

HP Service Health Reporter

Версия ПО: 9.40

Операционные системы Windows®

Руководство по разработке содержимого

Дата выпуска документа: Январь 2015 г.
Дата выпуска программного обеспечения: Январь 2015 г.



Официальные уведомления

Гарантийные обязательства

Компания HP не предоставляет никаких дополнительных гарантий в отношении продуктов и услуг, кроме тех, что отражены в заявлениях о явных гарантиях для соответствующих продуктов или услуг. Никакая часть настоящего документа не подразумевает принятия на себя дополнительных гарантийных обязательств. HP не несет ответственности за технические или лингвистические ошибки либо упущения, содержащиеся в этом документе.

Информация, содержащаяся в настоящем документе, может быть изменена без уведомления.

Пояснения в отношении ограниченных прав

Конфиденциальное компьютерное программное обеспечение. Для владения, использования или копирования необходима действующая лицензия компании HP. В соответствии с положениями FAR 12.211 и 12.212 коммерческое компьютерное программное обеспечение, документация компьютерного программного обеспечения и технические данные коммерческих продуктов лицензируются государственным учреждениям США на условиях стандартной коммерческой лицензии поставщика.

Заявление об авторских правах

© Copyright 2010 - 2015 Hewlett-Packard Development Company, L.P.

Уведомление о товарных знаках

Adobe® является товарным знаком корпорации Adobe Systems Incorporated.

Microsoft® и Windows® — товарные знаки корпорации Майкрософт, зарегистрированные в США.

UNIX® является зарегистрированным товарным знаком консорциума The Open Group.

Обновление документации

На титульной странице настоящего документа представлены следующие идентификационные данные.

- Номер версии программного обеспечения.
- Дата выпуска документа, которая изменяется при каждом его обновлении.
- Дата выпуска программы, которая указывает дату выпуска данной версии ПО.

Чтобы проверить наличие обновлений или убедиться в том, что используется последняя редакция документа, откройте сайт

<http://h20230.www2.hp.com/selfsolve/manuals>

Для доступа к этому сайту необходимо зарегистрироваться в службе HP Passport и войти в систему. Регистрация HP Passport ID производится на сайте <http://h20229.www2.hp.com/passport-registration.html>

или по ссылке **New users - please register** на странице входа в HP Passport.

Оформление подписки в службе поддержки соответствующего продукта также позволит получать обновленные и новые редакции. Подробные сведения можно получить у торгового представителя компании HP.

Поддержка

Сайт службы поддержки HP Software в Интернете находится по адресу: <http://www.hp.com/go/hpsupport>

На этом веб-сайте приведена контактная информация и подробные сведения о продуктах, услугах и поддержке, предоставляемых компанией HP в сфере программного обеспечения.

Служба поддержки программного обеспечения компании HP в Интернете предоставляет заказчикам возможности для самостоятельного устранения неполадок. Она обеспечивает быстрый и эффективный доступ к интерактивным средствам технической поддержки, которые необходимы для управления бизнесом. Клиенты службы поддержки могут использовать этот сайт для решения следующих задач:

- Поиск необходимых документов в базе знаний.
- Подача и отслеживание заявок в службу технической поддержки и запросов на расширение функциональных возможностей.
- Загрузка исправлений программного обеспечения.
- Управление договорами на оказание поддержки.
- Поиск контактной информации службы поддержки компании HP.
- Просмотр сведений о доступных услугах.
- Участие в обсуждениях с другими пользователями программного обеспечения.
- Поиск курсов обучения по программному обеспечению и регистрация для участия в них.

Для получения доступа к большинству разделов поддержки сначала необходимо зарегистрироваться в качестве пользователя службы HP Passport, а затем войти в систему. Для ряда разделов поддержки также необходимо наличие договора на оказание поддержки. Чтобы зарегистрировать учетную запись HP Passport, перейдите по адресу:

<http://h20229.www2.hp.com/passport-registration.html>

Получить более подробные сведения об уровнях доступа можно по адресу

http://h20230.www2.hp.com/new_access_levels.jsp

Веб-сайт портала **HP Software Solutions Now** содержит сведения о решениях и интеграциях HPSW. Этот сайт позволяет изучить продукты HP на предмет соответствия бизнес-потребностям, содержит полный список интеграций между продуктами HP, а также список процессов ITIL. URL-адрес этого веб-сайта: <http://h20230.www2.hp.com/sc/solutions/index.jsp>

Содержание

Часть I	10
Глава 1: Введение	11
Об этом руководстве	11
Кому следует ознакомиться с этим руководством	11
Предварительные условия и справочная документация	12
Глава 2: Архитектура и разработка пакетов содержимого	14
Архитектура пакетов содержимого	14
Компоненты пакета содержимого	14
Компонент «Сфера деятельности»	15
Извлечение, преобразование и загрузка компонента (ETL)	16
Компонент «Отчеты»	17
Представление среды разработки содержимого	17
Глава 3: Создание пакета содержимого — начало работы	19
Подготовка к разработке содержимого	21
Изучение сферы деятельности	21
Определение пользовательских ролей и требований к отчетности	21
Извлечение и настройка CPC	21
Когда следует использовать упрощенную CPC?	22
Создание и установка компонента «Сфера деятельности»	22
Создание структуры каталогов	23
определение детализации, измерений и фактов	24
разработка модели данных	24
Реализация модели данных в XML	25
создание потоков рабочих процессов	26
Использование CPC для создания пакета компонента «Сфера деятельности»	27
Установка пакета компонента «Сфера деятельности»:	28
Создание и установка компонента ETL	28
Настраиваемая загрузка данных с использованием файлов CSV	28
Проверка потоков рабочих процессов на административной консоли	29
Создание и установка компонента «Отчеты»	30
Предварительные условия	32
Создание структуры каталогов	32
Написание документа XML-модели	32
Создание XML-файла манифеста	33
Создание подключения к SAP BusinessObjects	33
Использование CPC для создания системы SAP BusinessObjects Universe	34
Экспорт системы Universe в репозиторий SAP BusinessObjects	34
создание отчетов Web Intelligence	36
Включение функции детализации по времени в отчетах	38
Просмотр отчетов в SAP BusinessObjects InfoView	38
Демонстрация отчета о розничных продажах	39

Экспорт отчетов в SHR	39
Экспорт файла Business Intelligence Archive Resource (BIAR)	39
Создание XML-файла манифеста	45
Использование CPCS для создания пакета компонента «Отчеты»	45
Установка пакета компонента «Отчеты»	45
Просмотр отчетов в SAP BusinessObjects InfoView	46
Разработка контента в Linux с помощью CPCS	46
Предварительные условия	46
Извлечение и настройка CPCS	47
Создание компонента приложения	47
Глава 4: Создание пакета компонента ETL	51
Предварительные требования	52
Создание и установка пакета компонента «Сфера деятельности»	52
Анализ источника данных	52
Создание структуры каталогов	52
Определение политики сбора данных в XML	52
Определение правил преобразования данных	53
Определение правил промежуточного хранения	53
определение потоков рабочих процессов	54
Создание компонента ETL	55
Установка компонента ETL	55
Работа с компонентом ETL	55
Настройка универсальной базы данных	55
Проверка компонента ETL	56
Просмотр отчетов	57
Глава 5: Создание пакета содержимого — упрощенный метод	58
Анализ источника данных	58
Универсальная база данных как источник данных	58
CSV-файлы как источник данных	59
Агент HP Performance как источник данных	60
Создание пакетов компонента «Сфера деятельности» и ETL	61
Установка пакетов компонента «Сфера деятельности» и ETL	62
Просмотр отчетов SHR	62
Создание источника данных для компонента ETL	63
Разработка содержимого для пакетов содержимого, включающих только компоненты приложения и ETL	64
Создание базы данных PostgreSQL	65
Создание таблиц базы данных	66
Вставка данных в базу данных	66
Часть II	67
Обзор	67
Предварительные условия	67
Определение терминов	68
Глава 6: Определение XML-модели	69
Структура XML-модели	69

Элемент XML: factTable	71
Сопоставление компонентов между XML-моделью и пространством BO	84
Примеры определения модели	85
Пример файла XML-модели	85
Ссылки на элементы в зависимом файле XML-модели	86
Задание иерархий и уровней	88
Глава 7: Определение XML политики сбора данных RTSM	90
Структура XML	90
Примеры политики сбора данных RTSM	94
Пример XML-файла политики сбора данных RTSM	94
Структура XML политики сбора данных SiteScore API	94
Глава 8: Определение XML политики сбора данных OM	97
Структура XML	97
Примеры политики сбора данных OM	100
Пример XML-файла политики сбора данных OM	100
Глава 9: Определение XML политики сбора данных OA	101
Структура XML	101
Элемент XML: etldefinition	101
Элемент XML: domain	102
Примеры политики сбора данных OA	104
Пример XML-файла политики сбора данных OA	104
Глава 10: Определение XML политики сбора данных БД	106
Структура XML	106
Примеры политики сбора данных БД	114
Пример XML-файла политики сбора данных из универсальной БД	114
Глава 11: Определение XML политики преобразования	116
Структура XML	116
Элемент XML: etldefinition	116
Элемент XML: recordSet	116
Примеры политики преобразования	118
Поворотное преобразование:	118
Использование условий для фильтрации данных	120
Использование функций	122
Глава 12: Определение XML правила сверки	123
Структура XML	123
Элемент XML: etldefinition	123
Элемент XML: rule	124
Примеры правила сверки	127
Глава 13: Определение XML правила промежуточного хранения	132
Структура XML	132
Элемент XML: stagerule	132
Элемент XML: targets	133
Примеры правила промежуточного хранения	135
Пример XML-файла правила промежуточного хранения	135
Пример объединения столбцов CSV	136
Глава 14: Определение XML потока ABC	137

Структура XML	137
Элемент XML: JobStream	137
Элемент XML: JobStreamSteps	138
Элемент XML: JobStreamLinks	141
Примеры определения потока ABC	141
Поток ETL («Извлечение, преобразование и загрузка»)	141
Поток хранилища данных	141
Глава 15: Атрибуты type и category в политиках ETL	142
Атрибуты type и category в политиках сбора данных	142
Атрибуты type и category в политике сбора данных RTSM	142
Атрибуты type и category в политике сбора данных OM	143
Атрибуты type и category в политике сбора данных OA	144
Атрибуты type и category в политике сбора данных БД	145
Атрибуты type и category в политике преобразования	145
Атрибуты type и category в правиле сверки	146
Атрибуты type и category в правиле промежуточного хранения	146
Рабочий процесс CSV-файла	147
Глава 16: Определение XML стратегии	149
Структура XML	149
Элемент XML: schema	150
Элемент XML: factTables	151
Элемент XML: dimensionTables	153
Пример определения стратегии	156
Приложение А: Фильтры и функции в политиках SHR ETL	156
Фильтры	156
Функции	157
Поддерживаемые функции в элементах Aggregate и Forecast	160
Ограничения	160
Глоссарий	161
Отправить отзыв о документации	162

Часть I

Руководство по разработке содержимого

Глава 1: Введение

В данном руководстве приведены обзор разработки содержимого программы HP Service Health Reporter (SHR) и описание процесса создания пакета содержимого. Для создания примера пакета содержимого используются файлы-образцы, имеющиеся на вашем установочном носителе SHR, и указания, приведенные в данном руководстве.

Примечание. HP представляет поддержку пакетов содержимого, созданных или измененных в CPC или иным способом в рамках стандартных соглашений о поддержке и обслуживании продуктов HP. Однако любые дефекты, обнаруженные в базовом средстве HP, устраняются HP.

