  - Удалите из таблицы *WBIRMADM.RMMETADATA* все записи метаданных, связанные с моделью бизнес-величин.
- Схема базы данных: В основном, в случае возникновения ошибки при генерации схемы откат выполняется до начала развертывания измененной модели. Текущий набор артефактов репликации это не затрагивает.

**Повторное развертывание**



Когда все артефакты, поддерживающие модель бизнес-величин, будут удалены, можно будет снова запустить генератор схемы, выбрав опцию **Игнорировать предыдущие развертывания**. Если схема создана успешно, то не разворачивайте сценарии DDL, а просто опять разверните сценарии репликации.

## Создание и настройка баз данных

Создание и настройка баз данных WebSphere Business Monitor является важнейшей задачей в процессе установки.

В WebSphere Business Monitor имеется четыре базы данных:

- Хранилище
- Состояние
- Среда выполнения
- Хронология

Во время установки WebSphere Business Monitor базы данных могут быть созданы только с помощью панели запуска. Если базы данных были удалены после установки, можно создать их повторно с помощью панели запуска либо вручную. Создание баз данных включает в себя создание статических таблиц, табличных пространств, индексов и соответствующую настройку базы данных. База данных состояний, рабочая база данных и база данных хронологии содержат как статические, так и динамические таблицы, в то время как база данных хранилища содержит только статические таблицы. Сценарии, необходимые для создания баз данных, расположены в каталоге `<каталог_установки_монитора\install\mondb`.

Для создания баз данных вручную выполните следующие действия:

### 1. В системе Windows:

- Перейдите к **Пуск -> Программы -> IBM DB2 -> Command Line Tools -> Command Window**.
- Для создания базы данных хранилища выполните следующий сценарий: `db2CreateRepository.bat <%имя_базы_данных_хранилища%> <%ИД_пользователя_DB2%> <DB2Password> Create_Repository.sql <%каталог_установки%>`.
- Для создания базы данных состояний выполните следующий сценарий: `db2CreateState.bat createStateDB.ddl <%каталог_установки%>`.
- Для создания рабочей базы данных выполните следующий сценарий: `db2CreateRuntime.bat createRuntimeDB.ddl <%каталог_установки%>`.
- Для создания базы данных хронологии выполните следующий сценарий: `db2CreateHistorical.bat createDatamartDB.ddl <%каталог_установки%>`.

**Примечание:** `<%каталог_установки%>` по умолчанию: "C:\IBM\Websphere\Monitor".

### 2. В системе AIX:

- Войдите в систему как пользователь экземпляра DB2.
- Для создания базы данных хранилища выполните следующий сценарий: `db2CreateRepository.sh <%имя_базы_данных_хранилища%> <%ИД_пользователя_DB2%> <DB2Password> <%путь_к_DDL%>/Create_Repository.sql <%каталог_установки%>`.
- Для создания базы данных состояний выполните следующий сценарий: `db2CreateState.sh <%путь_к_DDL%>/createStateDB.ddl <%каталог_установки%> <%ИД_пользователя_DB2%>`.
- Для создания рабочей базы данных выполните следующий сценарий: `db2CreateRuntime.sh <%путь_к_DDL%>/createRuntimeDB.ddl <%каталог_установки%> <%ИД_пользователя_DB2%>`.
- Для создания базы данных хронологии выполните следующий сценарий: `db2CreateHistorical.sh <%путь_к_DDL%>/createDatamartDB.ddl <%каталог_установки%> <%ИД_пользователя_DB2%>`.

**Примечание:**

- В случае установки в другой каталог следует изменить соответствующие пути на основании каталога установки.

- `<%Путь_к_DDL%>` по умолчанию: `/opt/IBM/WebSphere/Monitor/Install/mondb.`
- `<%каталог_установки%>` по умолчанию: `/opt/IBM/WebSphere/Monitor.`
- `<%ИД_пользователя_DB2%>` по умолчанию: `db2inst1.`

## Управление базами данных во время выполнения

Управление базами данных WebSphere Business Monitor во время выполнения включает в себя развертывание созданных генератором схемы артефактов в административной консоли WebSphere Business Monitor. Развертывание этих артефактов повторяется при каждом импорте новой или измененной модели бизнес-величин.

Управление базами данных во время выполнения состоит из следующих задач.

### Создание динамических таблиц базы данных

Динамические таблицы соответствуют заданной модели бизнес-величин. Необходимые для создания этих таблиц сценарии создаются генератором схемы.

Для создания этих сценариев необходимо выполнить генератор схемы в административной консоли WebSphere Business Monitor.

Сценарии создают таблицы и индексы и устанавливают параметры конфигурации для каждой динамической таблицы в базе данных состояний, рабочей базе данных и базе данных хронологии. Расположение созданных сценариев задается во время конфигурации генератора схемы в административной консоли WebSphere Business Monitor.

#### База данных состояний:

Для создания динамических таблиц в базе данных состояний необходимо выполнить следующие действия. Сценарии расположены в определенном пользователем расположении. Это расположение определяется при установке конфигурации генератора схемы в административной консоли WebSphere Business Monitor.

Сценарий на языке определения данных (DDL), необходимый для создания таблиц в рабочей базе данных, хранится в файле **state.ddl** в корневом каталоге. Для развертывания сценариев выполните следующие действия:

1. Откройте **DB2 Command Window**. В системе UNIX, если настроена среда оболочки, можно вызвать процессор командной строки DB2.
2. Измените путь к расположению файла сценария.
3. Создайте резервную копию базы данных состояний перед развертыванием новой модели бизнес-величин.
4. Выполните команду **db2 terminate**. Это позволяет убедиться, что все предыдущие фоновые процессы, для которых могло использоваться другое значение кодовой страницы, не будут применяться, и для обработки данного запроса будет создан новый фоновый процесс.
5. Установите для переменной среды **DB2CODEPAGE** значение 1208. Процессор командной строки DB2 по умолчанию будет интерпретировать все символьные данные, используя текущую кодовую страницу. Созданный файл *state.ddl*, однако, содержит символы UTF-8, которые будут повреждены, если для переменной среды DB2CODEPAGE не будет задано значение 1208.
  - a. В операционной системе UNIX.
    - С помощью оболочек типа *sh*, *ksh*, *bash* выполните команду **export DB2CODEPAGE=1208**.
    - С помощью оболочек типа *csh*, *tsch* выполните команду **setenv DB2CODEPAGE 1208**.
  - b. В операционной системе Windows выполните команду **set DB2CODEPAGE=1208**.
6. Подключитесь к базе данных состояний, выполнив следующую команду: **db2 connect to <имя\_базы\_данных\_состояний>**.
7. Выполните команду **db2 +c "stvf state.ddl > state.log"**. Это запустит сценарий и сохранит файл протокола, содержащий транзакции, для возможности устранения неполадок. Проверьте созданный файл протокола

на наличие ошибок перед тем как фиксировать изменения. При необходимости отката выполните команду **db2 rollback**, которая отменит все действия. Если не произошло ошибок, выполните команду **db2 commit** для фиксации изменений.

8. Отсоединитесь от базы данных состояний после выполнения сценария с помощью команды **db2 disconnect <имя\_базы\_данных\_состояний>**.
9. Выполните команду **db2 terminate** для завершения фоновых процессов.

#### Рабочая база данных:

Для создания динамических таблиц в рабочей базе данных необходимо выполнить следующие действия. Сценарии расположены в определенном пользователем расположении. Это расположение определяется при установке конфигурации генератора схемы в административной консоли WebSphere Business Monitor.

Сценарии на языке определения данных (DDL), необходимые для создания таблиц в рабочей базе данных, хранятся в файле **runtime.ddl** в корневом каталоге. Для развертывания сценариев выполните следующие действия:

1. Откройте **DB2 Command Window**. В системе UNIX, если настроена среда оболочки, можно вызвать процессор командной строки DB2.
2. Измените путь к расположению файла сценария.
3. Создайте резервную копию рабочей базы данных перед развертыванием новой модели бизнес-величин.
4. Выполните команду **db2 terminate**. Это позволяет убедиться, что все предыдущие фоновые процессы, для которых могло использоваться другое значение кодовой страницы, не будут применяться, и для обработки данного запроса будет создан новый фоновый процесс.
5. Установите для переменной среды **DB2CODEPAGE** значение 1208. Процессор командной строки DB2 по умолчанию будет интерпретировать все символьные данные, используя текущую кодовую страницу. Созданный файл **runtime.ddl**, однако, содержит символы UTF-8, которые будут повреждены, если для переменной среды DB2CODEPAGE не будет задано значение 1208.
  - a. В операционной системе UNIX.
    - С помощью оболочек типа *sh*, *ksh*, *bash* выполните команду **export DB2CODEPAGE=1208**.
    - С помощью оболочек типа *csh*, *tsch* выполните команду **setenv DB2CODEPAGE 1208**.
  - b. В операционной системе Windows выполните команду **set DB2CODEPAGE=1208**.
6. Подключитесь к рабочей базе данных, выполнив следующую команду: **db2 connect to <имя\_рабочей\_базы\_данных>**.
7. Выполните команду **db2 +c -stv runtime.ddl > runtime.log**. Это запустит сценарий и сохранит файл протокола, содержащий транзакции, для возможности устранения неполадок. Проверьте созданный файл протокола на наличие ошибок перед тем как фиксировать изменения. При необходимости отката выполните команду **db2 rollback**, которая отменит все действия. Если не произошло ошибок, выполните команду **db2 commit** для фиксации изменений.
8. Отсоединитесь от рабочей базы данных после выполнения сценария с помощью команды **db2 disconnect <имя\_рабочей\_базы\_данных>**.
9. Выполните команду **db2 terminate** для завершения фоновых процессов.

#### Базы данных хронологии:

Для создания динамических таблиц в базе данных хронологии необходимо выполнить следующие действия. Сценарии расположены в определенном пользователем расположении. Это расположение определяется при установке конфигурации генератора схемы в административной консоли WebSphere Business Monitor.

Сценарии на языке определения данных (DDL), необходимые для создания таблиц в базе данных хронологии, хранятся в файле **datamart.ddl** в корневом каталоге. Для развертывания сценариев выполните следующие действия:

1. Откройте **DB2 Command Window**. В системе UNIX, если настроена среда оболочки, можно вызвать процессор командной строки DB2.
2. Измените путь к расположению файла сценария.
3. Создайте резервную копию базы данных хронологии перед развертыванием новой модели бизнес-величин.
4. Выполните команду **db2 terminate**. Это позволяет убедиться, что все предыдущие фоновые процессы, для которых могло использоваться другое значение кодовой страницы, не будут применяться, и для обработки данного запроса будет создан новый фоновый процесс.
5. Установите для переменной среды **DB2CODEPAGE** значение 1208. Процессор командной строки DB2 по умолчанию будет интерпретировать все символьные данные, используя текущую кодовую страницу. Созданный файл *datamart.ddl*, однако, содержит символы UTF-8, которые будут повреждены, если для переменной среды DB2CODEPAGE не будет задано значение 1208.
  - a. В операционной системе UNIX.
    - С помощью оболочек типа *sh*, *ksh*, *bash* выполните команду **export DB2CODEPAGE=1208**.
    - С помощью оболочек типа *csh*, *tsch* выполните команду **setenv DB2CODEPAGE 1208**.
  - b. В операционной системе Windows выполните команду **set DB2CODEPAGE=1208**.
6. Подключитесь к базе данных хронологии, выполнив следующую команду: **db2 connect to <имя\_базы\_данных\_хронологии>**. Этот сценарий будет выполнен без автоматической фиксации изменений.
7. Выполните команду **db2 +c -stv datamart.ddl > datamart.log**. Это запустит сценарий и сохранит файл протокола, содержащий транзакции, для возможности устранения неполадок. Проверьте созданный файл протокола на наличие ошибок перед тем как фиксировать изменения. При необходимости отката выполните команду: **db2 rollback**, которая отменит все действия. Если не произошло ошибок, выполните команду **db2 commit** для фиксации изменений.
8. Отсоединитесь от базы данных хронологии после выполнения сценария с помощью команды **db2 disconnect <имя\_базы\_данных\_хронологии>**.
9. Выполните команду **db2 terminate** для завершения фоновых процессов.

**Примечание:** В некоторых случаях, когда *datamart.ddl* выполняется для новой версии существующей модели бизнес-величин, могут возникать ошибки, подобные следующей: SQL0605W Индекс не был создан, поскольку в требуемом описании индекс "WBI.I\_1133789461307" уже существует. SQLSTATE=01550. Эти ошибки можно игнорировать и зафиксировать транзакцию, если не возникли другие ошибки.

## Развертывание служб перемещения данных

Перед развертыванием службы перемещения данных из базы данных состояний в рабочую базу данных и из рабочей базы данных в базу данных хронологии должны быть созданы динамические таблицы баз данных. Любые ошибки, возникшие при выполнении сценариев создания динамических таблиц баз данных, приведут к неполадкам во время развертывания службы перемещения данных.

В процессе генерации схемы будет создано до трех сжатых (ZIP-файлы или JAR-файлы) архивов с именами (DS\_State\_setup, DS\_Runtime\_setup и DS\_Datamart\_setup), содержащих файлы установки службы перемещения данных. Эти три архива всегда создаются при первом выполнении генерации схемы для модели бизнес-величин. В ходе последующих генераций, например, после изменения модели бизнес-величин, новые архивы могут не создаваться, либо может быть создано один, два или три архива. Архив будет создан только в том случае, если для согласования изменений в модели бизнес-величин необходимо внести изменения в существующую среду репликации. Архивы службы перемещения данных расположены в каталоге, который был указан в конфигурации административной консоли генератора схемы во вкладке "Общее".

Развертывание службы перемещения данных включает в себя создание и настройку объектов как базы данных источника (из которой перемещаются данные), так и базы данных приемника (в которую перемещаются данные).

- *DS\_State\_setup* содержит сценарий развертывания для установки базы данных источника службы перемещения данных из базы данных состояний в рабочую базу данных.
  - *DS\_Runtime\_setup* содержит сценарий развертывания для установки базы данных приемника службы перемещения данных из базы данных состояний в рабочую базу данных и сценарий развертывания для установки базы данных приемника службы перемещения данных из рабочей базы данных в базу данных хронологии.
  - *DS\_Datamart\_setup* содержит сценарий развертывания для установки базы данных приемника службы перемещения данных из рабочей базы данных в базу данных хронологии.
1. Разверните службу перемещения данных из базы данных состояний в рабочую базу данных:
    - a. Определите, в какой системе будут развернуты исходные артефакты для службы перемещения данных из базы данных состояний в рабочую базу данных. В большинстве случаев - это система, являющаяся хостом базы данных состояний.
    - b. В этой системе создайте рабочий каталог и скопируйте в него (либо передайте, если эта система является удаленной) созданный архив *DS\_State\_setup*. Необходимо выбрать путь с длиной не более 100 символов, поскольку в операционной системе существует ограничение на длину пути.
    - c. Распакуйте архив (в файле .zip в системе Windows и в файле .jar в системе UNIX) в рабочий каталог.
    - d. Во время развертывания для создания и настройки объектов баз данных будут применяться различные утилиты DB2. Среду баз данных необходимо настроить для работы с этими утилитами. Для этого в системе Microsoft Windows откройте командное окно DB2. В системе UNIX убедитесь, что заданы соответствующие переменные среды.
    - e. Перейдите в каталог, в который был извлечен архив *DS\_State\_setup*.
    - f. Выполните *State\_to\_Runtime\_setup\_source.bat* (в системе UNIX расширение .sh) и действуйте в соответствии с приглашениями. Сценарий будет показывать сообщения о состоянии, указывающие, что определенная команда: выполнена успешно, создано предупреждение либо не выполнена.
    - g. Проверьте созданный файл протокола *State\_to\_Runtime\_setup\_source.log* на наличие предупреждений или сообщений об ошибках. Не продолжайте работу при наличии ошибок.
    - h. Создайте резервную копию рабочего каталога. Служба поддержки IBM support может использовать содержимое этого каталога при устранении неполадок.
    - i. Определите, в какой системе будут развернуты целевые артефакты для службы перемещения данных из базы данных состояний в рабочую базу данных. В большинстве случаев - это система, являющаяся хостом рабочей базы данных.
    - j. В этой системе создайте рабочий каталог и скопируйте в него (либо передайте, если эта система является удаленной) созданный архив *DS\_Runtime\_setup*. Необходимо выбрать путь с длиной не более 100 символов, поскольку в операционной системе существует ограничение на длину пути.
    - k. Распакуйте архив (в файле .zip в системе Windows и в файле .jar в системе UNIX) в рабочий каталог.
    - l. Во время развертывания для создания и настройки объектов баз данных будут применяться различные утилиты DB2. Среду баз данных необходимо настроить для работы с этими утилитами. Для этого в системе Microsoft Windows откройте командное окно DB2. В системе UNIX убедитесь, что заданы соответствующие переменные среды.
    - m. Перейдите в каталог, в который был извлечен архив *DS\_Runtime\_setup*.
    - n. Выполните *State\_to\_Runtime\_setup\_target.bat* (в системе UNIX расширение .sh) и действуйте в соответствии с приглашениями. Сценарий будет показывать сообщения о состоянии, указывающие, что определенная команда выполнена успешно, создано предупреждение либо не выполнена.
    - o. Проверьте созданный файл протокола *State\_to\_Runtime\_setup\_source.log* на наличие предупреждений или сообщений об ошибках. Не продолжайте работу при наличии ошибок.
    - p. Создайте резервную копию рабочего каталога. Служба поддержки IBM support может использовать содержимое этого каталога при устранении неполадок.
    - q. Если нет сообщений о неполадках, служба перемещения данных из базы данных состояний в рабочую базу данных установлена.
  2. Разверните службу перемещения данных из рабочей базы данных в базу данных хронологии:



- a. Определите, в какой системе будут развернуты исходные артефакты для службы перемещения данных из рабочей базы данных в базу данных хронологии. В большинстве случаев - это система, являющаяся хостом рабочей базы данных. Если в этой же системе были развернуты целевые артефакты службы перемещения данных из базы данных состояний в рабочую базу данных, то можно перейти к шагу е, поскольку требуемые файлы развертывания уже были извлечены.
- b. Если развертывание выполняется не в системе, являющейся хостом рабочей базы данных, создайте рабочий каталог и скопируйте в него (либо передайте, если эта система является удаленной) созданный архив `DS_Runtime_setup`. Необходимо выбрать путь с длиной не более 100 символов, поскольку в операционной системе существует ограничение на длину пути.
- c. Распакуйте архив (в файле `.zip` в системе Windows и в файле `.jar` в системе UNIX) в рабочий каталог.
- d. Во время развертывания для создания и настройки объектов баз данных будут применяться различные утилиты DB2. Среду баз данных необходимо настроить для работы с этими утилитами. Для этого в системе Microsoft Windows откройте командное окно DB2. В системе UNIX убедитесь, что заданы соответствующие переменные среды.
- e. Перейдите в каталог, в который был извлечен архив `DS_Runtime_setup`.
- f. Выполните `Runtime_to_Historical_setup_source.bat` (в системе UNIX расширение `.sh`) и действуйте в соответствии с приглашениями. Сценарий будет показывать сообщения о состоянии, указывающие, что определенная команда выполнена успешно, создано предупреждение либо не выполнена.
- g. Проверьте созданный файл протокола `Runtime_to_Historical_setup_source.log` на наличие предупреждений или сообщений об ошибках. Не продолжайте работу при наличии сообщений об ошибках.
- h. Создайте резервную копию рабочего каталога. Служба поддержки IBM support может использовать содержимое этого каталога при устранении неполадок.
- i. Определите, в какой системе будут развернуты целевые артефакты для службы перемещения данных из рабочей базы данных в базу данных хронологии. В большинстве случаев - это система, являющаяся хостом базы данных хронологии.
- j. В этой системе создайте рабочий каталог и скопируйте в него (либо передайте, если эта система является удаленной) созданный архив `DS_Datamart_setup`. Необходимо выбрать путь с длиной не более 100 символов, поскольку в операционной системе существует ограничение на длину пути.
- k. Распакуйте архив (в файле `.zip` в системе Windows и в файле `.jar` в системе UNIX) в рабочий каталог.
- l. Во время развертывания для создания и настройки объектов баз данных будут применяться различные утилиты DB2. Среду баз данных необходимо настроить для работы с этими утилитами. Для этого в системе Microsoft Windows откройте командное окно DB2. В системе UNIX убедитесь, что заданы соответствующие переменные среды.
- m. Перейдите в каталог, в который был извлечен архив `DS_Runtime_setup`.
- n. Выполните `Runtime_to_Historical_setup_target.bat` (в системе UNIX расширение `.sh`) и действуйте в соответствии с приглашениями. Сценарий будет показывать сообщения о состоянии, указывающие, что определенная команда выполнена успешно, создано предупреждение либо не выполнена.
- o. Проверьте созданный файл протокола `State_to_Runtime_setup_source.log` на наличие предупреждений или сообщений об ошибках. Не продолжайте работу при наличии сообщений об ошибках.
- p. Создайте резервную копию рабочего каталога. Служба поддержки IBM support может использовать содержимое этого каталога при устранении неполадок.
- q. Если нет сообщений о неполадках, служба перемещения данных из рабочей базы данных в базу данных хронологии для данной модели бизнес-величин установлена.

## Настройка опций служб перемещения данных

Для каждого сервера сбора данных, созданного и настроенного компонентом служб перемещения данных, существует два параметра, которые могут повлиять на поведение компонентов сбора данных. Это параметры `lag_limit` и `startmode`.

Значения по умолчанию параметров `lag_limit` и `startmode`: "7 дней" и "WARMSI". Дополнительная информация об этих параметрах приведена в Справочнике и руководстве по репликации DB2 SQL.

Эти параметры нельзя изменить до развертывания артефактов. Однако эти параметры можно изменить до запуска серверов компонента сбора данных; можно также изменить параметры для уже запущенных серверов сбора данных.

**Примечание:** Для того чтобы изменения, выполненные во время работы серверов сбора данных, вступили в силу, следует остановить и перезапустить эти серверы.