Об этом руководстве

Данное руководство включает следующие разделы:

1. См. раздел [Введение](#) (эта глава): предварительные требования и справочные материалы, помогающие понять, что такое пакеты содержимого, и разобраться в терминологии разработки содержимого, используемой в данном руководстве.
2. См. раздел ["Архитектура и разработка пакетов содержимого"](#) на [странице 14](#): описание архитектуры пакетов содержимого и представление среды разработки содержимого (CPC). CPC представляет собой набор содержащихся в SHR инструментов, которые позволяют создать пакет содержимого.
3. См. раздел ["Создание пакета содержимого — начало работы"](#) на [странице 19](#): пошаговая инструкция по созданию пакета содержимого на примере пункта розничной торговли. Для создания рабочего пакета содержимого используются файлы-образцы и шаблоны, содержащиеся на носителе SHR.
4. См. раздел ["Создание пакета компонента ETL"](#) на [странице 51](#): указания и файлы-образцы для создания универсального компонента ETL пакета содержимого с ограниченными возможностями, предназначенного для примера пункта розничной торговли.
5. См. [Глоссарий](#).

Кому следует ознакомиться с ЭТИМ руководством

Это руководство предназначено для разработчиков, желающих создавать новые пакеты содержимого с помощью SHR или расширять имеющиеся пакеты содержимого.

Предварительные условия и справочная документация

Для использования данного руководства необходимо предварительно ознакомиться с указанной ниже документацией.

Предварительные требования	Справочная документация
<p>Общее руководство по использованию программы HP Service Health Reporter</p>	<p>Ознакомьтесь со следующими документами, выбрав пункт Пуск -> Программы -> HP Software -> SH Reporter -> Документация</p> <ul style="list-style-type: none"> • Общее руководство: В этом руководстве объясняются основные понятия, архитектура и типичный рабочий процесс SHR. Ознакомьтесь с этим руководством для понимания концепции и функционирования пакетов содержимого, прежде чем приступать к их разработке. • Руководство по интерактивной установке Это руководство содержит описание необходимых условий и подробное описание процедуры установки SHR. Кроме того, здесь описываются распространенные проблемы установки и методы их разрешения. • Руководство по настройке: Это руководство поможет спланировать сценарии развертывания и настроить SHR в поддерживаемых средах. • Интерактивная справка для администраторов: В этой справке содержится информация о мониторинге установленных пакетов содержимого. • Интерактивная справка для пользователей. В этой справке приведена информация о встроенных пакетах содержимого SHR.
<p>Концепция хранилища данных</p>	<p>Описание концепции и примеры хранилищ данных можно найти в Интернете.</p>
<p>Основы отчетности в SAP BusinessObjects</p>	<ul style="list-style-type: none"> • SAP BusinessObjects Enterprise InfoView User's Guide: это руководство доступно в следующем разделе Пуск -> Программы -> BusinessObjects > BusinessObjects Enterprise -> Documentation. В нем приведены указания по созданию и работе с отчетами Web Intelligence. • Интерактивная справка SAP BusinessObjects Universe Designer. В этой справке вы найдете информацию о создании, построении и

Предварительные требования	Справочная документация
	<p>управлении системами Universe. Открыть справку можно в пользовательском интерфейсе Universe Designer.</p> <p>Дополнительные сведения и последнюю версию справочной документации см. по адресу http://help.sap.com/businessobject/product_guides/.</p>
Основы XML и создание документов XML	Основы и примеры документов XML можно найти в Интернете. SHR не содержит рекомендаций по использованию какого-либо конкретного ресурса.

Глава 2: Архитектура и разработка пакетов содержимого

SHR позволяет создавать на платформе базы данных управления производительностью следующее содержимое:

- **Пакет содержимого:** создание новых и расширение встроенных пакетов содержимого SHR. В этом руководстве приведено описание этапов создания пакета содержимого с использованием примера.
- **Отчеты Web Intelligence:** использование интерфейса приложения SAP BusinessObjects InfoView для создания новых отчетов и настройки встроенных отчетов SHR.

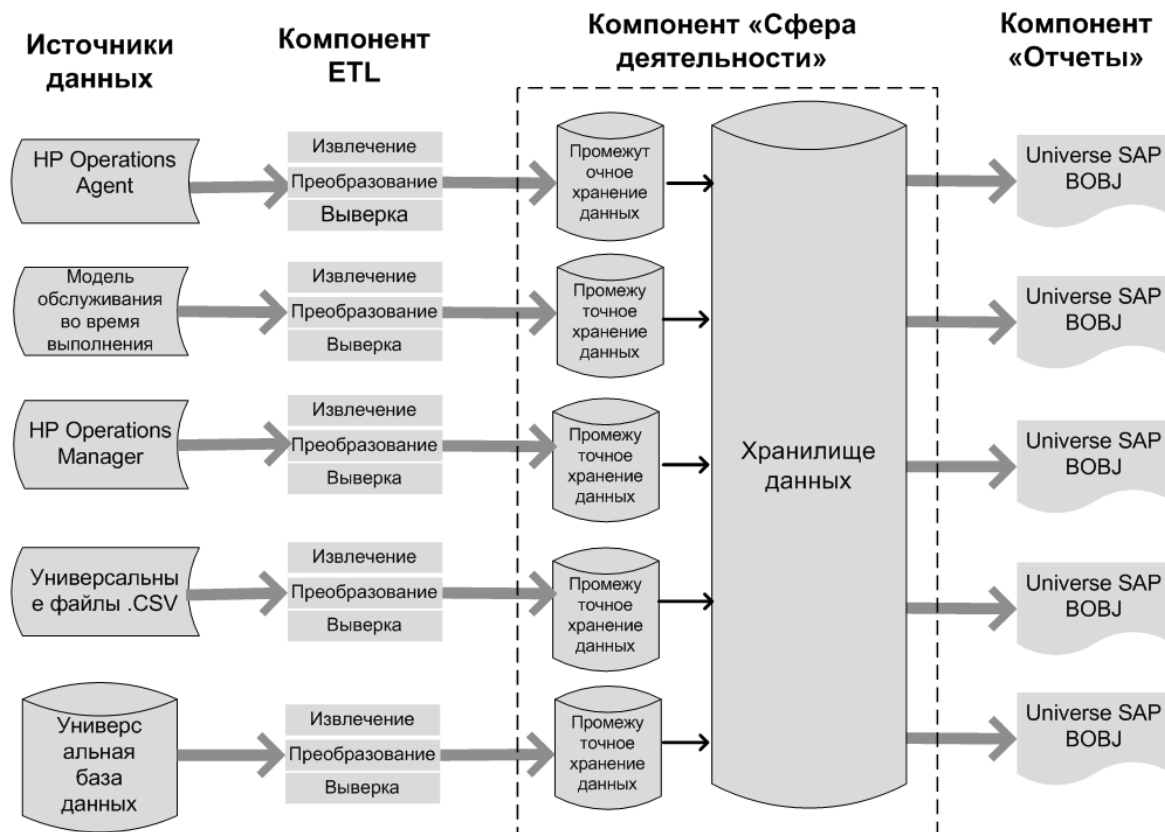
Архитектура пакетов содержимого

Пакет содержимого — это киоск данных для определенной сферы деятельности или области применения, развернутый на платформе базы данных управления производительностью SHR. Пакеты содержимого используются для определения собираемых метрик и способов их обработки и хранения, а также для представления обработанных данных в отчетах.

Компоненты пакета содержимого

Как правило, пакет содержимого состоит из трех компонентов: «Сфера деятельности» «Извлечение, преобразование и загрузка» (ETL) и «Отчеты».

На приведенной ниже диаграмме показаны типичные потоки данных между компонентами.



Примечание. Этапы преобразования и вверки в компоненте ETL не являются обязательными и могут относиться не ко всем источникам данных.

Компонент «Сфера деятельности»

Компонент «Сфера деятельности» определяет модель данных той области деятельности, о которой составляется отчетность, а также логику обработки этих данных. Он требует моделирования этих данных экспертами в этой сфере в соответствии с потребностями предприятия. Этот компонент не зависит от источника данных. Компонент «Сфера деятельности» включает следующие элементы:

- Модель данных, включающая факты и измерения сферы деятельности, о которой составляется отчетность, и отношение между ними.
- Поток рабочих процессов, обеспечивающие управление и мониторинг обработки данных. Поток состоит из этапов, которые связаны друг с другом последовательными отношениями. Пакет содержимого содержит набор потоков рабочих процессов, которые определяют и обеспечивают управление потоком данных от одного этапа к другому. В компоненте «Сфера деятельности» пакета содержимого потоки рабочих процессов определены в файлах XML, предназначенных для загрузки данных в таблицы и автономного агрегирования данных.
- Измерения и кубы бизнес-представления, которое будет использоваться одним или несколькими компонентами «Отчеты» (факультативно).

Извлечение, преобразование и загрузка компонента (ETL)

Компонент ETL зависит от источника данных: он определяет сбор данных из указанного источника данных и их последующее преобразование и загрузку в хранилище данных. Поэтому в каждой области анализа для каждого приложения источника данных используется отдельный компонент ETL пакета содержимого. Прежде чем приступить к созданию компонента ETL, необходимо указать источник данных, предоставляющий метрики, пригодные для ввода в модель данных сферы деятельности. Компонент ETL включает следующие элементы:

- **Правила сбора (извлечения) данных:** после указания источника данных необходимо создать программу-сборщик или использовать имеющуюся программу-сборщик для сбора необходимых фактов и измерений из источника данных. Для определения метрик, собираемых программой-сборщиком из источника данных, политика сбора данных должна быть написана на языке XML. Программа-сборщик собирает данные в соответствии с политикой сбора данных и помещает их в файлы `csv`.
SHR поддерживает сбор данных из набора известных источников данных и содержит программы-сборщики для каждого из таких источников данных.
SHR поддерживает следующие источники данных:
 - модель обслуживания во время выполнения (RTSM);
 - Агент HP Operations Agent;
 - HP Operations Manager;
 - база данных профилей HP Business Service Management;
 - универсальные файлы `.CSV`;
 - базы данных с поддержкой JDBC.
- **Правила преобразования данных (факультативно):** правила преобразования данных требуются в том случае, если собранные данные, например в виде файлов `.csv`, необходимо преобразовать перед их загрузкой в хранилище данных. Например, можно написать правило для удаления строк, содержащих пустое значение в столбце «host name» («Имя хоста»). Для написания правил преобразования используются файлы XML. Для готовых правил преобразования SHR содержит служебную программу преобразования данных под названием «mapper».
- **Правила выверки данных (факультативно):** выверка данных представляет собой процесс установления связей между данными фактов и соответствующими данными измерений. В SHR написание правил выверки данных используется для установления связей между данными фактов из одного источника и соответствующими данными измерений из другого источника с помощью распространенных бизнес-ключей. Например, в среде моста служб и операций (SaOB) источником данных об измерениях является RTSM, а источником данных о фактах — HP Operations Agent. Для выверки данных о фактах и данных об измерениях написание правил выверки осуществляется с использованием XML.
- **Правила промежуточного хранения данных:** после сбора, преобразования и выверки данных (в форме файлов `csv`) они переносятся в таблицы промежуточного хранения.

Правила промежуточного хранения данных определяют способ перемещения данных в таблицы промежуточного хранения, включая процесс слияния столбцов и строк.

- **Определение потоков рабочих процессов:** в компоненте ETL потоки рабочих процессов определяются в формате XML для управления перемещением данных от сбора к промежуточному хранению с использованием, если необходимо, промежуточных этапов преобразования и выверки.