Если при развертывании артефактов репликации применялись значения по умолчанию параметров `lag_limit` `istartmode`, и сервер сбора данных не работал дольше 7 дней, а затем был перезапущен, компонентом сбора данных будет возвращена ошибка. В сообщении об ошибке будет указано, что сервер сбора данных не может быть запущен, поскольку данные устарели. Поведение по умолчанию можно изменить несколькими способами. Далее описано три способа:

1. Измените параметры по умолчанию, как указано в таблице `<CAPTURESERVERSCHEMA>.IBMSNAP_CAPPARMS`. После развертывания артефактов репликации можно определить число серверов сбора данных, которые создаются компонентом служб перемещения данных, выполнив следующий запрос по рабочей базе данных.

```
CONNECT TO RUNTIME DATABASE
```

```
SELECT DISTINCT OM_NAME, SERVICE_NAME, SRC_RM_CAP_SVR_NAME
```

```
FROM WBIRMADM.RMMETADATA
```

```
ORDER BY 1,2,3
```

Вы увидите таблицу, подобную следующей:

Таблица 1. пример *RMMETADATA*

OM_NAME	SERVICE_NAME	SRC_RM_CAP_SVR_NAME
SubDoctor3	Из рабочей базы данных в базу данных хронологии	CAPTURE_18
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_1
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_115
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_156
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_194
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_212
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_250
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_41
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_59
SubDoctor3	Из базы данных состояний в рабочую	CAPTURE_97

`OM_NAME` - это имя проекта WebSphere Business Modeler. `SERVICE_NAME` указывает службу перемещения данных, а `SRC_RM_CAP_SVR_NAME` - это идентификатор (CAPTURE SCHEMA) сервера сбора данных, который используется как часть службы перемещения данных. В приведенной выше таблице задан один сервер сбора данных для службы перемещения данных из рабочей базы данных в базу данных хронологии и девять серверов - для службы перемещения из базы данных состояний в рабочую базу данных.

**Примечание:** Число и имена серверов различны в зависимости от применяемой модели и параметров политики, заданных в процессе генерации артефактов.

Компонент служб баз данных поддерживает все опции параметров `lag_limit` `istartmode`, однако следует иметь в виду, что быстродействие может в значительной степени снизиться при увеличении числа "холодных" запусков (запусков серверов сбора данных после сбоя). В случае частых "холодных" запусков компонент ETL службы перемещения данных обрабатывает все существующие записи, а не только изменения. После того, как все серверы сбора данных, которые необходимо изменить, опознаны, можно



изменить параметры по умолчанию в базе данных. После изменения параметров по умолчанию для каждого сервера сбора данных можно запустить эти серверы.

2. Либо измените командную строку, запускающую сервер сбора данных. В процессе создания артефактов служб баз данных создаются стандартные сценарии, которые могут запускать и останавливать службы сбора данных и применения изменений. Сценарии запуска сбора данных (StartCapture\_#.bat или StartCapture\_#.sh) расположены в каталоге <имя\_службы\_перемещения\_данных>\source. Каждый из этих сценариев содержит команду **asncap**, которая применяется для запуска программы сбора данных. Дополнительная информация об этих параметрах приведена в документации по IBM DB2. Измените сценарий запуска, сохраните его, а затем для запуска сервера сбора данных выполните этот сценарий с новыми параметрами.
3. Либо измените сервер сбора данных в процессе его работы. Следуйте инструкциям, приведенным в документации по IBM DB2, в которых описано, как временно изменить параметры работающего сервера сбора данных.

## Завершение установки служб перемещения данных

Развернутые экземпляры компонентов сбора данных и применения изменений по умолчанию используют идентификационные данные пользователя, запустившего этих экземпляров. В то время как это может оказаться важным в некоторых топологиях, существует два сценария, в которых необходимо использовать другие идентификационные данные.

- **Первый сценарий - Смена идентификационных данных пользователя:** Администратор базы данных (DBA) собирается войти в систему как *пользователь1*, но утилита перемещения данных должна подключиться к базам данных как *пользователь2*.
- **Второй сценарий - Распределенная среда:** DBA планирует выполнить утилиту в *системе1*. Исходная или целевая база данных расположена в *системе2*.

Для поддержки этих сценариев необходимо создать файлы паролей, содержащие идентификационные данные пользователя, которые должны использоваться вместо текущих. Поскольку файлы паролей не создаются автоматически при развертывании, необходимо выполнить следующие действия:

1. Подготовьте файл, в котором будут храниться *ИД пользователя* и *пароль*, используемые при подключении к исходной базе данных. В командной строке DB2 введите следующую команду и подставьте подходящие значения заменителей, помеченных как: <имя\_заменителя>.

asnpwd INIT encrypt all using <файл\_паролей>. Утилита asnpwd создает пустой файл: <файл\_паролей>.

Пример вызова: asnpwd INIT encrypt all using password.aut

2. Для каждой базы данных, к которой должна подключаться утилита репликации, сохраните информацию доступа к базе данных (ИД пользователя, пароль и имя базы данных). В командной строке DB2 введите следующую команду и подставьте подходящие значения заменителей, помеченных как: <имя\_заменителя>.

asnpwd ADD alias <имя\_DB> ID <ИД\_пользователя> PASSWORD <пароль> using <файл\_паролей>.

Повторите этот шаг для каждой базы данных, если это необходимо. Программа шифрует введенную информацию и сохраняет ее в <файле\_паролей>.

Примеры вызова:

- asnpwd ADD alias STMD7 id MYUSRID password MYPASSWRD using password.aut
- asnpwd ADD alias RTMD7 id MYUSRID2 password MYPASSWRD2 using password.aut

3. Измените файлы конфигурации утилит, откорректировав сгенерированные исполняемые сценарии запуска (StartCapture и StartApply). Добавьте параметр файла паролей в командную строку вызова утилиты репликации. Утилита использует зашифрованные идентификационные данные пользователя, которые хранятся в указанном файле, вместо идентификационных данных по умолчанию. Файл паролей должен находиться в рабочем каталоге, определенном в параметре CAPTURE\_PATH (или APPLY\_PATH).

Пример изменений:

- Исходный файл сценария запуска компонента сбора данных: db2cmd asncap CAPTURE\_SERVER=stmd7 CAPTURE\_SCHEMA=CAPTURE\_1 CAPTURE\_PATH="c:\tmp\state\_capture\_log"

- Измененный файл сценария запуска компонента сбора данных: db2cmd asncap CAPTURE\_SERVER=stmt7 CAPTURE\_SCHEMA=CAPTURE\_1 CAPTURE\_PATH="c:\tmp\state\_capture\_log" pwdfile="password.aut"
  - Исходный файл сценария запуска компонента применения изменений: db2cmd asnapply APPLY\_QUAL=Apply\_1 CONTROL\_SERVER=RTMD7 APPLY\_PATH="C:\tmp\apply"
  - Измененный файл сценария запуска компонента применения изменений: db2cmd asnapply APPLY\_QUAL=Apply\_1 CONTROL\_SERVER=RTMD7 APPLY\_PATH="C:\tmp\apply" pwdfile="password.aut"
4. Скопируйте <файл\_паролей>, созданный на первом и втором этапах, в подходящий каталог. Утилиты репликации будут при запуске пытаться открыть файл паролей. Если файл <файл\_паролей> не существует в рабочем каталоге, определенном в параметре CAPTURE\_PATH (или APPLY\_PATH), то произойдет ошибка. Если параметр рабочего каталога не задан, утилиты будут искать файл в текущем каталоге.

Дополнительная информация об утилитах DB2 приведена в справочном руководстве по репликации DB2 SQL.

## Объединение сценариев запуска и остановки

Для упрощения процессов запуска и остановки службы перемещения данных рекомендуется объединить созданные для них сценарии и вызывать их посредством основного сценария.

Поскольку компоненты сбора данных и применения изменений должны выполняться в той системе, где расположены базы данных, опции объединения меняются в зависимости от используемой топологии. Независимо от того, каким образом были объединены сценарии, следует убедиться, что не было запущено два экземпляра компонента одновременно.

Хотя можно запустить или остановить каждый экземпляр компонента сбора данных и применения изменений отдельно, более удобно объединить содержание всех сценариев запуска и остановки экземпляров компонентов в один сценарий для каждой модели бизнес-величин. Можно объединять сценарии следующим образом:

1. Определите сценарии запуска и остановки экземпляров компонента сбора данных для исходной базы данных.
2. Создайте главные сценарии запуска и остановки сбора данных, которые вызывают сценарии запуска и остановки экземпляров компонента сбора данных для исходной базы данных.
3. Определите сценарии запуска и остановки экземпляров компонента применения изменений для целевой базы данных.
4. Создайте главные сценарии запуска и остановки сбора данных, которые вызывают сценарии запуска и остановки экземпляров компонента сбора данных для целевой базы данных.

В результате этих объединений останется только четыре сценария запуска (или остановки), необходимые для запуска и остановки служб перемещения данных для модели бизнес-величин.

Если нет необходимости запускать или останавливать две службы перемещения данных отдельно, то можно произвести дальнейшее объединение. В этом случае необходимо только три сценария:

- Сценарий, запускающий (останавливающий) все экземпляры компонента сбора данных в базе данных состояний.
- Сценарий, запускающий (останавливающий) все экземпляры компонента сбора данных в рабочей базе данных.
- Сценарий, запускающий (останавливающий) все экземпляры компонента применения изменений в базе данных хронологии.

Если все три базы данных расположены в одной и той же системе, то эти три сценария можно объединить в один, который будет запускать и останавливать все экземпляры компонентов сбора данных и применения изменений.

Иногда требуется объединить сценарии запуска и остановки службы перемещения данных, созданные при других развертываниях. Во время первоначального развертывания службы перемещения данных создаются сценарии запуска и остановки для всех групп бизнес-величин. Последующие развертывания, производимые в результате внесения изменений в модель бизнес-величин, не содержат сценариев запуска и остановки для существующих групп бизнес-величин. Вместо этого становятся доступными только сценарии запуска и остановки для новых групп бизнес-величин. Следует вручную обновить созданные ранее объединенные сценарии.

Следующий пример иллюстрирует этот случай: первоначальное развертывание службы перемещения данных для модели бизнес-величин *FinanceModel* содержит три группы бизнес-величин. Создано три сценария запуска и остановки компонента сбора данных для базы данных состояний. Впоследствии модель изменена и добавлены новые бизнес-величины. При развертывании создается только один сценарий запуска и остановки компонента сбора данных для новых групп бизнес-величин. Для включения службы перемещения данных необходимы четыре сценария запуска и остановки компонента сбора данных.

## Запуск и остановка службы перемещения данных

Запуск и остановка службы перемещения данных для данной модели бизнес-величин выполняется путем запуска и остановки связанных экземпляров компонентов сбора данных и применения изменений. Во время развертывания службы перемещения данных создаются сценарии запуска и остановки. Эти сценарии можно применять для запуска и остановки служб перемещения данных.

В топологии определяется, в каких системах должны выполняться экземпляры компонента. В основном, экземпляры компонентов сбора данных должны выполняться в системе, владеющей базой данных состояний (для службы перемещения данных из базы данных состояний в рабочую базу данных), и в системе, владеющей рабочей базой данных (для службы перемещения данных из рабочей базы данных в базу данных хронологии). Экземпляры компонентов применения изменений должны выполняться в системе, владеющей рабочей базой данных (для службы перемещения данных из базы данных состояний в рабочую базу данных), и в системе, владеющей базой данных хронологии (для службы перемещения данных из рабочей базы данных в базу данных хронологии). В этой конфигурации экземпляры компонентов применения изменений будут выталкивать данные из исходной базы данных (что приведет к увеличению быстродействия), если они находятся в системе, владеющей базой данных состояний (для службы перемещения данных из базы данных состояний в рабочую базу данных), и в системе, владеющей рабочей базой данных (для службы перемещения данных из рабочей базы данных в базу данных хронологии).

Следующая информация описывает запуск службы перемещения данных из базы данных состояний в рабочую базу данных и из рабочей базы данных в базу данных хронологии. Также описывается остановка службы перемещения данных из базы данных состояний в рабочую базу данных и из рабочей базы данных в базу данных хронологии.

**Примечание:** Службы перемещения данных из базы данных состояний в рабочую базу данных и из рабочей базы данных в базу данных хронологии не зависят друг от друга. Однако рекомендуется сначала запустить службу перемещения данных из базы данных состояний в рабочую базу данных. В некоторых случаях предпочтительно запускать службу перемещения данных из рабочей базы данных в базу данных хронологии после того, как Сервер монитора обработает записи для модели бизнес-величин, и после заполнения службой перемещения данных из базы данных состояний в рабочую базу данных таблиц рабочей базы данных, поддерживающих данную модель. Это позволяет поместить данные в базу данных хронологии быстрее, чем с помощью службы перемещения данных из рабочей базы данных в базу данных хронологии.

### Запуск службы перемещения данных из базы данных состояний в рабочую базу данных:

Архивы развертывания *DS\_State\_setup* и *DS\_Runtime\_setup* содержат исполняемые сценарии, которые можно использовать для запуска экземпляров компонентов сбора данных и применения изменений службы перемещения данных из базы данных состояний в рабочую базу данных. Если архив был создан в результате изменения в модели бизнес-величин, то в нем содержатся только сценарии запуска новых экземпляров компонентов сбора данных и применения изменений.

**Примечание:** Можно объединить сценарии для запуска службы перемещения данных. Дополнительная информация об объединении сценариев приведена в разделе “Объединение сценариев запуска и остановки” на стр. 45.

Однако приведенные ниже инструкции можно использовать даже в том случае, если объединение не выполнено. Для запуска службы перемещения данных из базы данных состояний в рабочую базу данных для определенной модели бизнес-величин:

1. Опознайте все экземпляры компонента сбора данных, присвоенные модели бизнес-величин в базе данных состояний.

Если уже выполнено объединение всех сценариев запуска экземпляра компонента сбора данных, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента сбора данных, которые были созданы для данной модели бизнес-величин. Сценарии запуска экземпляра компонента сбора данных автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии запуска только для новых экземпляров компонента сбора данных. Для того чтобы опознать все соответствующие сценарии запуска, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин.

- a. Перейдите в каталог, в который было выполнено развертывание данной модели.
- b. Перейдите в подкаталог `State_to_Runtime\source` и найдите все сценарии `StartCapture_<номер>`.
- c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.

2. Запустите экземпляры компонента сбора данных

Опознанные экземпляры компонента сбора данных должны быть запущены в системе, владеющей базой данных состояний. В случае объединения сценариев запуска запустите объединенный сценарий. Если объединение не выполнено, необходимо выполнить каждый сценарий запуска, опознанный на предыдущем шаге. Сценарии не должны запускаться одновременно, в противном случае, инициализация утилиты сбора данных не удастся. Однако, не имеет значения, в каком порядке запускаются сценарии запуска. ИД пользователя, запускающего экземпляры компонента сбора данных, должен содержать следующее:

- Права доступа администратора базы данных (DBADM) к базе данных состояний.
- Права доступа на запись в каталог, ссылка на который содержится в параметре `CAPTURE_PATH` в сценариях запуска.
- Права доступа на чтение файла, ссылка на который находится в необязательном параметре `PWDFILE` в сценариях запуска.

3. Опознайте все экземпляры компонента применения изменений, присвоенные модели бизнес-величин в рабочей базе данных.

Если уже выполнено объединение всех сценариев запуска экземпляра компонента применения изменений, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента применения изменений, которые были созданы для данной модели бизнес-величин. Сценарии запуска экземпляра компонента применения изменений автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии запуска только для новых экземпляров компонента применения изменений. Для того чтобы опознать все соответствующие сценарии запуска, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин:

- a. Перейдите в каталог, в который было выполнено развертывание данной модели.
- b. Перейдите в подкаталог `State_to_Runtime\target` и найдите все сценарии `StartApply_<номер>`.
- c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.

4. Запустите экземпляры компонента применения изменений

Опознанные экземпляры компонента применения изменений должны быть запущены в системе, владеющей рабочей базой данных. В случае объединения сценариев запуска запустите объединенный

сценарий. Если объединение не выполнено, необходимо выполнить каждый сценарий запуска, опознанный на предыдущем шаге. Сценарии не должны запускаться одновременно, в противном случае, инициализация утилиты применения изменений не удастся. Однако, не имеет значения, в каком порядке запускаются сценарии запуска. ИД пользователя, запускающего экземпляры компонента применения изменений, должен содержать следующее:

- Права доступа SELECT/INSERT/UPDATE/DELETE для связанных управляющих таблиц компонента сбора данных в базе данных состояний.
- Права доступа SELECT для связанных рабочих таблиц экземпляра компонента сбора данных в базе данных состояний.
- Права доступа SELECT/INSERT/UPDATE/DELETE для связанных промежуточных таблиц репликации в рабочей базе данных.
- Права доступа SELECT/INSERT/UPDATE/DELETE для связанных управляющих таблиц экземпляра компонента применения изменений в рабочей базе данных.
- Права доступа на запись в каталог, ссылка на который содержится в параметре *APPLY\_PATH* в сценариях запуска.
- Права доступа на чтение файла, ссылка на который находится в необязательном параметре *PWDFILE* в сценариях запуска.

5. Убедитесь, что все экземпляры компонентов сбора данных и применения изменений запущены успешно.

#### **Запуск службы перемещения данных из рабочей базы данных в базу данных хронологии:**

Архивы развертывания DS\_Runtime\_setup и DS\_Datamart\_setup содержат исполняемые сценарии, которые можно использовать для запуска экземпляров компонентов сбора данных и применения изменений службы перемещения данных из рабочей базы данных в базу данных хронологии. Если архив был создан в результате изменения в модели бизнес-величин, то в нем содержатся только сценарии запуска новых экземпляров компонентов сбора данных и применения изменений.

**Примечание:** Можно объединить сценарии для запуска службы перемещения данных. Дополнительная информация об объединении сценариев приведена в разделе “Объединение сценариев запуска и остановки” на стр. 45.

Однако приведенные ниже инструкции можно использовать даже в том случае, если объединение не выполнено. Для запуска службы перемещения данных из рабочей базы данных в базу данных хронологии для определенной модели бизнес-величин:

1. Опознайте все экземпляры компонента сбора данных, присвоенные модели бизнес-величин в рабочей базе данных.

Если уже выполнено объединение всех сценариев запуска экземпляра компонента сбора данных, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента сбора данных, которые были созданы для данной модели бизнес-величин. Сценарии запуска экземпляра компонента сбора данных автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии запуска только для новых экземпляров компонента сбора данных. Для того чтобы опознать все соответствующие сценарии запуска, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин.

- a. Перейдите в каталог, в который было выполнено развертывание данной модели.
- b. Перейдите в подкаталог Runtime\_to\_Historical\source и найдите все сценарии StartCapture\_<номер>.
- c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.

2. Запустите экземпляры компонента сбора данных.

Опознанные экземпляры компонента сбора данных должны быть запущены в системе, владеющей рабочей базой данных. В случае объединения сценариев запуска запустите объединенные сценарии. Если объединение не выполнено, необходимо выполнить каждый сценарий запуска, опознанный на предыдущем шаге. Сценарии не должны запускаться одновременно, в противном случае, инициализация



утилиты сбора данных не удастся. Однако, не имеет значения, в каком порядке запускаются сценарии запуска. ИД пользователя, запускающего экземпляры компонента сбора данных, должен содержать следующее:

- Права доступа администратора базы данных (DBADM) к рабочей базе данных.
- Права доступа на запись в каталог, ссылка на который содержится в параметре *CAPTURE\_PATH* в сценариях запуска.
- Права доступа на чтение файла, ссылка на который находится в необязательном параметре *PWDFILE* в сценариях запуска.

3. Опознайте все экземпляры компонента применения изменений, присвоенные модели бизнес-величин в базе данных хронологии.

Если уже выполнено объединение всех сценариев запуска экземпляра компонента применения изменений, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента применения изменений, которые были созданы для данной модели бизнес-величин. Сценарии запуска экземпляра компонента применения изменений автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии запуска только для новых экземпляров компонента применения изменений. Для того чтобы опознать все соответствующие сценарии запуска, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин:

- а. Перейдите в каталог, в который было выполнено развертывание данной модели.
- б. Перейдите в подкаталог `Runtime_to_Historical\target` и найдите все сценарии `StartApply_<номер>`.
- в. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.

4. Запустите экземпляры компонента применения изменений.

Опознанные экземпляры компонента применения изменений должны быть запущены в системе, владеющей базой данных хронологии. В случае объединения сценариев запуска запустите объединенные сценарии. Если объединение не выполнено, необходимо выполнить каждый сценарий запуска, опознанный на предыдущем шаге. Сценарии не должны запускаться одновременно, в противном случае, инициализация утилиты применения изменений не удастся. Однако, не имеет значения, в каком порядке запускаются сценарии запуска. ИД пользователя, запускающего экземпляры компонента применения изменений, должен содержать следующее:

- Права доступа `SELECT/INSERT/UPDATE/DELETE` для связанных управляющих таблиц компонента сбора данных в рабочей базе данных.
- Права доступа `SELECT` для связанных рабочих таблиц экземпляра компонента сбора данных в рабочей базе данных.
- Права доступа `SELECT/INSERT/UPDATE/DELETE` для связанных промежуточных таблиц репликации в базе данных хронологии.
- Права доступа `SELECT/INSERT/UPDATE/DELETE` для связанных управляющих таблиц экземпляра компонента применения изменений в базе данных хронологии.
- Права доступа на запись в каталог, ссылка на который содержится в параметре *APPLY\_PATH* в сценариях запуска.
- Права доступа на чтение файла, ссылка на который находится в необязательном параметре *PWDFILE* в сценариях запуска.

5. Убедитесь, что все экземпляры компонентов сбора данных и применения изменений запущены успешно.