Компонент «Отчеты»

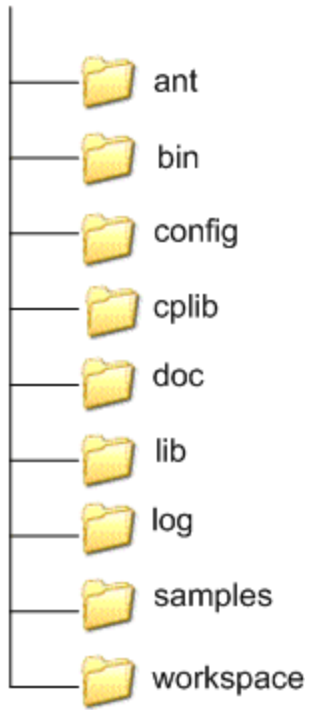
Компонент «Отчеты» содержит отчеты и системы Universe Web Intelligence программы SAP BusinessObjects. Система Universe пакета содержимого обеспечивает ориентированное на решение деловых задач значимое представление соответствующей комплексной базы данных и упрощает создание отчетов. Это логическое представление базовой модели данных, определяемой в компоненте «Сфера деятельности». С помощью компонента «Отчеты» осуществляется импорт измерений и кубов, определенных в соответствующем компоненте «Сфера деятельности».

Представление среды разработки содержимого

Среда разработки содержимого (CPC) включает набор инструментов, которые используются в процессе разработки содержимого. В этих инструментах используются файлы XML, написанные разработчиком пакета содержимого для создания пакетов компонентов пакета содержимого, предназначенных для установки.

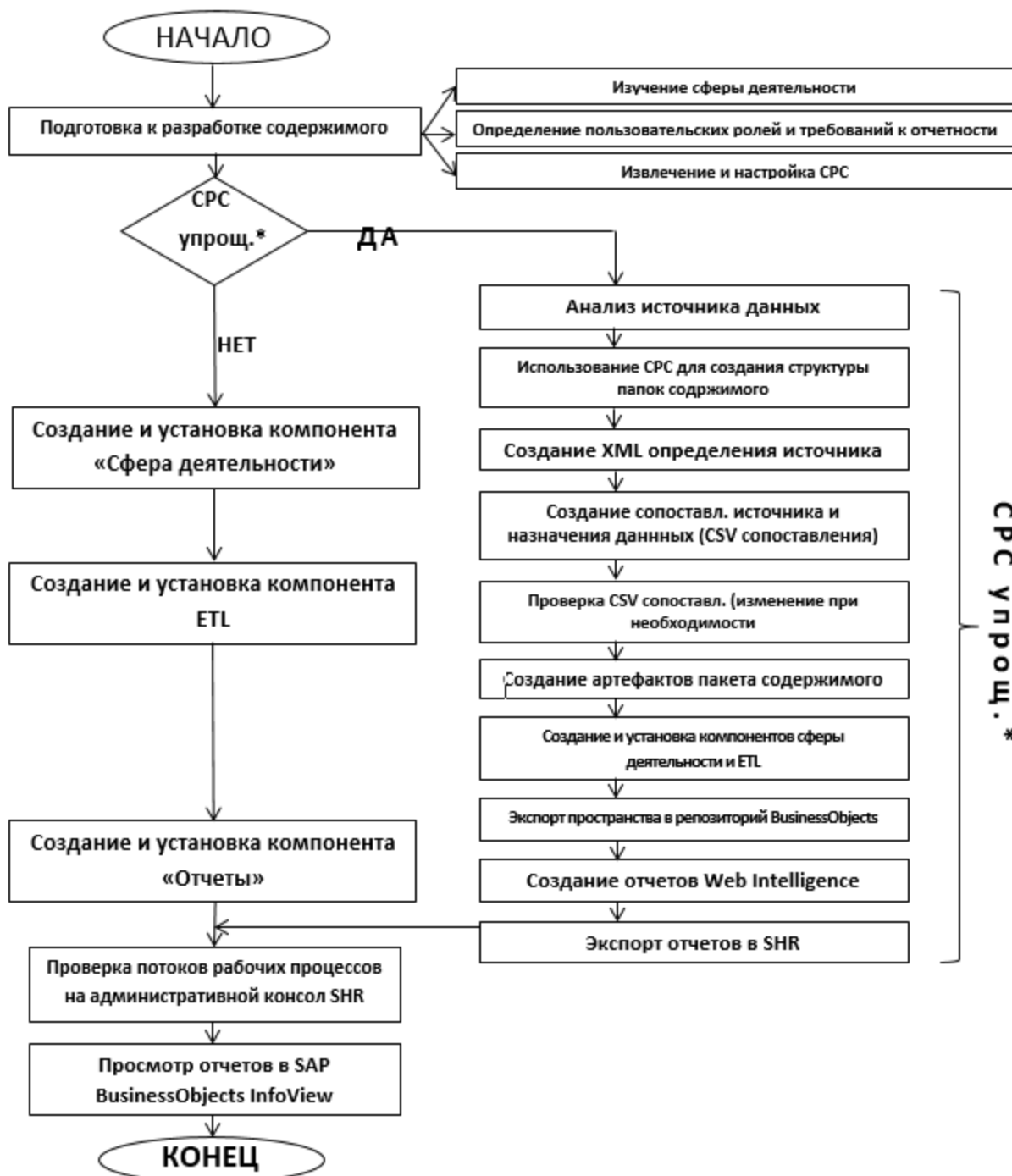
Инструменты CPC содержатся на носителе SHR в виде самоизвлекающегося файла CDE.exe, который находится в следующей папке: <каталог_установки>. После извлечения содержимого файла CPC создает структуру папок, показанную на приведенном ниже рисунке.

CDE



Глава 3: Создание пакета содержимого — начало работы

В этой главе описаны этапы создания пакета содержимого на практическом примере из области розничной торговли. На приведенной ниже схеме показана типичная процедура создания пакета содержимого, а также упрощенная процедура. Чтобы оценить применимость упрощенной процедуры, ознакомьтесь с разделом "[Когда следует использовать упрощенную СРС?](#)" на [странице 22](#)".



Упрощенная CPC* — см. раздел «Использование упрощенной CPC», чтобы определить применимость этого метода.

Подготовка к разработке содержимого

Изучение сферы деятельности

Рассмотрим пример крупной сети розничной торговли электроникой, в которой:

- Имеется 100 магазинов, расположенных в 5 населенных пунктах, и
- на полках каждого магазина имеется около 10 тыс. отдельных товаров, обозначаемых товарными позициями (SKU).

Розничная сеть является автоматизированной, и на каждом изделии имеется сканируемый ярлык. Сбор данных о транзакциях осуществляется главным образом системой кассовых терминалов на выходе из магазина, где штрих-коды сканируются и непосредственно вводятся в систему. Здесь определяется количество купленных товаров.

Ознакомившись с характером деятельности сети розничной торговли, можно переходить к определению пользовательских ролей предприятия и соответствующих требований к отчетности.

Определение пользовательских ролей и требований к отчетности

В примере с пунктом розничной торговли в качестве пользователей отчетов рассматривается управленческий персонал предприятия.

Пользователей-управленцев интересует сводный отчет о продажах, в котором информация о реализации товаров представлена по различным товарным категориям и магазинам, находящимся в различных населенных пунктах. Информация о продажах должна быть представлена по различным периодам времени, например за год, квартал, месяц и день.

На этом этапе рекомендуется создать дизайн-макет требуемых отчетов. Первоначальный дизайн-макет можно создать на бумаге или с помощью любого инструмента разработки по вашему выбору. Разработку самих отчетов Web Intelligence в системе SAP BusinessObjects можно осуществить после создания пакета компонента «Отчеты».

Извлечение и настройка CPC

1. Войдите в систему, в которой установлено приложение SHR. В каталоге `<installation_directory>` вы найдете приложение `CDE.exe`.
2. Запустите файл `CDE.exe`, чтобы извлечь средства CPC в папку **CDE**.

Примечание. Укажите `<installation_directory>` в кавычках. Кроме того, создание пакетов содержимого с помощью инструментов в папке CDE может закончиться неудачей, если `<installation_directory>` включает пробелы.

3. Введите **cmd** и нажмите клавишу **ВВОД**, чтобы открыть диалоговое окно командной строки.
4. Введите в командной строке и выполните команду `cd` для перехода в каталог, где установлена программа SHR. Затем выполните следующие команды для смены каталога:

- a. `<installation_directory>cd CDE` для перехода в папку `<installation_directory>\CDE`.
- b. `<installation_directory>\CDE> cd bin` для перехода в папку `<installation_directory>\CDE\bin`.

Здесь `<installation_directory>`— это каталог, в котором установлено ПО SHR.

5. Выполните следующую команду:

```
setenv.bat
```

Для этого пути задаются следующие переменные среды:

```
CDE_HOME
```

```
ANT_HOME
```

```
JRE_HOME
```

Когда следует использовать упрощенную СРС?

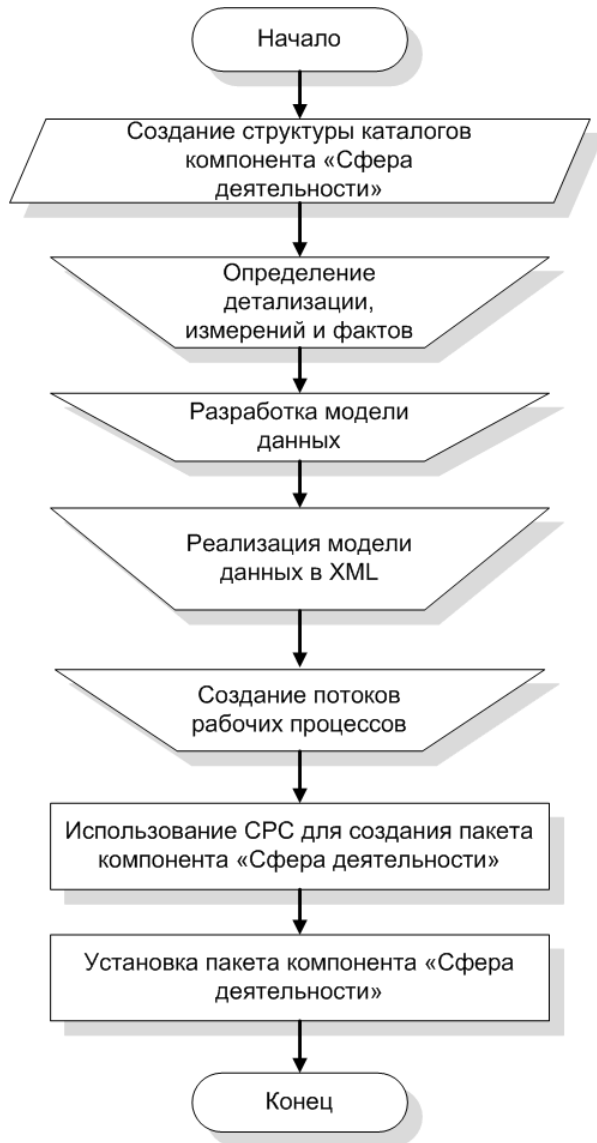
Упрощенный метод создания пакетов содержимого можно использовать в следующих случаях:

1. Когда таблица фактов связана с одним или несколькими измерениями (схема «звезда») и эти измерения нормализованы (не имеют родительской таблицы).
2. Когда все факты должны быть выверены хостом (если источником данных является агент HP Operations Agent).
3. Когда созданные отчеты не требуют сведения или детализации данных.

См. описание упрощенной процедуры создания СРС в разделе ["Создание пакета содержимого — упрощенный метод"](#) на странице 58. В противном случае используйте стандартный метод разработки содержимого в разделе ["Создание пакета содержимого — начало работы"](#) на странице 19".