**Остановка службы перемещения данных из базы данных состояний в рабочую базу данных:**

Процесс остановки службы перемещения данных из базы данных состояний в рабочую базу данных очень похож на процесс запуска этой службы. Архивы развертывания `DS_State_setup` и `DS_Runtime_setup` содержат исполняемые сценарии, которые можно использовать для остановки экземпляров компонентов сбора данных и применения изменений службы перемещения данных из рабочей базы данных в базу данных хронологии.

Если архив был создан в результате изменения в модели бизнес-величин, то в нем содержатся только сценарии остановки новых экземпляров компонентов сбора данных и применения изменений.

**Примечание:** Рекомендуется объединить сценарии до остановки службы перемещения данных. Дополнительная информация об объединении сценариев репликации приведена в разделе “Объединение сценариев запуска и остановки” на стр. 45.

Однако приведенные ниже инструкции можно использовать даже в том случае, если объединение не выполнено.

Для остановки службы перемещения данных из базы данных состояний в рабочую базу данных для определенной модели бизнес-величин:

1. Опознайте все экземпляры компонента сбора данных, присвоенные модели бизнес-величин в базе данных состояний. Если уже выполнено объединение всех сценариев остановки экземпляра компонента сбора данных, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента сбора данных, которые были созданы для данной модели бизнес-величин. Сценарии остановки экземпляра компонента сбора данных автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии остановки только для новых экземпляров компонента сбора данных. Для того чтобы опознать все соответствующие сценарии остановки, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин.
  - a. Перейдите в каталог, в который было выполнено развертывание данной модели.
  - b. Перейдите в подкаталог `State_to_Runtime\source` и найдите все сценарии `StopCapture_<номер>`.
  - c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.
2. Остановите экземпляры компонента сбора данных. Опознанные экземпляры компонента сбора данных должны быть остановлены в системе, владеющей базой данных состояний. В случае объединения сценариев остановки запустите объединенный сценарий. Если объединение не выполнено, необходимо выполнить каждый сценарий остановки, опознанный на предыдущем шаге. Не имеет значения, в каком порядке запускаются сценарии остановки.

**Примечание:** Сценарии остановки выполняются асинхронно. Может существовать задержка между запуском команды остановки и остановкой компонента сбора данных. Это происходит вследствие того, что экземпляр компонента сбора данных до остановки завершает транзакцию.

3. Опознайте все экземпляры компонента применения изменений, присвоенные модели бизнес-величин в рабочей базе данных. Если уже выполнено объединение всех сценариев остановки экземпляра компонента применения изменений, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента применения изменений, которые были созданы для данной модели бизнес-величин. Сценарии остановки экземпляра компонента применения изменений автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии остановки только для новых экземпляров компонента применения изменений. Для того чтобы опознать все соответствующие сценарии остановки, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин:
  - a. Перейдите в каталог, в который было выполнено развертывание данной модели.
  - b. Перейдите в подкаталог `State_to_Runtime\target` и найдите все сценарии `StopApply_<номер>`.
  - c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.
4. Остановите экземпляры компонента применения изменений.

Опознанные экземпляры компонента применения изменений должны быть остановлены в системе, владеющей рабочей базой данных. В случае объединения сценариев остановки запустите объединенный



сценарий. Если объединение не выполнено, необходимо выполнить каждый сценарий остановки, опознанный на предыдущем шаге. Сценарии остановки можно запускать в любом порядке.

**Примечание:** Сценарии остановки выполняются асинхронно. Может существовать задержка между запуском команды остановки и остановкой компонента применения изменений. Это происходит вследствие того, что экземпляр компонента применения до остановки завершает одну или несколько транзакций.

#### **Остановка службы перемещения данных из рабочей базы данных в базу данных хронологии:**

Процесс остановки службы перемещения данных из базы данных состояний в рабочую базу данных очень похож на процесс запуска этой службы. Архивы развертывания DS\_Runtime\_setup и DS\_Datamart\_setup содержат исполняемые сценарии, которые можно использовать для остановки экземпляров компонентов сбора данных и применения изменений службы перемещения данных из рабочей базы данных в базу данных хронологии.

Если архив был создан в результате изменения в модели бизнес-величин, то в нем содержатся только сценарии остановки новых экземпляров компонентов сбора данных и применения изменений.

**Примечание:** Рекомендуется объединить сценарии до остановки службы перемещения данных. Дополнительная информация об объединении сценариев репликации приведена в разделе “Объединение сценариев запуска и остановки” на стр. 45.

Однако приведенные ниже инструкции можно использовать даже в том случае, если объединение не выполнено.

Для остановки службы перемещения данных из рабочей базы данных в базу данных хронологии для определенной модели бизнес-величин:

1. Опознайте все экземпляры компонента сбора данных, присвоенные модели бизнес-величин в рабочей базе данных. Если уже выполнено объединение всех сценариев остановки экземпляра компонента сбора данных, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все экземпляры компонента сбора данных, которые были созданы для данной модели бизнес-величин. Сценарии остановки экземпляра компонента сбора данных автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии остановки только для новых экземпляров компонента сбора данных. Для того чтобы опознать все соответствующие сценарии остановки, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин.
  - a. Перейдите в каталог, в который было выполнено развертывание данной модели.
  - b. Перейдите в подкаталог Runtime\_to\_Historical\source и найдите все сценарии StopCapture\_<номер>.
  - c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.
2. Остановите экземпляры компонента сбора данных. Опознанные экземпляры компонента сбора данных должны быть остановлены в системе, владеющей рабочей базой данных. В случае объединения сценариев остановки запустите объединенный сценарий. Если объединение не выполнено, необходимо выполнить каждый сценарий остановки, опознанный на предыдущем шаге. Не имеет значения, в каком порядке запускаются сценарии остановки.

**Примечание:** Сценарии остановки выполняются асинхронно. Может существовать задержка между запуском команды остановки и остановкой компонента сбора данных. Это происходит вследствие того, что экземпляр компонента сбора данных до остановки завершает транзакцию.

3. Опознайте все экземпляры компонента применения изменений, присвоенные модели бизнес-величин в базе данных хронологии. Если уже выполнено объединение всех сценариев остановки экземпляра компонента применения изменений, никакие действия не требуются. Переходите к следующему шагу. Если сценарии еще не объединены (и вы не желаете выполнять объединение), следует опознать все

экземпляры компонента применения изменений, которые были созданы для данной модели бизнес-величин. Сценарии остановки экземпляра компонента применения изменений автоматически создаются в процессе первого выполнения генератора схемы для модели бизнес-величин. При последующих генерациях схемы (например, после обновления модели бизнес-величин) создаются сценарии остановки только для новых экземпляров компонента применения изменений. Для того чтобы опознать все соответствующие сценарии остановки, необходимо повторить следующие шаги для каждого развертывания, выполненного для данной модели бизнес-величин:

- a. Перейдите в каталог, в который было выполнено развертывание данной модели.
  - b. Перейдите в подкаталог `Runtime_to_Historical\target` и найдите все сценарии `StopApply_<номер>`.
  - c. Повторите описанные выше шаги для каждого развертывания данной модели бизнес-величин.
4. Остановите экземпляры компонента применения изменений.
- Опознанные экземпляры компонента применения изменений должны быть остановлены в системе, владеющей рабочей базой данных. В случае объединения сценариев остановки запустите объединенный сценарий. Если объединение не выполнено, необходимо выполнить каждый сценарий остановки, опознанный на предыдущем шаге. Сценарии остановки можно запускать в любом порядке.

**Примечание:** Сценарии остановки выполняются асинхронно. Может существовать задержка между запуском команды остановки и остановкой компонента применения изменений. Это происходит вследствие того, что экземпляр компонента сбора данных до остановки завершает одну или несколько транзакций.

## Развертывание схемы базы данных Cube Views

Генератор схемы создает метаданные Cube Views в файле формата XML. Они представляют определения DB2 Cube Views, соответствующие модели бизнес-величин. Определения cube views развертываются и на платформе Windows, и на платформе AIX.

### Развертывание схемы базы данных Cube Views в системе Windows.:

Метаданные Cube Views хранятся в папке вывода генератора схемы. Эта папка вывода определяется пользователем с помощью административной консоли WebSphere Business Monitor.

Для развертывания файла метаданных Cube Views выполните следующие действия:

1. Запустите центр DB2 OLAP. Появится окно диалога **Подключение к базе данных DB2**.
2. В окне диалога **Подключение к базе данных DB2** выполните следующее:
  - a. В поле **Имя базы данных** введите имя базы данных хронологии.
  - b. В поле **Имя пользователя** введите ИД пользователя, имеющего права доступа администратора к базе данных.
  - c. В поле **Пароль** введите пароль пользователя, имеющего права доступа администратора к базе данных.
  - d. Нажмите **ОК**.
  - e. При первом подключении к базе данных может появиться сообщение о том, что базу данных необходимо настроить для Cube Views. Нажмите **Да** для начала инициализации и конфигурации.
3. В окне OLAP Center импортируйте файл метаданных Cube Views следующим образом:
  - a. В меню выберите **OLAP Center → Импорт**. Будет запущен мастер импорта.
  - b. Выберите файл XML Cube Views, который хранится в папке вывода генератора схемы. Имя файла - `model_cv.xml`.
  - c. Нажмите **Готово**. Начнется процесс импорта.
4. После завершения импорта на странице **Опции импорта** в окне **Мастер импорта** нажмите **Готово**.

### Развертывание схемы базы данных Cube Views в системе AIX:

Метаданные Cube Views хранятся в папке вывода генератора схемы. Эта папка вывода определяется пользователем с помощью административной консоли WebSphere Business Monitor.

Для развертывания файла метаданных Cube Views выполните следующие действия:

1. Откройте редактор DB2 **Command Window**.
2. Подключитесь к базе данных хронологии в качестве пользователя базы данных (например: db2inst1), выполнив следующую команду: **db2 connect to имя\_базы\_данных\_хронологии**.
3. Измените каталог на **<Домашний\_каталог\_DB2>/sqllib/misc**, а затем выполните команду: **db2 -tvf db2mdapi.sql**.
4. Выполните команду: **db2mdapiclient -d HISTORY -i <каталог\_генерации>/schemagen/import\_model.xml -m <каталог\_генерации>/schemagen/model\_cv.xml -u <ИД\_пользователя> -p <pw> -o <каталог\_генерации>/schemagen/myoutput.xml**.

Где:

- -d - это имя базы данных хронологии.
- -i - это файл import\_model.xml, созданный генератором схемы.
- -u - это ИД пользователя.
- -p - это пароль.
- -o это имя файла операции вывода, в котором хранится вывод DB2.
- -m это команда метаданных ввода или команда DB2. Генератор схемы создает файл model\_cv.xml, который используется в качестве многомерных метаданных.
- <каталог\_генерации> указывает на каталог вывода, в котором генератор схемы сохраняет создаваемые артефакты.

Пример:

```
su - db2inst1
db2 connect to HISTORY
cd /home/db2inst1/sqllib/misc
db2 -tvf db2mdapi.sql
db2mdapiclient -d HISTORY
-i /opt/IBM/WebSphere/Monitor/generation/schemagen/import_model.xml
-m /opt/IBM/WebSphere/Monitor/generation/schemagen/model_cv.xml
-u db2inst1 -p monPa55w -o /tmp/import_output.xml
```

### Создание кубов ABX вручную:

Кубы ABX создаются вручную в системе, в которой установлен сервер IBM DB2 ALPHABLOX. Эти кубы будут использоваться сводными панелями WebSphere Business Monitor.