Создание и установка компонента «Сфера деятельности»

На приведенной ниже схеме показаны типичные этапы создания компонента «Сфера деятельности» пакета содержимого.



Создание структуры каталогов

Чтобы создать структуру каталогов для исходных файлов компонента «Сфера деятельности», введите в командной строке и выполните следующую команду:

```
<installation_directory>\CDE\bin>CreateCPFolders.bat -package RetailPOS -subpackage RetailPOSDomain -type domain
```

Здесь:

- *<installation_directory>* — это каталог, в который выполнена установка SHR
- RetailPOS — это имя разрабатываемого пакета содержимого
- RetailPOSDomain — это имя компонента «Сфера деятельности» в RetailPOS

Каталог будет содержать шаблоны программы SHR, которые вы будете использовать для создания исходных файлов компонента «Сфера деятельности»: файлы XML-модели и XML-файлы потоков рабочих процессов.

SHR содержит образцы исходных файлов для пакета содержимого пункта розничной торговли, которые размещены в указанной ниже папке. Эти файлы можно использовать для справки при создании своего собственного пакета содержимого.

`%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\`

определение детализации, измерений и фактов

Наибольшей детализацией характеризуются данные в таблице фактов. В примере с пунктом розничной торговли уровень детализации данных соответствует отдельной позиции в транзакции кассового терминала.

В качестве измерений используются:

- Дата
- товар;
- магазин

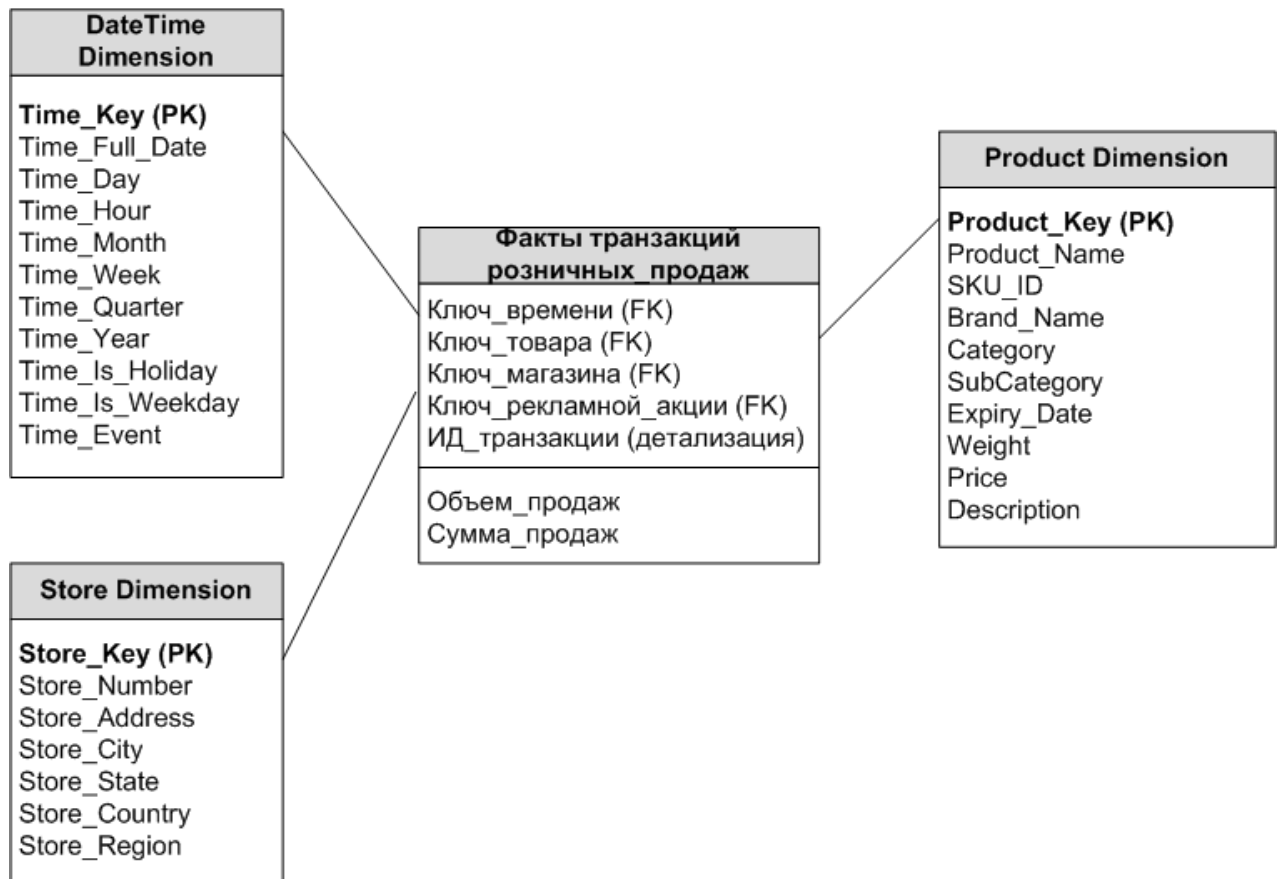
Фактами, сбор которых осуществляется системой кассовых терминалов, являются:

- количество реализованных товаров;
- сумма реализации.

разработка модели данных

Модель данных отражает отношения между объектами (таблицами фактов и измерений) и их атрибутами (столбцами таблиц в базе данных). В SHR моделью данных является файл XML. Чтобы создать модель данных, начните с создания схематической диаграммы, затем реализуйте ее в XML.

Ниже изображена схематическая диаграмма, которая будет использоваться для создания модели данных пакета содержимого RetailPOS.



Реализация модели данных в XML

Разработанную в предыдущем этапе схему необходимо реализовать в XML для использования средой разработки содержимого при создании пакета компонента «Сфера деятельности». Этот XML-файл называется XML-моделью.

XML-модель, как правило, состоит из следующих разделов:

- **Реляционный раздел**, в котором определены таблицы фактов и таблицы измерений и отношения между фактами и измерениями.
- **Логический раздел**, в котором определены кубы, иерархии и уровни. Для каждой таблицы фактов определяется один куб.
- **Раздел агрегирования**, в котором определено, какое агрегирование будет осуществляться в исходных таблицах.

После создания структуры каталогов компонента «Сфера деятельности» найдите шаблон `model_template.xml` в следующей папке:

```

%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap\source\model

```

Этот файл `.xml` можно редактировать для создания XML-модели.

Кроме того, для справки можно использовать файл-образец XML-модели компонента `RetailPOSDomain`, размещенный в указанной ниже папке:

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS\RetailPOSDomain.ap\source\model
```

Чтобы ознакомиться с содержимым образца XML-файла модели с политикой сбора данных, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите файл RetailPOS_dimension_model.xml. Дважды щелкните, чтобы открыть файл XML в окне браузера.

создание потоков рабочих процессов

SHR предоставляет платформу рабочих процессов для управления и мониторинга рабочих процессов обработки данных. Эта платформа состоит из потоков рабочих процессов. Поток состоит из этапов, которые связаны друг с другом последовательными отношениями. Каждый пакет содержимого включает набор потоков, которые определяют и обеспечивают управление движением данных от одного этапа к другому.

Например, поток рабочих процессов компонента «Сфера деятельности» может включать следующие этапы:

Загрузка данных в таблицу норм -> агрегирование по часам -> агрегирование по дням

Потоки рабочих процессов необходимо реализовать в XML для использования средой разработки содержимого при создании пакета компонента «Сфера деятельности». С помощью XML необходимо создать следующие потоки:

- Один XML-поток рабочих процессов для загрузки и агрегирования фактов.
- Один XML-поток рабочих процессов для загрузки измерения Магазинов.
- Один XML-поток рабочих процессов для загрузки измерения Товаров.

После создания структуры каталогов компонента «Сфера деятельности» найдите шаблон потока рабочих процессов ABC_stream_template.xml в следующей папке:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap\source\orchestration
```

Шаблон ABC_stream_template.xml можно редактировать для создания XML-файлов потоков рабочих процессов.

Кроме того, для справки можно использовать образцы XML-файлов потоков рабочих процессов компонента **RetailPOSDomain**, размещенные в указанной ниже папке:

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\
```

Чтобы ознакомиться с содержимым XML-файлов потоков рабочих процессов, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите следующие файлы XML:

- Реализация модели данных в XML: XML-поток рабочих процессов для загрузки и агрегирования фактов;
- Dimension_Store_stream.xml: XML-поток рабочих процессов для загрузки измерения Магазинов.
- Dimension_Product_stream.xml: XML-поток рабочих процессов для загрузки измерения Товаров.

Дважды щелкните для открытия файлов XML в окне браузера.

Использование CPCS для создания пакета компонента «Сфера деятельности»

Чтобы создать пакет компонента «Сфера деятельности», выполните следующие действия:

1. Создайте XML-файл манифеста.

XML-файл манифеста содержит определения исходных файлов, предназначенных для использования средой разработки содержимого при создании пакета компонента «Сфера деятельности».

Чтобы создать XML-файл манифеста с помощью CPCS, выполните следующие действия:

- a. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap
```

- b. Выполните следующую команду:

```
ant createManifestTemplate
```

XML-файл манифеста с именем `RetailPOSDomain_manifest_template.xml` создается в папке `%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap\source`

В указанной ниже папке размещен образец XML-файла манифеста для компонента `RetailPOSDomain`.

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS\RetailPOSDomain.ap\source\RetailPOSDomain_manifest_template.xml
```

Чтобы ознакомиться с содержимым образца XML-файла манифеста с политикой сбора данных, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите файл `создается в папке`. Дважды щелкните, чтобы открыть файл XML в окне браузера.

2. Создайте пакет компонента «Сфера деятельности».

Чтобы создать пакет компонента «Сфера деятельности» с помощью CPCS, выполните следующие действия:

- a. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap
```