После развертывания определений cube views, но перед использованием сводных панелей выполните следующие действия:

1. Укажите в web-браузере: **http://<имя\_хоста>:9081/AlphaBlox/home/Admin**, и войдите в административную консоль IBM DB2 ALPHABLOX.
2. Выберите вкладку **ADMINISTRATION**.
3. Нажмите **Cubes**.
4. Для создания куба нажмите **Создать**.
  - a. В списке **Относительный источник данных** выберите соответствующую базу данных хронологии, определенную при установке.
  - b. Отметьте переключатель **Включено** возле **Имя куба DB2 AlphaBlox**.
  - c. Включите переключатель **Включить параметры DB2 Cube Views**. Подождите несколько секунд, пока поля не станут видимыми.

5. Для каждого куба, определенного в **Модели куба**, необходимо создать куб.
  - a. В списке **Модель куба** выберите модель.
  - b. В списке **Куб** выберите куб. Для каждой модели куба существует один куб.
  - c. В поле **Имя куба DB2 AlphaBlox** введите имя куба. Имя должно в точности соответствовать имени в списке **Куб**. Пример: CISS.NOOP. Не включайте в имя CISS. Это имя схемы.
  - d. Выберите опцию **Использовать бизнес-имена**.
  - e. Выберите **Импорт определения куба** и ожидайте выполнения процесса.
6. Для сохранения куба нажмите **ОК**.
7. Повторите шаг 5 (создание куба) для каждого существующего куба.

## Заполнение размерных таблиц вручную

Могут существовать данные, которые будут использоваться как размерные данные (например, база данных сведений о клиентах, которую следует заполнить в размерности заказчиков). Можно использовать базу данных хронологии, для того чтобы вручную заполнить размерные таблицы этими данными.

При заполнении этих таблиц следует иметь в виду несколько моментов.

- Следует обратить внимание на создание размерности в WebSphere Business Modeler, так чтобы эту размерность можно было заполнить существующими данными. Убедитесь, что размерность, определенная в WebSphere Business Modeler, содержит подходящие показатели с соответствующими типами данных, так чтобы можно было сохранить существующие данные в размерной таблице, созданной генератором схемы.
- При внесении данных вручную используйте отрицательные значения в поле SK\_<>. Это суррогатный ключ для таблицы. Положительные значения суррогатного ключа используются службами данных при заполнении этих таблиц; для того чтобы избежать конфликтов, следует использовать отрицательные значения.
- При вставке данных в размерную таблицу убедитесь, что для поля не установлено пустое значение. Если нет значащего значения для вставки в данное поле, необходимо выбрать и использовать значащее значение по умолчанию. Никогда не вставляйте в эту таблицу пустое значение. Однако, пустая строка ("" ) допустима для строковых типов данных.
- При отображении экземпляров процессов на внесенные в таблицу размерные данные экземпляр процесса может не соответствовать существующим данным (например, процесс, связанный новым заказчиком, который в данный момент не включен в размерную таблицу). В этом случае в таблице для этого набора данных будет создана новая запись. Таблица теперь содержит введенные ранее данные и эти новые данные.
- Неключевой атрибут размерности обновляется при поступлении новых данных. Например, предположим, что имеется размерность заказчика, в которой ключевой показатель - "Имя заказчика", а неключевой показатель - "Кредит". Первоначально данная таблица может содержать запись ['Widgets, Inc', 50000] в соответствии с текущими данными о заказчике. При обработке нового события, содержащего значение 75000 для Кредита в записи 'Widgets, Inc', запись в таблице размерностей заказчика будет изменена на ['Widgets, Inc', 75000]. Это изменение происходит только в том случае, если ключевой показатель соответствует существующей записи, а неключевой показатель - нет. При этом неключевые значения обновляются в соответствии с новыми данными.

Для того чтобы установить, какая из размерных таблиц соответствует размерности, заполненной вручную, и какие поля в этой таблице соответствуют различным атрибутам размерности, используйте текстовый файл *datamartMapping.txt*, расположенный в каталоге вывода генератора схемы (после запуска генератора схемы).

---

## Схема базы данных хронологии

Схемы баз данных описывают таблицы базы данных и связи между ними. С помощью схем баз данных можно планировать их размер.

Информация о схемах базы данных хронологии помогает понять взаимосвязь между импортированной моделью бизнес-величин и таблицами базы данных. Сводные панели используют базу данных хронологии для многомерного анализа и составления отчетов.

**Примечание:**

- Базы данных хранилища, состояний и рабочая база данных предназначены не только для внутреннего использования, и они могут быть изменены без предупреждения.
- Пользовательский код для прямого доступа к рабочей базе данных и базам данных хранилища и состояний не поддерживаются IBM.
- Вы не можете создать собственную сводную панель с помощью схемы базы данных хронологии.

База данных хронологии изначально заполнена данными дата/время в диапазоне с 1995 года по 2009 год. Если предполагается наличие дат/времени (либо в качестве времени начала/завершения процесса либо других данных показателя) вне этого диапазона, следует использовать следующий сценарий SQL для добавления дат в таблицу DIM\_TIME базы данных хронологии:

```
insert into <имя схемы WBI>.dim_time( surrogate_key, year, month, day)
with WBITIME (skey, ldate) as
(select surrogate_key+1 as skey,
COALESCE(
DATE(SUBSTR(DIGITS(YEAR),7,4) || '-' ||
SUBSTR(DIGITS(MONTH),4,2) || '-' ||
SUBSTR(DIGITS(DAY), 4,2)) + 1 DAYS,
DATE('YYYY-MM-DD первого дня в том случае,
если таблица DIM_TIME пуста.')
)as ldate
from sysibm.sysdummy1, <имя схемы WBI>.dim_time
where
DATE(
SUBSTR(DIGITS(YEAR) ,7,4) || '-' ||
SUBSTR(DIGITS(MONTH),4,2) || '-' ||
SUBSTR(DIGITS(DAY) ,4,2)
) =
(
SELECT
MAX(
DATE(SUBSTR(DIGITS(YEAR),7,4) || '-' ||
SUBSTR(DIGITS(MONTH),4,2) || '-' ||
SUBSTR(DIGITS(DAY), 4,2)))
FROM <имя схемы WBI>.DIM_TIME
)
UNION ALL
SELECT parent.skey+1, ldate + 1 DAYS
from WBITIME parent
where YEAR(ldate + 1 days) < where YEAR(ldate + 1 days) <
<YYYY 4-значный год, в который не следует завершать данные>
)
select a.skey, year(a.ldate), month(a.ldate), day(a.ldate)
from WBITIME a
WHERE
a.ldate >= DATE('YYYY-MM-DD: Начало диапазона, который следует ввести.')
AND a.ldate <= DATE('YYYY-MM-DD: Окончание диапазона, который следует ввести.')
```

**Примечание:** Существует четыре места в этом сценарии, которые обновляются для указания начальной и конечной дат, которые следует вставить в таблицу DIM\_TIME. Существует также четыре места, в которых необходимо указать имя схемы WBI (обычно "WBI").

---

## Глава 2. Службы баз данных

Эта справочная информация поможет вам в работе со службами баз данных.

---

### Схема базы данных хронологии

В базе данных хронологии есть два вида таблиц. Это статические таблицы, созданные во время установки WebSphere Business Monitor, и динамические таблицы, создаваемые для каждой импортируемой модели бизнес-величин.

Описание обоих типов таблиц базы данных хронологии и связь столбцов с моделью бизнес-величин приведены в следующих таблицах.

#### Примечание:

- Допускается пустое значение: означает, что значение в этом столбце может быть равным null
- Описание: соответствие между столбцом и определениями модели бизнес-величин. Описание необязательно должно быть в каждом столбце.

### Статические таблицы базы данных

#### *DIM\_TIME*

Размерность времени.

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
SURROGATE_KEY	INTEGER	Первичный ключ	N
DAY	SMALLINT	День	N
MONTH	SMALLINT	Месяц	N
YEAR	INTEGER	Год	N

### Динамические таблицы базы данных

База данных построена в виде звезды, где центральная таблица Fact окружена несколькими отходящими от нее "листьями" размерностей. Таблица Факт подобна таблице context в рабочей базе данных и базе данных состояний. Таблица context является центром звезды, другая звезда отходит от таблицы операций для context. Например, может быть один экземпляр для context, и для каждой такой таблицы по одной таблице экземпляра операций в рабочей базе данных и базе данных состояний.

#### *Таблица Context Fact*

Соглашение об именах: FCT\_<системное сгенерированное имя контекста>

Следующие столбцы присутствуют всегда:

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
MCI_MCPID	DECIMAL(19,0)	Уникальный идентификатор экземпляра операций и первичный ключ таблицы.	N
PARENT_MCPID	DECIMAL(19,0)	Уникальный идентификатор родительского экземпляра процесса, если он имеется.	Y



### Таблица Context Fact

Соглашение об именах: FCT\_<системное сгенерированное имя контекста>

Следующие столбцы присутствуют всегда:

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
SK_<системное сгенерированное название размерности>	INTEGER	Внешний ключ для таблицы размерностей. Зависимость FK определена. Один из этих столбцов определен для каждой из размерностей, присутствующих в контексте.	Y
GMT_<системное сгенерированное название показателя>	TIMESTAMP	Здесь хранится время по Гринвичу для любого типа данных показателей системного времени. (Этот столбец создается только для показателей типа времени, помеченных как размерности. Когда показатели типа времени помечены как размерности, они хранятся с точностью до дня, а для них требуется знать точное время.)	Y

Следующие три типа столбцов применяются только для показателей типа факт (не размерность).

Типы столбцов, которые применяются только для показателей типа факт

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
M_<системное сгенерированное имя>	Тип данных, зависящий от типа, определенного в модели бизнес-величин.	Представляет значение показателя или определения ключа.	Y
C_<системное сгенерированное имя>	BIGINT	Представляет счетчики.	Y
T1_<системное сгенерированное имя>	BIGINT	Представляет полное время для таймеров. (Таймерам соответствует один столбец в базе данных хронологии и несколько столбцов в базе данных состояний.)	Y

### Таблица размерностей

В каждом контексте могут быть определены ноль или более таблиц размерностей в зависимости от от числа размерностей . Как правило, задана хотя бы размерность времени.

Соглашение об именах: DIM\_<системное сгенерированное имя размерности>

Следующие столбцы определены всегда:

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
SURROGATE_KEY	INTEGER	Первичный ключ для этой строки размерности, сгенерированный системой. РК определен.	N

Далее перечислены столбцы, зависящие от определения. В таблице размерностей будет задан столбец для каждого из показателей, определенных для этой размерности.