- b. Выполните следующую команду:

```
ant
```

Пакет компонента «Сфера деятельности», предназначенный для установки, создается в следующем каталоге:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap\dist
```

3. Перейдите в каталог `%CDE_HOME%\workspace\RetailPOS\RetailPOSDomainCP.ap\dist`
4. Скопируйте `RetailPOS` в каталог `%PMDV_HOME%\packages`. Таким образом пакет становится доступен для установки в диспетчере развертывания на административной консоли.

Примечание. Нет необходимости в том, чтобы приложение SHR было установлено на компьютере, на котором вы создаете пакет компонента «Сфера деятельности». Если вы создали пакет компонента «Сфера деятельности» на другом компьютере, необходимо скопировать его на компьютер с SHR в папку `%PMDB_HOME%\packages`.

Установка пакета компонента «Сфера деятельности»:

Для установки пакетов компонентов пакета содержимого в SHR используется диспетчер развертывания, доступный на административной консоли. См. указания по установке компонентов пакета содержимого в *Руководстве по установке и настройке HP Service Health Reporter*.

Создание и установка компонента ETL

Компонент ETL включает правила сбора, преобразования, выверки и промежуточного хранения данных. Создание полного компонента ETL, содержащего все указанные правила, может являться достаточно сложной задачей. Поэтому, чтобы помочь вам в создании примера пакета содержимого, в этой главе описан простой альтернативный способ создания данных в форме файлов `.csv` и их загрузки в хранилище данных.

В следующей главе "[Создание пакета компонента ETL](#)" на [странице 51](#) описано создание пакета компонента ETL, который можно установить с помощью диспетчера развертывания, доступного на административной консоли.

Настраиваемая загрузка данных с использованием файлов CSV

В соответствии с этим способом создается набор файлов CSV в нужном формате, которые помещаются в таблицы хранилища данных SHR.

`%PMDB_HOME%\stage (Windows)`

`$PMDB_HOME/stage (Linux)`

Для создания и загрузки файлов `.csv` выполните следующие действия:

1. **Установите пакет компонента «Сфера деятельности»:** прежде чем приступить к созданию предназначенных для загрузки файлов `.csv`, убедитесь в наличии пакета компонента «Сфера деятельности», созданного в соответствии с указаниями в "[Установка пакета компонента «Сфера деятельности»:](#)" [выши](#). С помощью компонента «Сфера деятельности» создается файл `html stage interface`, который содержит формат, в котором должны создаваться файлы `.csv`. Файл `html stage interface` создается в следующей папке.

`%PMDB_HOME%\packages\RetailPOS\RetailPOSDomainCP.ap\doc (Windows)`

`$PMDB_HOME/packages/RetailPOS/RetailPOSDomainCP.ap/doc (Linux)`

См. для справки файл-образец `RetailPOSDomain_INTERFACE.html` в каталоге `%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS\RetailPOSDomain.ap\doc`

(Windows) или `$CDE_HOME/samples/RetailPOS_Demo_Content_Pack/RetailPOS/RetailPOSDomain.ap/doc` (Linux).

2. **Создайте файлы csv:** SHR содержит простую программу — генератор файлов csv для создания образцов файлов csv для пакета содержимого пункта розничной торговли (RetailPOS). Чтобы создать с помощью программы — генератора файлов .csv файлы .csv с заданным форматом, содержащимся в шаблоне .html, выполните следующие действия:

- a. Перейдите к папке:
 - `%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Sample CSV Generator.` (Windows)
 - `$CDE_HOME/samples/RetailPOS_Demo_Content_Pack/Sample CSV Generator.` (Linux)
- b. Скопируйте следующие файлы в указанное место, как показано в приведенной ниже таблице.

Файл	Скопировать в папку
retailpos_csvgen.exe	<code>%PMDB_HOME%\bin</code> (Windows) <code>\$PMDB_HOME\bin</code> (Linux)
retailpos_csvgen.ini	<code>%PMDB_HOME%\config\startup</code> (Windows) <code>\$PMDB_HOME\config\startup</code> (Linux)
retailposcsvgen.jar	<code>%PMDB_HOME%\lib</code> (Windows) <code>\$PMDB_HOME\lib</code> (Linux)

- c. Для Windows запустите файл `retailpos_csvgen.exe` в командной строке. Для Linux перейдите в каталог `$PMDB_HOME\bin` и запустите файл `ln -s launcher retailpos_csvgen` в командной строке.

Для создаваемого примера пакета содержимого RetailPOS программа-генератор файлов .csv создает файлы .csv за два месяца и помещает их в папку `%PMDB_HOME%\stage` (Windows) или `$PMDB_HOME/stage` (Linux). Ранее установленный вами компонент «Сфера деятельности» загружает файлы CSV в таблицы хранилища данных SHR.

Проверка потоков рабочих процессов на административной консоли

После запуска службы `HP_PMDB_Platform_Timer` необходимо войти в административную консоль, чтобы проверить статус потоков рабочих процессов компонента «Сфера деятельности». Выполните следующие действия:

1. В интерфейсе администрирования выберите пункт **Внутренний мониторинг > Обработка данных**.
2. На вкладке **Сведения о потоках** просмотрите статус потоков пакета содержимого RetailPOSDomain. В случае успешного завершения у всех потоков будет указан статус ОК.

В примере с пунктом розничной торговли компонент «Сфера деятельности» содержит следующие потоки рабочих процессов, при этом каждый поток содержит один или несколько этапов.

- XML-поток рабочих процессов для загрузки и агрегирования факта Retail_Sales.
- XML-поток рабочих процессов для загрузки измерения Магазин.
- XML-поток рабочих процессов для загрузки измерения Товар.
- XML-поток рабочих процессов для загрузки измерения Рекламная акция.

Как показано на приведенном ниже изображении, успешное завершение потоков обозначается зеленым цветом.

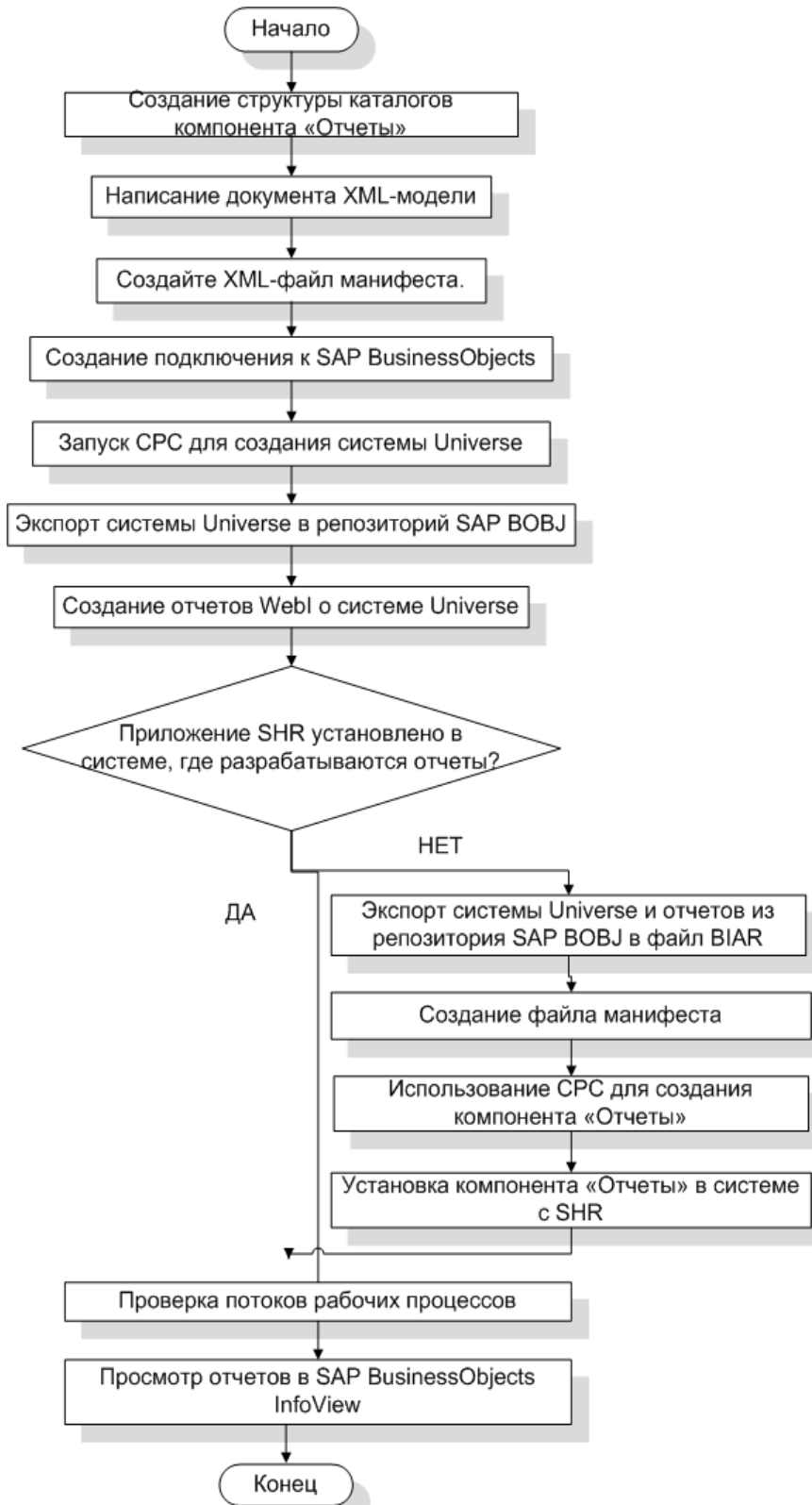
Обработка данных

Обработка данных					
		Сведения о потоке		Хронологический обзор потока	
		Хронологические сведения о потоке			
Имя компонента пакета содержимого	Число потоков	Сведения о статусе потока			
		ОК	Предупреждение	Ошибка	Всего
Core	26	26	0	0	26
RetailPOSDomain	5	5	0	0	5
ETL_SM_VI_SoI_Zones_PA	0	0	0	0	0
ETL_SystemManagement_SIS	0	0	0	0	0
SystemManagement	0	0	0	0	0
MSAppCore	1	1	0	0	1

Сведения о потоке для компонента пакета содержимого : RetailPOSDomain			
Имя потока	Статус шага (Выполнен/Всего)	Статус шага	Время начала
RetailPOSDomain@Product	1/1	SUCCESS	01.11.2012 14:50:14
RetailPOSDomain@Store	1/1	SUCCESS	01.11.2012 16:10:13
RetailPOSDomain@Downtime	1/3	SUCCESS	01.11.2012 14:50:14
RetailPOSDomain@Retail_Sales	1/4	SUCCESS	01.11.2012 16:10:13
RetailPOSDomain@Promotion	1/1	SUCCESS	01.11.2012 16:10:12

Создание и установка компонента «Отчеты»

На приведенной ниже схеме показаны этапы создания компонента «Отчеты».



Предварительные условия

Перед созданием компонента «Отчеты» необходимо убедиться в следующем:

- СРС установлена на том же компьютере, где вы установили SHR и SAP BusinessObjects.
- Компонент «Сфера деятельности», созданный в "[Установка пакета компонента «Сфера деятельности»](#):" на [странице 28](#), установлен. Для установки компонента «Сфера деятельности» используйте диспетчер развертывания на административной консоли. См. инструкции в *интерактивной справке для администраторов HP Service Health Reporter*.

Создание структуры каталогов

Чтобы создать структуру каталогов для исходных файлов компонента «Отчеты», введите в командной строке и выполните следующую команду:

```
<installation_directory>\CDE\bin> CreateCPFolders.bat -package RetailPOS -subpackage RetailPOSApp -type application
```

Здесь:

- *<installation_directory>* — это каталог, в который выполнена установка SHR
- RetailPOS — это имя разрабатываемого пакета содержимого
- RetailPOSReporting — это имя компонента отчетов в RetailPOS

Каталог будет содержать шаблоны программы SHR, которые вы будете использовать для создания исходных файлов компонента «Отчеты».

SHR содержит образцы исходных файлов для пакета содержимого пункта розничной торговли, которые размещены в указанной ниже папке. Эти файлы можно использовать для справки при создании своего собственного пакета содержимого.

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\
```

Написание документа XML-модели

Документ XML-модели компонента «Отчеты» должен содержать логический раздел. В этом разделе необходимо указать ссылку на куб, который вы определили в документе XML-модели компонента «Сфера деятельности».

Для справки в указанной ниже папке размещен файл-образец XML-модели для компонента RetailPOSReporting.

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS\RetailPOSReporting.ap\source\model
```

Чтобы ознакомиться с содержимым образца XML-файла модели с политикой сбора данных, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите файл RetailPOSDomain_manifest_template.xml. Дважды щелкните, чтобы открыть файл XML в окне браузера.

Создание XML-файла манифеста

XML-файл манифеста содержит определение документа XML-модели, созданного в этапе «[Написание документа XML-модели](#)» на предыдущей странице» для использования средой разработки содержимого.

Чтобы создать XML-файл манифеста с помощью CPCS, выполните следующие действия:

1. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap
```

2. Выполните следующую команду:

```
antcreateManifestTemplate
```

XML-файл манифеста с именем `RetailPOS_manifest_template.xml` создается в папке `%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap\source`

В указанной ниже папке размещен образец XML-файла манифеста для компонента `RetailPOSReporting`.