Столбцы, зависящие от определения

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
M_<системное сгенерированное имя>	Тип данных, зависящий от типа, определенного в модели бизнес-величин.	Представляет значение показателя или определения ключа.	Y
C_<системное сгенерированное имя>	BIGINT	Представляет счетчики.	Y
T1_<системное сгенерированное имя>	BIGINT	Представляет полное время для таймеров. (Таймерам соответствует один столбец в базе данных хронологии и несколько столбцов в базе данных состояний.)	Y

Схема звезды операций применяет те же соглашения об именах, но приставки таблиц имеют вид AFC\_ и ADM\_ соответственно.

## Управляющая таблица службы перемещения данных

В этом разделе описана структура управляющей таблицы службы перемещения данных. Базы данных состояния, рабочая и хронологии содержат две управляющих таблицы. Эти таблицы позволяют настроить работу служебных компонентов, отвечающих за перемещение локальных данных. Управляющие таблицы являются статическими.

### RMCONTROL

Содержит параметры конфигурации, отвечающие за работу экземпляров компонента ETL. Эта таблица заполняется и применяется только в рабочей базе данных и базе данных хронологии, потому что компонент ETL не требуется в базе данных состояний. Каждая строка в этой таблице соответствует одной целевой таблице, которую требуется заполнить. Изменения значений столбцов для какой-либо строки влияют только на экземпляр компонента ETL, который был выделен для заполнения этой целевой таблицы.

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
TARGETTABLE	CHARACTER	Полное имя целевой таблицы, которая заполняется хранимой процедурой, управляемой этой записью.	N
COMMITINTERVAL	NUMERIC	Интервал фиксации, используемый хранимой процедурой при работе с курсором для вставки строк в целевую таблицу.	Y
LOGLEVEL	NUMERIC	Уровень протокола, задающий объем информации, заносимой хранимой процедурой в таблицу WBIRMADM.RMLOG. Допустимые значения: 0 и 1. 0 - минимальный уровень, 1 - максимальный.	Y
LASTSEQUENCE	CHARACTER	Последнее значение SEQUENCE, обработанное хранимой процедурой ETL из промежуточной таблицы. Этот столбец обновляется хранимой процедурой динамически.	N

## **RMCONTROL**

Содержит параметры конфигурации, отвечающие за работу экземпляров компонента ETL. Эта таблица заполняется и применяется только в рабочей базе данных и базе данных хронологии, потому что компонент ETL не требуется в базе данных состояний. Каждая строка в этой таблице соответствует одной целевой таблице, которую требуется заполнить. Изменения значений столбцов для какой-либо строки влияют только на экземпляр компонента ETL, который был выделен для заполнения этой целевой таблицы.

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
LASTUPDATED	TIMESTAMP	Время последнего запланированного вывода. Этот столбец управляется хранимой процедурой и применяется только для целей планирования.	Y
NEXTSTARTTIME	TIMESTAMP	Время следующего вызова ETL.	Y
ETLSCHEDMETHOD	NUMERIC	Применяемый метод планирования. Должен быть задан равным 0.	Y
ETL_0_MINUTES	NUMERIC	Время в минутах между запланированными вызовами ETL.	Y
TGT_RM_SPETL_NAME	CHARACTER	Полное имя хранимой процедуры, отвечающей за заполнение целевой таблицы.	Y

Во избежание ошибок не изменяйте значения в следующих столбцах:

- TARGETTABLE
- LASTSEQUENCE
- LASTUPDATED
- ETLSCHEDMETHOD
- TGT\_RM\_SPETL\_NAME

Изменения в данных столбцах вступят в силу при следующем вызове экземпляра компонента ETL:

- COMMITINTERVAL
- NEXTSTARTTIME
- LOGLEVEL
- ETL\_0\_MINUTES

## **RMPRUNECTRL**

Содержит параметры конфигурации, отвечающие за работу экземпляров компонента жизненного цикла. Эта таблица заполняется и используется базами данных состояний, хронологии и рабочей. Каждая строка в этой таблице соответствует одной исходной или рабочей таблице <TABLE\_NAME>, которую требуется очистить. Изменения значений столбцов для какой-либо строки влияют только на экземпляр компонента жизненного цикла, который был выделен для очистки таблицы <TABLE\_NAME>.

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
TABLE_NAME	CHARACTER	Полное имя таблицы, которую требуется очистить.	N
LAST_PRUNED	TIMESTMP	Время последней очистки таблицы.	Y

## RMPRUNCTRL

Содержит параметры конфигурации, отвечающие за работу экземпляров компонента жизненного цикла. Эта таблица заполняется и используется базами данных состояний, хронологии и рабочей. Каждая строка в этой таблице соответствует одной исходной или рабочей таблице <TABLE\_NAME>, которую требуется очистить. Изменения значений столбцов для какой-либо строки влияют только на экземпляр компонента жизненного цикла, который был выделен для очистки таблицы <TABLE\_NAME>.

Название столбца	Тип столбца	Описание столбца	Допускается пустое значение
LOGLEVEL	NUMERIC	Уровень протокола, задающий объем информации, заносимой в таблицу WBIRMADM.RMLOG. Допустимые значения: 0 и 1. 0 - минимальный уровень; 1 - максимальный.	N
PRUNE_ENABLED	NUMERIC	Флаг, задающий необходимость операций очистки. 0 означает нет, 1 - да.	N
PRUNE_INTERVAL	NUMERIC	Минимальный интервал времени между операциями очистки.	N
RETENTION_IN_MINUTES	NUMERIC	Время в минутах, после которого будет очищена допустимая строка.	N
ROWS_PRUNED	NUMERIC	Число строк, очищенных в ходе последней операции очистки.	N

Во избежание ошибок не изменяйте значения в следующих столбцах:

- LAST\_PRUNED
- ROWS\_PRUNED
- TABLE\_NAME

Изменения в данных столбцах вступят в силу при следующем вызове экземпляра компонента ETL:

- LOGLEVEL
- PRUNE\_ENABLED
- PRUNE\_INTERVAL
- RETENTION\_IN\_MINUTES

## Таблица метаданных и протокола службы перемещения данных

В этом разделе приведена справочная информация о таблицах ведения протокола баз данных WebSphere Business Monitor. Таблицы протокола являются статическими.

### RMMETADATA

Для служб перемещения данных в модели бизнес-величин применяется переменное число экземпляров компонентов. Для каждой модели бизнес-величин в базах данных хронологии, состояний и рабочей есть таблица со списком имен экземпляров компонентов и прочей полезной внутренней информацией. Эта таблица обновляется всякий раз, когда на стадии развертывания создаются и настраиваются экземпляры компонентов. Ее содержимое не следует изменять вручную.

Название столбца	Тип столбца	Описание столбца
ИД	NUMERIC	Не применяется

## RMMETADATA

Для служб перемещения данных в модели бизнес-величин применяется переменное число экземпляров компонентов. Для каждой модели бизнес-величин в базах данных хронологии, состояний и рабочей есть таблица со списком имен экземпляров компонентов и прочей полезной внутренней информацией. Эта таблица обновляется всякий раз, когда на стадии развертывания создаются и настраиваются экземпляры компонентов. Ее содержимое не следует изменять вручную.

Название столбца	Тип столбца	Описание столбца
OM_NAME	CHARACTER	Имя соответствующего проекта модели бизнес-величин, с которым работают эти объекты репликации.
OM_ID	NUMERIC	Не применяется
MC_NAME	CHARACTER	Имя соответствующей группы бизнес-величин, с которой работают эти объекты репликации.
MC_ID	NUMERIC	Не применяется
TGT_TAB_NAME	CHARACTER	Полное имя целевой таблицы, которая заполняется хранимой процедурой ETL.
TGT_RM_APP_SVR_NAME	CHARACTER	Имя сервера, отвечающего за выполнение операций apply репликации.
TGT_RM_APP_SS_NAME	CHARACTER	Группа, управляемая сервером операций репликации DB2 SQL (если она задана). Это набор подписки.
TGT_RM_APP_STG_TAB_NAME	CHARACTER	Полное имя промежуточной таблицы, применяемой программой выполнения репликации. <b>Примечание:</b> В системе существует две другие таблицы с расширениями <i>_BKUP</i> и <i>_M</i> , которые относятся к этой строке.
TGT_RM_APP_ERR_TAB_NAME	CHARACTER	Полное имя таблицы, сохраняющей указатели на строки в промежуточной таблице, которые еще будут обработаны ETL.
TGT_RM_APP_PRUNE_SP_NAME	CHARACTER	Полное имя хранимой процедуры, отвечающей за очистку промежуточной таблицы в целевой системе.
TGT_RM_APP_TMP_TAB_NAME	CHARACTER	Полное имя временной таблицы, которая применяется программой ETL для определения строк, которые будут загружены в целевую таблицу.
TGT_RM_SPETL_NAME	CHARACTER	Полное имя хранимой процедуры ETL, отвечающей за заполнение целевой таблицы данными из промежуточной таблицы.
SRC_TAB_NAME	CHARACTER	Полное имя исходной таблицы, которая будет скопирована в промежуточную таблицу.
SRC_RM_CAP_SVR_NAME	CHARACTER	Имя сервера, отвечающего за выполнение операций capture репликации.

## RM METADATA

Для служб перемещения данных в модели бизнес-величин применяется переменное число экземпляров компонентов. Для каждой модели бизнес-величин в базах данных хронологии, состояний и рабочей есть таблица со списком имен экземпляров компонентов и прочей полезной внутренней информацией. Эта таблица обновляется всякий раз, когда на стадии развертывания создаются и настраиваются экземпляры компонентов. Ее содержимое не следует изменять вручную.

Название столбца	Тип столбца	Описание столбца
SRC_RM_CAP_STG_TAB_NAME	CHARACTER	Полное имя таблицы, применяемой сервером фиксирования копии для сохранения изменений исходной таблицы.
SRC_RM_PRUNE_TRG_NAME	CHARACTER	Полное имя триггера, отвечающего за удаление выбранных строк из исходной таблицы во время цикла очистки сервера фиксирования копии. Выбранные ряды могут включать ряды с завершенными операциями.
SERVICE_NAME	CHARACTER	Метка, которая используется для идентификации службы, к которой принадлежат эти артефакты, например, State_to_Runtime или Runtime_to_Historical.

Рассмотрим простой пример:

Мы видим, что источник WBI.CTX\_TQ4MUF в базе данных состояний отслеживается экземпляром

OM_NAME	SRC_TAB_NAME	SRC_RM_CAP_SV...	SRC_RM_CA...	TGT_RM_AP...	TGT_RM_AP...	TGT_TAB_NAME	SERVICE_NAME
STEW_S	wbi.CTX_TQ4MUF...	CAPTURE_1	CAP.CD_2	APPLY_4	APP.CCD_6	wbi.CTR_TQ4MUF...	State to Runtime
STEW_S	wbi.AIR_BVSOYAP...	CAPTURE_1	CAP.CD_3	APPLY_4	APP.CCD_7	wbi.AIR_BVSOYA...	State to Runtime

компонента фиксации копии CAPTURE\_1. Все изменения в исходной таблице записываются в рабочую таблицу CAP.CD\_2 и затем заносятся экземпляром компонента APPLY\_4 в рабочую таблицу APP.CCD\_6. Эта таблица используется экземпляром компонента ETL для заполнения целевой таблицы WBI.CTX\_TQ4MUF\_TQ4MUF в рабочей базе данных.