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS\RetailPOSReporting.ap\source
```

Создание подключения к SAP BusinessObjects

Для создания системы `SAP BusinessObjects Universe` необходимо создать безопасное подключение базы данных к программе `SAP BusinessObjects` с помощью приложения `Universe Designer`. `SHR` содержит пакетный сценарий, предназначенный для создания этого соединения. Чтобы создать подключение с помощью этого сценария, выполните следующие действия:

1. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\bin
```

2. Введите в командной строке и выполните следующую команду:

```
setenv.bat
```

3. Введите в командной строке и выполните следующий пакетный сценарий:

```
updateCDEProperty.bat
```

Примечание. Сценарий `updateCDEProperty.bat` запрашивает следующие данные:

- Имя хоста (полное доменное имя) системы `SAP BusinessObjects`
- Имя пользователя `SAP BusinessObjects Designer Studio` (значение по умолчанию: `Administrator`)
- Пароль для пользователя выше (по умолчанию: пустой пароль)

```
createUniverseConnection.bat
```

На экране появится следующее сообщение об успешном создании подключения: `Default BO Universe connection ("MA") was created successfully`.

Примечание. Имя подключения должно быть уникальным для всех компонентов пакета содержимого.

Чтобы обеспечить уникальность имени подключения, используйте шаблон SHR_<content_pack_name>. Вы можете изменить имя подключения, отредактировав параметр `bo.connection` в файле `cde.properties` в каталоге `$CDE/config`.

Использование CPC для создания системы SAP BusinessObjects Universe

Чтобы создать систему Universe с помощью CPC, введите в командную строку:

1. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap
```

2. Выполните следующую команду:

```
ant
```

Система Universe создается с расширением имени файла `.unv` и помещается в следующую папку:

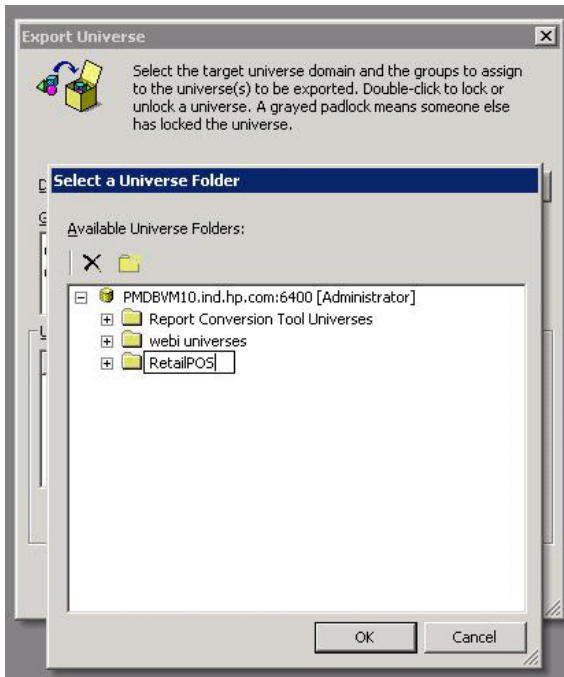
```
%CDE_
HOME%
\workspace\RetailPoS\RetailPoSReporting.ap\dist\RetailPOS\RetailPOSReporting.ap
```

Совет. Систему Universe можно также редактировать, добавляя в нее дополнительные иерархии. См. дополнительную информацию в *интерактивной справке приложения SAP BusinessObjects Universe Designer*.

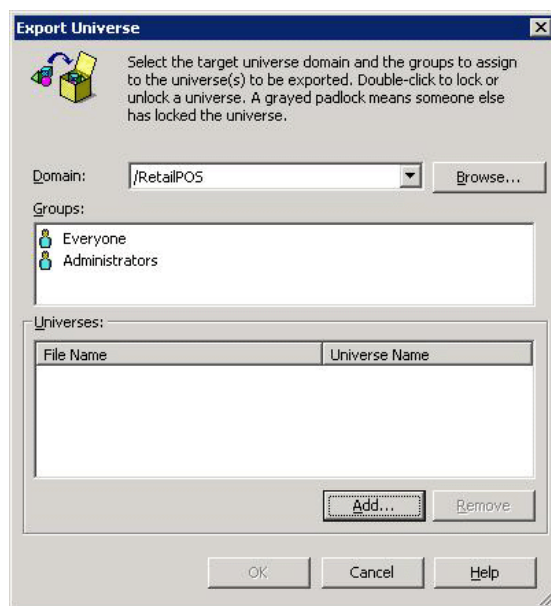
Экспорт системы Universe в репозиторий SAP BusinessObjects

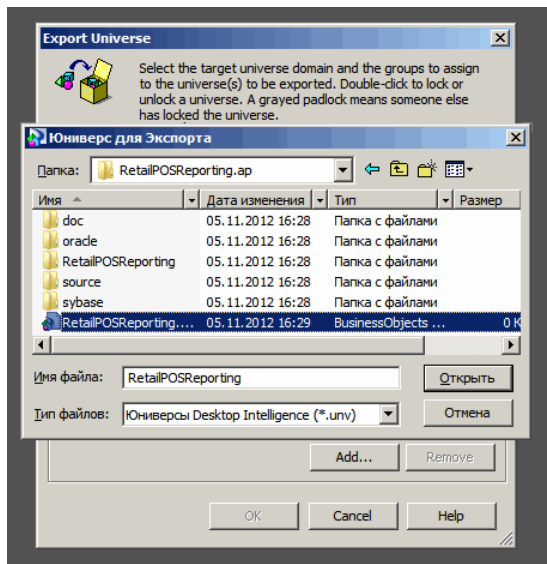
В приложении SAP BusinessObjects Universe Designer выполните следующие действия:

1. Выберите **File->Export**. Откроется окно «Export Universe».
2. Выберите из списка соответствующую папку системы Universe, в данном примере — RetailPOS. Нажмите кнопку **OK**.

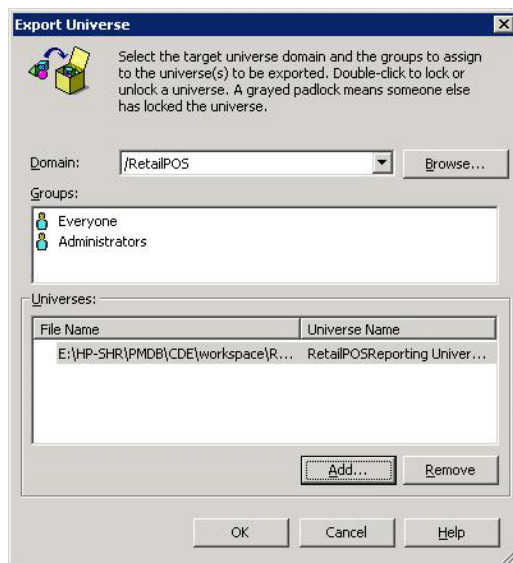


3. В окне Export Universe перейдите в папку, где находится система Universe, в данном примере — RetailPOSReporting.unv. Выберите **Open**.





В окне Export Universe показано, что файл **RetailPOSReporting.unv** добавлен к списку экспортируемых систем Universe. Нажмите кнопку **OK**.



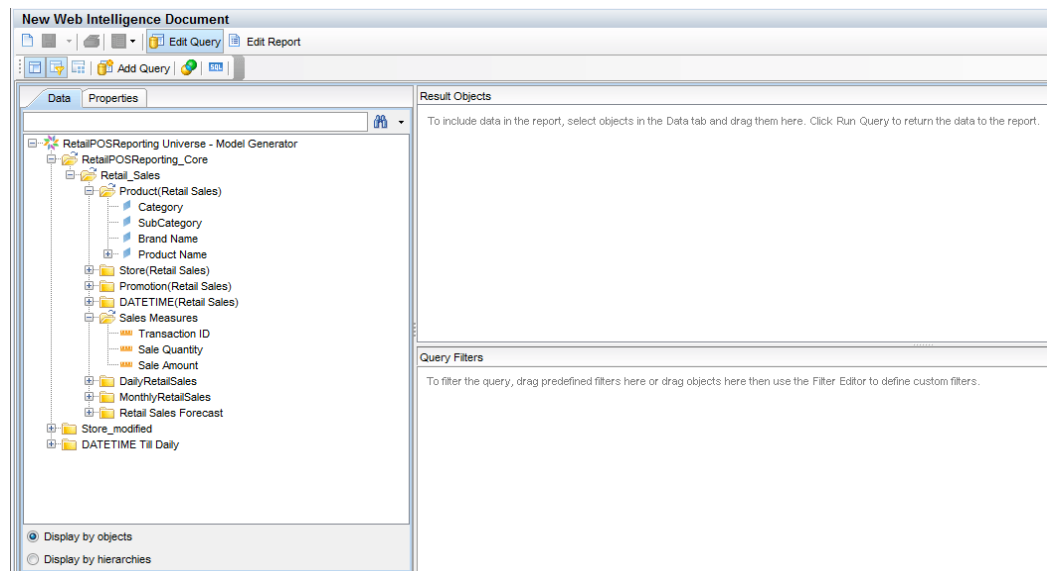
4. Откроется сообщение universe successfully exported.

СОЗДАНИЕ ОТЧЕТОВ Web Intelligence

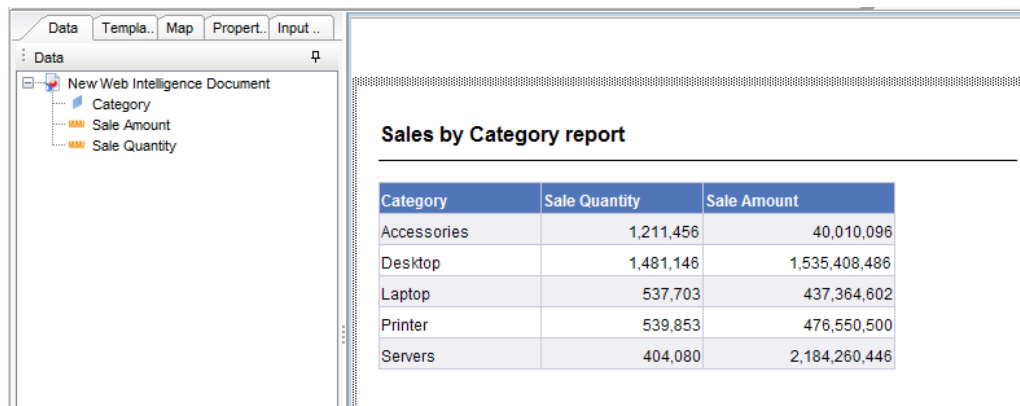
Вы можете создавать отчеты Web Intelligence, выбирая соответствующую систему Universe в интерфейсе SAP BusinessObjects InfoView и создавая один или несколько запросов для определения данных, которые должны содержаться в отчетах.

Чтобы создать простой отчет о продажах, содержащий таблицу с данными об объеме и сумме продаж по товарным категориям, выполните следующие действия:

1. Войдите в интерфейс SAP BusinessObjects InfoView одним из следующим способом:
 - В адресной строке веб-браузера введите URL-адрес системы SAP BusinessObjects. URL-адрес компьютера имеет следующий формат: `http://<имя хоста или IP-адрес>:PORT NO (8080)/InfoViewApp/logon.jsp`.
 - На административной консоли выберите пункты **Администрирование ->SAP BOBJ**, затем нажмите **Запустить InfoView**. Откроется страница входа в InfoView.
2. Нажмите **Document List**.
3. Нажмите **New ->Web Intelligence Document**. Отобразится список систем Universe приложения BusinessObjects.
4. Выберите **RetailPOSReporting Universe - Model Generator**.Откроется окно New Web Intelligence Document. На вкладке Data показаны объекты — доступные в системе Universe измерения и показатели, как показано на следующем изображении.



5. Для включения данных в отчет выберите следующие объекты на вкладке Data и перетащите их в окно Result Objects. Вы можете также поместить эти объекты в окно Result Objects, дважды нажав на них кнопкой мыши.
 - Измерение: Category (в разделе Product(Retail Sales))
 - Показатели:
 - Sale Quantity (в разделе Sales Measures)
 - Sale Amount (в разделе Sales Measures)
6. Чтобы ввести данные в отчет, нажмите кнопку **Run Query**.
Будет создана таблица с данными об объеме и сумме продаж по товарным категориям. Вы можете переименовать таблицу по своему усмотрению.



Включение функции детализации по времени в отчетах

Чтобы включить в отчете функцию детализации по времени, нажмите кнопку **Drill** на панели инструментов InfoView. Вы можете выполнять детализацию и сведение данных по измерению товарной категории.

Примечание. Вариант 1. Если вы разрабатывали отчеты Web Intelligence в другой системе, где приложение SHR не установлено, необходимо выполнить следующие действия:

- экспортируйте файл BIAR в систему, где установлена программа SHR, и
- создайте компонент «Отчеты» и установите его с помощью диспетчера развертывания.

См. инструкции в разделе ["Экспорт отчетов в SHR" на лицевой странице](#)

Вариант 2. Если вы разрабатывали отчеты Web Intelligence в системе, где установлено приложение SHR, вы можете проверить потоки рабочих процессов и просмотреть отчеты в интерфейсе SAP BusinessObjects InfoView, как описано в следующих разделах.

Просмотр отчетов в SAP BusinessObjects InfoView

После установки пакетов компонентов «Сфера деятельности» и «Отчеты» и загрузки данных в хранилище данных отчеты отображаются в интерфейсе SAP BusinessObjects InfoView.

Если вы установили образец пакета RetailPOS_Demo_Content_Pack из папки %CDE_HOME%\samples\, вы увидите отчет с именем Retail Sales Report доступный в окне Document List в InfoView. См. указания по входу в интерфейс InfoView и просмотру отчетов в [интерактивной справке HP Service Health Reporter для пользователей](#).

Отчет о розничных продажах содержит сводку продаж с информацией о выручке от продаж по каждому из измерений, определенных в модели данных, — расположению, времени и товарной категории. Вы можете выполнять детализацию и сведение данных по измерениям для просмотра информации с соответствующим уровнем детализации.

Демонстрация отчета о розничных продажах

Нажмите для воспроизведения видеозаписи. Нажмите правой кнопкой мыши для доступа к параметрам просмотра.

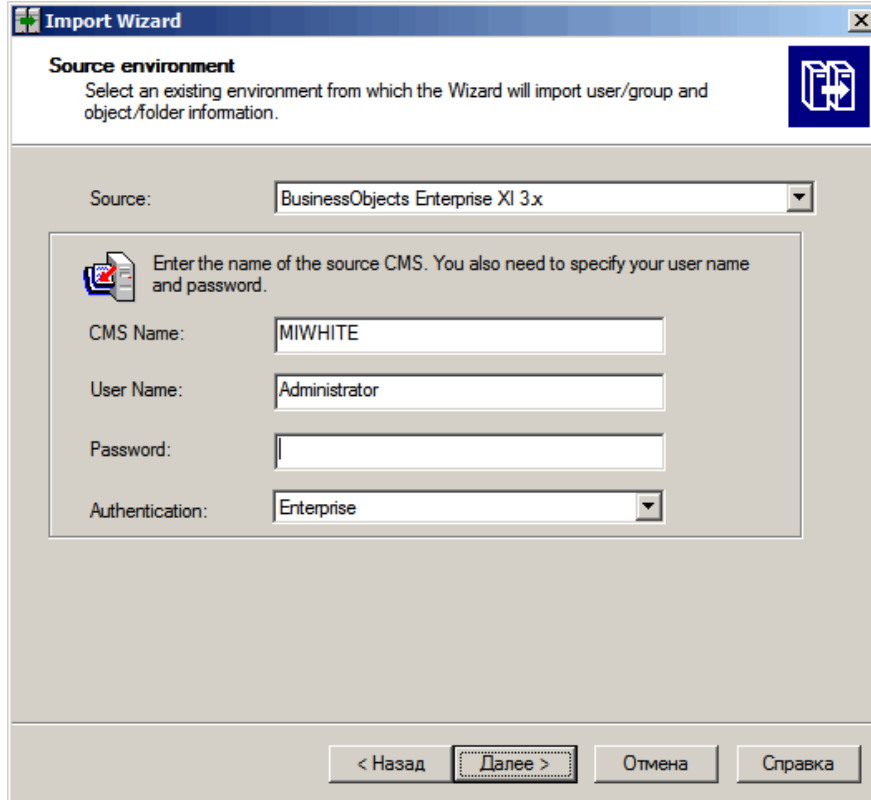
Экспорт отчетов в SHR

Экспорт файла Business Intelligence Archive Resource (BIAR)

Если вы разрабатывали отчеты в системе, где приложение SHR не установлено, необходимо экспортировать файл BIAR и установить компонент «Отчеты» в системе, где установлено приложение SHR. Необходимо выбрать исходную и конечную среду и объекты, которые вы хотите импортировать. Выполните следующие действия:

1. Откройте мастер импорта в приложении BusinessObjects.
2. На странице **Исходная среда** укажите следующие параметры:
 - Имя CMS: имя компьютера, на котором установлено приложение BusinessObjects
 - Имя пользователя: имя пользователя приложения BusinessObjects.
 - Пароль: пароль пользователя приложения BusinessObjects.

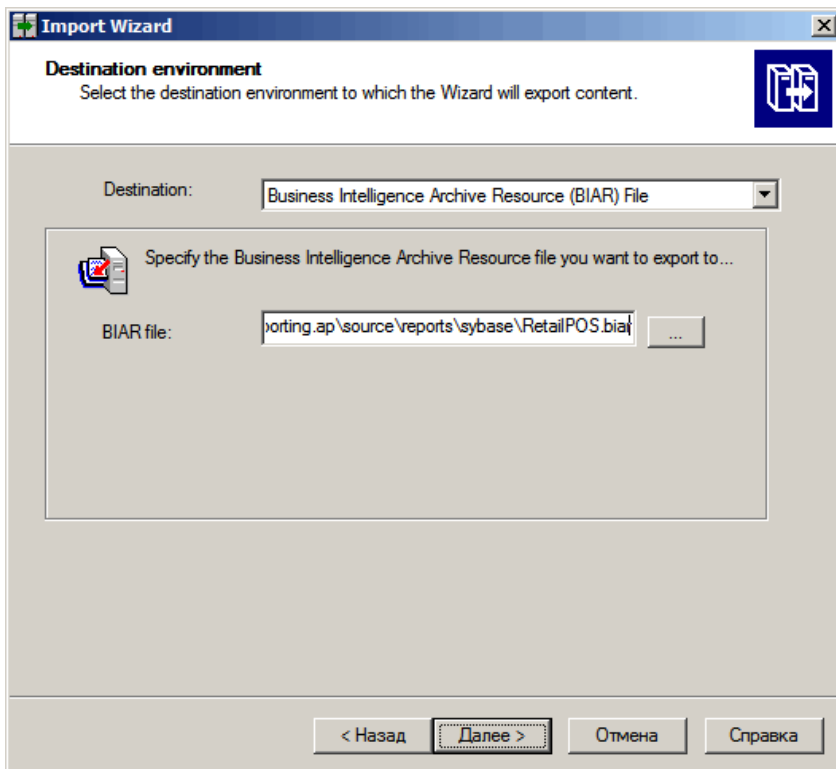
- Проверка подлинности: Выберите **Enterprise**.



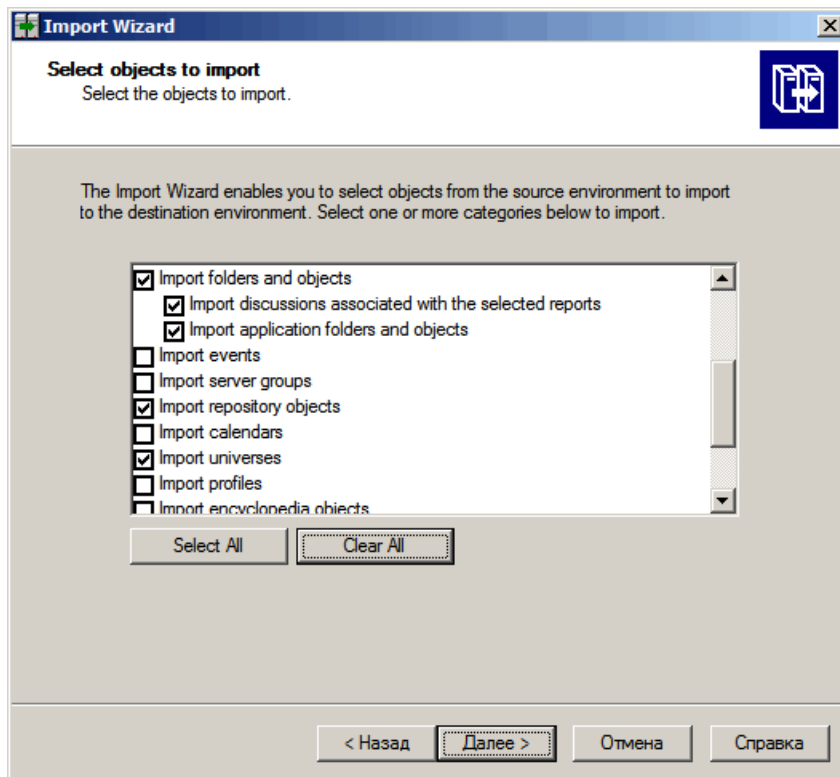
3. На странице **Конечная среда** выберите следующие параметры:

- Конечная среда: Файл Business Intelligence Archive Resource (BIAR).
- Файл BIAR: файл .biar, в который вы хотите экспортировать содержимое. Файл .biar находится в следующей папке:

`%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap\source\reports\sybase\`

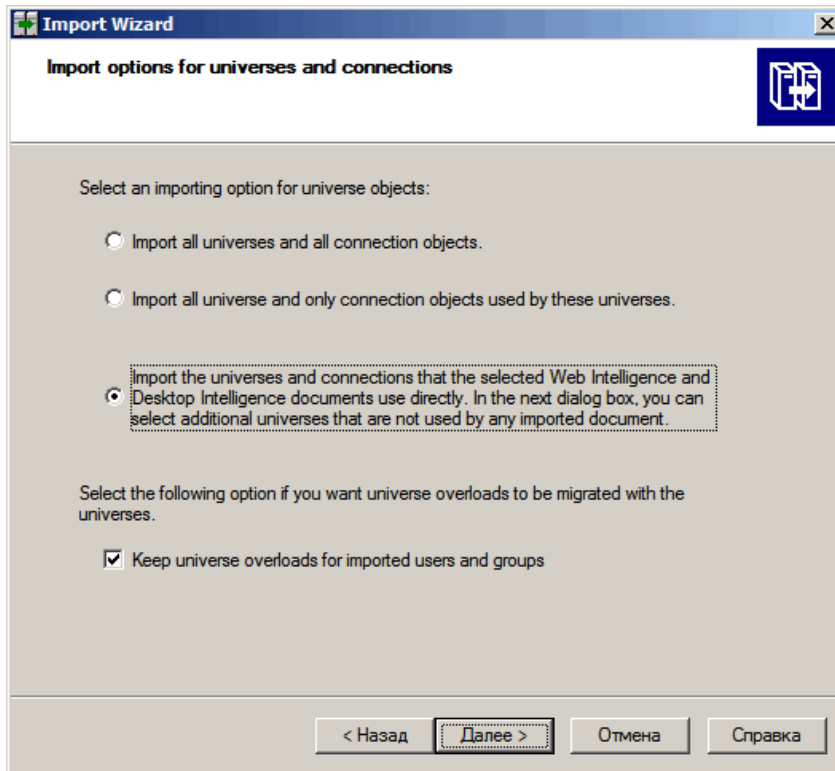


4. На странице **Выбор импортируемых объектов** выберите следующие параметры:
- Импорт папок и объектов.
 - Импорт объектов репозитория.
 - Импорт систем Universe.

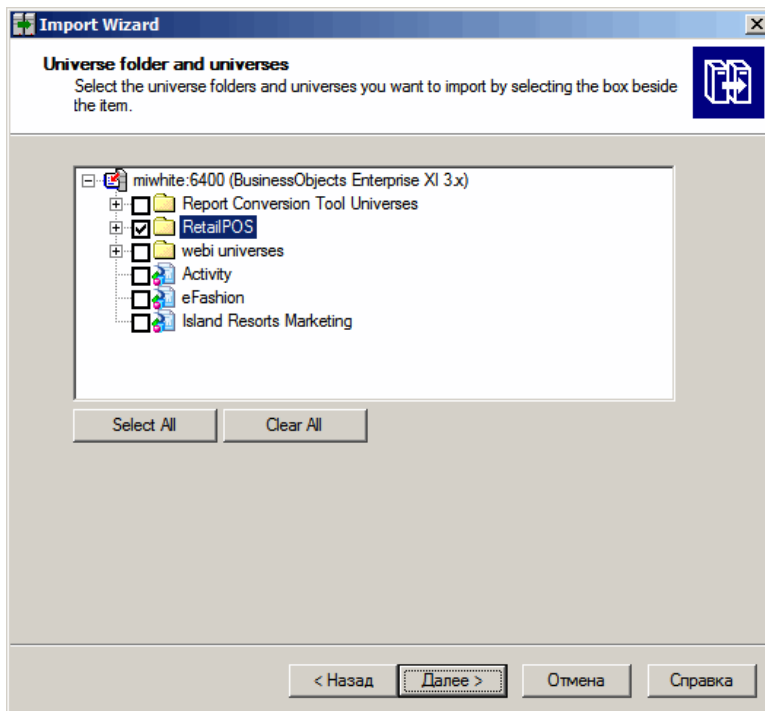


5. На странице **Параметры импорта для систем Universe и подключений** выберите следующий параметр:

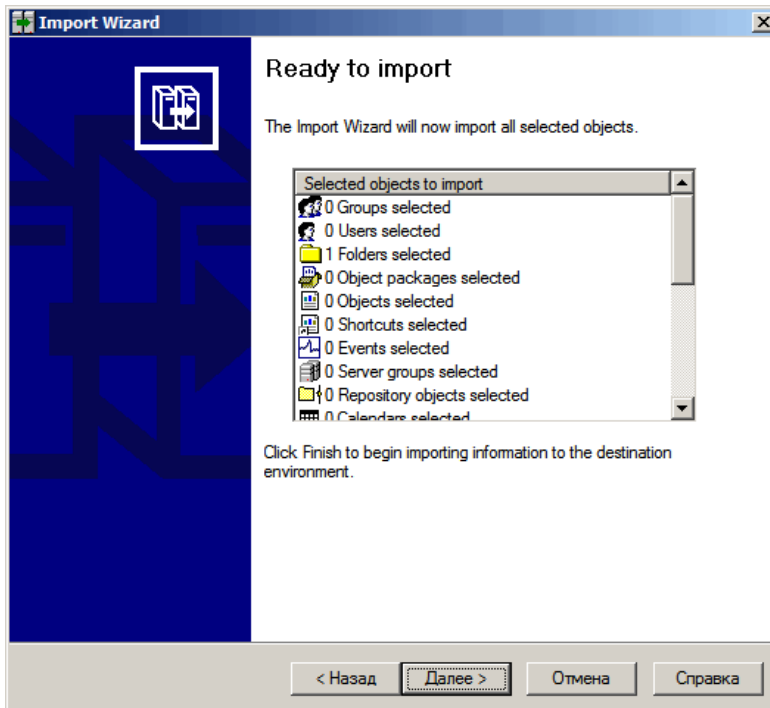
Импорт систем Universe и подключений, непосредственно используемых выбранными документами Web Intelligence и Desktop Intelligence.



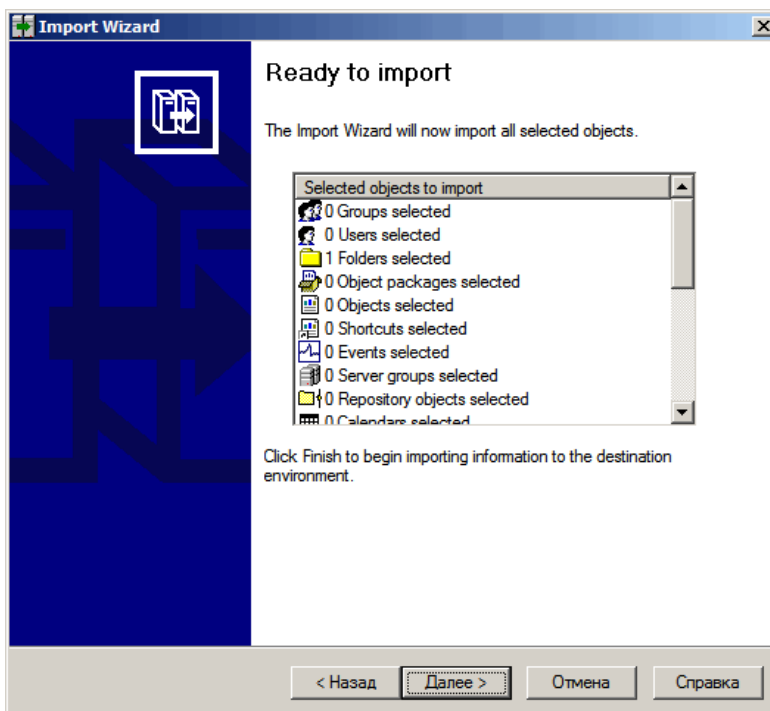
6. На странице **Папка Universe и системы Universe** выберите RetailPOS. Нажать кнопку **Next** («Далее»).



7. На странице **Параметры импорта для публикаций** выберите параметр Не импортировать получателей.



Отобразится страница **Импорт готов**. Чтобы импортировать Universe, нажмите кнопку **Готово**.



Создание XML-файла манифеста

XML-файл манифеста содержит определение файла BIAR, который вы импортировали в предыдущем этапе для использования в CPCS.

Чтобы создать XML-файл манифеста с помощью CPCS, выполните следующие действия:

- Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap
```