## RM LOG

Рабочая база данных и база данных хронологии содержат таблицу протоколирования, в которой накапливается информация о статистике, ходе выполнения операций, отладке и ошибках. Все компоненты ETL и компоненты жизненного цикла целевого объекта пишут в эту таблицу, не читая из нее. Некоторые сообщения можно подавить, задав минимальный уровень протокола.

Название столбца	Тип столбца	Описание столбца
ENTRYSTMP	TIMESTAMP(10)	Системное время записи в таблице.
ИД	NUMERIC	Идентификатор для обозначения нескольких строк, относящихся к одному и тому же экземпляру. Этот ИД происходит из SEQUENCE WBIRMDADM.RMSPTRIGID.
ROWS_INSERTED	NUMERIC	Указывает, сколько строк были вставлены при работе этого экземпляра.
ROWS_UPDATED	NUMERIC	Указывает, сколько строк были обновлены при работе этого экземпляра.
ROWS_DELETED	NUMERIC	Указывает, сколько строк были удалены при работе этого экземпляра.

## RMLOG

Рабочая база данных и база данных хронологии содержат таблицу протоколирования, в которой накапливается информация о статистике, ходе выполнения операций, отладке и ошибках. Все компоненты ETL и компоненты жизненного цикла целевого объекта пишут в эту таблицу, не читая из нее. Некоторые сообщения можно подавить, задав минимальный уровень протокола.

Название столбца	Тип столбца	Описание столбца
ROWS_INERROR	NUMERIC	Указывает, сколько строк были помечены как вызывающие неисправимую ошибку при работе этого экземпляра.
NAME	CHARACTER	Полное имя хранимой процедуры, триггера или процесса, занесшего эту запись в таблицу.
OPERATION	CHARACTER	Указывает операцию, выполнявшуюся во время создания этой записи.
RESULT	CHARACTER	Столбец с дополнительной информацией об операции.
ISTRACEENTRY	NUMERIC	Указывает, требует ли эта запись задания LOGLEVEL (в WBIRMADM.RMCONTROL) равным 1, или нет.  0: Эта запись протокола не является трассировочной.  1: Эта запись протокола является трассировочной и может быть выключена - см. таблицу WBI.RMCONTROL.

Каждая строка таблицы соответствует сообщению, созданному экземпляром компонента <NAME> в <ENTRYSTMP>. Строки с одинаковыми <ID> и <NAME> представляют сообщения, сгенерированные во время одного и того же вызова <NAME>. Далее приведен пример записей протокола, которые создаются экземплярами компонента ETL WBIRMADM.WBIRMSP\_10 и WBIRMADM.WBIRMSP\_14 и экземплярами компонента жизненного цикла целевого объекта WBIRMADM.WBIRMSP\_P13 и WBIRMADM.WBIRMSP\_P\_17. WBIRMADM.WBIRMSP\_10 (4:40:20 PM) и WBIRMADM.WBIRMSP\_14 (4:40:27 PM) отправили по пять сообщений, а WBIRMADM.WBIRMSP\_P\_13 (4:40:20 PM) и WBIRMADM.WBIRMSP\_P\_17 (4:40:20 PM) - по одному.

Эта таблица не очищается автоматически. Администратор баз данных должен регулярно отслеживать и

ENTRYSTMP	ID	NAME	OPERATION	ROWS_INSERTED
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	SP_START	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	DEL_TEMP	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	INS_TEMP	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	FETCH_TARGET_...	0
Oct 11, 2005 4:40:20 PM 3...	1	WBIRMADM.WBIRMSP_10	SP_END	0
Oct 11, 2005 4:40:20 PM 3...	2	WBIRMADM.WBIRMSP_P_13	PRUNESTAGING	0
Oct 11, 2005 4:40:20 PM 3...	3	WBIRMADM.WBIRMSP_P_17	PRUNESTAGING	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	SP_START	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	DEL_TEMP	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	INS_TEMP	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	FETCH_TARGET_...	0
Oct 11, 2005 4:40:27 PM 1...	4	WBIRMADM.WBIRMSP_14	SP_END	0

сокращать их. Сведения из WBIRMADM.RMMETADATA помогают определить, для какой модели бизнес-величин экземпляр компонента <NAME> выполняет служебные функции. Обратите внимание, что значения столбцов LOGLEVEL и ETL\_0\_MINUTES из таблицы WBIRMADM.RMCONTROL и значения LOGLEVEL и PRUNE\_INTERVAL из таблицы WBIRMADM.RMPRUNECTRL влияют на скорость роста этой таблицы. Число записей возрастает, если LOGLEVEL равен 1, или с уменьшением значений ETL\_0\_MINUTES и PRUNE\_INTERVAL.



---

## Глава 3. Устранение неполадок служб базы данных

При генерации, развертывании или выполнении служб базы данных WebSphere Business Monitor могут произойти ошибки, связанные со службами базы данных. Далее приведена информация об устранении неполадок, связанных с базой данных.

---

### Вопросы развертывания

Во время выполнения различных сценариев развертывания WebSphere Business Monitor артефактов баз данных, могут произойти ошибки. Ниже предложены способы исправления для каждой ошибки.

В процессе развертывания созданных артефактов базы данных могут возникать неполадки, которые являются результатом

- неправильной конфигурации
- недостаточных прав доступа пользователя
- Неполадки установки среды

*Таблица 2. Ошибки развертывания*

Ошибка	Исправление
Табличное пространство, назначенное таблице, не существует.	<ul style="list-style-type: none"><li>• Убедитесь в том, что табличные пространства, определенные в файле свойств табличного пространства, существуют и имеют описанные параметры.</li><li>• Или создайте недостающее пространство таблиц с подходящими параметрами и перезапустите сценарии DDL, или измените файл свойств табличного пространства, а затем повторно сгенерируйте схему.</li></ul>
Назначенное таблице пространство таблиц имеет слишком маленький размер для ее хранения.	<ul style="list-style-type: none"><li>• Убедитесь в том, что табличные пространства, определенные в файле свойств табличного пространства, существуют и имеют описанные параметры.</li><li>• Или исправьте это и перезапустите сценарии, или просто вручную измените в DDL присваивание табличного пространства.</li></ul>

Таблица 2. Ошибки развертывания (продолжение)

Ошибка	Исправление
Таблицы уже существуют в базе данных.	<p>Если тот же самый сценарий DDL не выполнялся раньше, то имеется две возможных причины этой ошибки.</p> <ul style="list-style-type: none"> <li>При создании этих сценариев DDL была выбрана опция <b>Игнорировать предыдущие развертывания</b>. Генератор схемы генерирует новые операторы создания таблиц, а не изменяет существующие таблицы. Эта опция должна использоваться только при попытке создания базы данных с нуля. То есть, после удаления существующих таблиц базы данных. Если имеются таблицы, которые необходимо сохранить, перезапустите генератор схемы без выбранной опции <b>Игнорировать предыдущие развертывания</b> и затем повторно выполните сценарии DDL.</li> <li>Другая причина может заключаться в том, что в какой-то момент версия этой модели бизнес-величин была удалена, и при этом не была выбрана опция <i>Удалить и сохранить для отчета</i>. Если версия модели бизнес-величин удалена без выбора опции <i>сохранить для отчета</i>, то генератор схемы не может продолжать управлять изменениями в поддерживаемых таблицах базы данных для этой модели бизнес-величин. В этом случае есть две возможности. <ul style="list-style-type: none"> <li>Создайте новую модель бизнес-величин на основе текущей и разверните ее как новую модель бизнес-величин с новым набором таблиц. Данные из существующих таблиц можно вручную перенести в новые таблицы.</li> <li>Удалите вручную существующие таблицы, поддерживающие эту модель бизнес-величин, используя предоставленные файлы преобразования в качестве руководства. После удаления таблиц перезапустите генератор схемы с опцией <b>Игнорировать предыдущие развертывания</b>. Полученные в результате сценарии DDL создадут новый набор таблиц, который поддерживает эту последнюю версию модели бизнес-величин.</li> </ul> </li> </ul> <p><b>Примечание:</b> Если вы не создадите вручную резервной копии данных из старых таблиц перед их удалением и затем перенесете данные во вновь созданные таблицы, то в хронологии не окажется информации о процессах, которые выполнялись в предыдущих версиях модели бизнес-величин.</p> <p>При удалении модели не рекомендуется использовать опцию <i>удалить</i> вместо опции <i>удалить и сохранить для отчета</i>, если только вы не собираетесь развернуть новую версию этой модели бизнес-величин.</p>
Табличное пространство слишком мало. (Хотя пространство таблиц, первоначально присвоенное этой таблице, было достаточным для размера поля, добавленные впоследствии показатели вышли за пределы размера страницы текущего табличного пространства.)	<p>Необходимо сохранить эту таблицу, удалить ее, и затем создать повторно, присвоив ей табличное пространство большего размера. Резервная копия данных должна быть затем загружена в новую таблицу. Создав повторно текущую таблицу в большем табличном пространстве, вы сможете выполнить последние сценарии DDL, которые добавляют необходимые поля.</p>

По разным причинам вы можете захотеть создать таблицы базы данных заново. Например, многие имеющиеся показатели, уже не представляющие интереса, продолжают находится в таблицах, потому что они существовали в предыдущих версиях модели бизнес-величин. Простейшим способом является переименование проекта в WebSphere Business Modeler. Модель бизнес-величин будет считаться новой, и в базе данных будут созданы новые уникальные таблицы.

**Примечание:** В этом случае не будет доступна информация хронологии из предыдущих экземпляров процессов.

Если нужно иметь доступ к информации хронологии, вы можете скопировать данные из первоначальных таблиц во вновь созданные таблицы. Имена столбцов не будут соответствовать, но типы данных останутся

теми же. Можно использовать файлы преобразования (созданные сценариями DDL) или комментарии к полям базы данных для определения соответствий между полями и показателями, и между таблицами и процессами.

**Примечание:** Некоторые поля исходных таблиц не будут иметь соответствующих полей в новых таблицах, если последняя версия модели бизнес-величин не содержит больше этих показателей.

---

## Вопросы выполнения

При перезапуске Сервера сбора данных, который не работал несколько дней, вы получите сообщение об ошибке, сгенерированное Сервером сбора данных из репликации IBM DB2. Это сообщение появляется в окне сбора данных системы Windows, в системном файле протокола, посылается по электронной почте как часть передачи данных монитора репликации или в таблице IBMSNAP\_CAPTRACE.

### Сообщение об ошибке

*ASN0121E CAPTURE "CAPTURE\_141" : "WorkerThread". "Горячий" запуск программы записи был неудачным. Существующие данные слишком устарели. Программа сбора данных будет прервана.*

Решение этой проблемы описано в разделе Подготовка развертывания артефактов базы данных

---

## Остановка базы данных Выполнение

При остановке или завершении работы рабочей базы данных WebSphere Business Monitor по какой-либо причине, сначала нужно остановить приложение диспетчера адаптивных действий.

Остановить приложение диспетчера адаптивных действий можно из административной консоли Сервера процессов WebSphere. Рабочая база данных WebSphere Business Monitor должна быть запущена перед запуском диспетчера адаптивных действий.