- Выполните следующую команду:

```
ant createManifestTemplate
```

XML-файл манифеста с именем `RetailPOSReporting_manifest_template.xml` создается в папке `%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap\source`

См. для справки образец XML-файла манифеста для компонента `RetailPOSReporting` в указанной ниже папке.

```
%CDE_HOME%\samples\
```

Использование CPCS для создания пакета компонента «Отчеты»

Чтобы создать пакет компонента «Отчеты» с помощью CPCS, выполните следующие действия:

1. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSReporting.ap
```

2. Выполните следующую команду:

```
ant createManifestTemplate
```

Пакет компонента «Отчеты» создается с именем `RetailPOSReporting.ap` в следующей папке:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap\dist
```

3. Перейдите в каталог `%CDE_HOME%\workspace\RetailPOS\RetailPOSDomain.ap\dist`.
4. Скопируйте `RetailPOSReporting.ap` в каталог `%PMDB_HOME%\packages\RetailPOS`

Установка пакета компонента «Отчеты»

Для установки компонентов пакета содержимого в SHR используется диспетчер развертывания пользовательского интерфейса администрирования платформы PMDB.

См. указания по установке компонентов пакета содержимого в *Руководстве по установке и настройке HP Service Health Reporter*.

После установки пакета компонента «Отчеты» можно проверить потоки рабочих процессов на административной консоли и просмотреть отчеты в интерфейсе SAP BusinessObjects InfoView.

См. "Проверка потоков рабочих процессов на административной консоли" на странице 29

См. "Просмотр отчетов в SAP BusinessObjects InfoView" ниже:

Просмотр отчетов в SAP BusinessObjects InfoView

После установки пакетов компонентов «Сфера деятельности» и «Отчеты» и загрузки данных в хранилище данных отчеты отображаются в интерфейсе SAP BusinessObjects InfoView.

Если вы установили образец пакета RetailPOS_Demo_Content_Pack из папки %CDE_HOME%\samples\, вы увидите отчет с именем Retail Sales Report доступный в окне Document List в InfoView. См. указания по входу в интерфейс InfoView и просмотру отчетов в *интерактивной справке HP Service Health Reporter для пользователей*.

Отчет о розничных продажах содержит сводку продаж с информацией о выручке от продаж по каждому из измерений, определенных в модели данных, — расположению, времени и товарной категории. Вы можете выполнять детализацию и сведение данных по измерениям для просмотра информации с соответствующим уровнем детализации.

Разработка контента в Linux с помощью CFC

Во время установки SHR в Linux также устанавливаются клиентские инструменты SAP BusinessObjects, но они не поддерживаются в Linux. Если приложение SHR устанавливается на сервере Linux, вы должны использовать Microsoft Windows XP или более позднюю версию для разработки и настройки содержимого приложений.

Предварительные условия

Система со следующим программным обеспечением:

1. Microsoft Windows XP или более поздняя ОС.
2. Java™ Platform, Standard Edition Development Kit (JDK™) 1.7.
Загрузите с сайта <http://www.oracle.com/>.
3. SAP BusinessObjectsXI 3.1 Client.
 - a. Скопируйте файл BusinessObjectsXI-3.1-Clienttools.zip из каталога *bits/packages/BO* установочного носителя Linux в системе Windows.
 - b. Извлеките содержимое ZIP-файла в системе Windows.
Будут созданы две папки — *BusinessObjectsXI-3.1-Clienttools\SP5Client* и *BusinessObjectsXI-3.1-Clienttools\SP5.3Client*.
 - c. Запустите файл setup.exe из обеих папок, начиная с файла SP5Client setup.exe.
 - d. Следуйте инструкциям в мастере установки.

Извлечение и настройка CPE

1. На сервере Linux, на котором установлено приложение SHR, скопируйте файл `CDE.zip` из каталога `/opt/HP/BSM`.
2. Извлеките содержимое ZIP-файла в каталог `root/CDE`.
3. Перейдите в каталог `root/CDE/bin` и предоставьте права выполнения сценариям оболочки.
4. Выполните команду `setenv.sh`.

Будут установлены следующие переменные среды:

- `CDE_HOME`
- `ANT_HOME`
- `JRE_HOME`
- `JRE_BIN`

Создание компонента приложения

1. На сервере Linux, на котором установлено приложение SHR, перейдите в каталог `/opt/HP/BSM`.
2. Скопируйте файл `CDE.zip` в систему Windows.
3. Извлеките содержимое GZ-файла в папку `C:\CDE`.
4. Из папки `C:\CDE\bin` запустите команду `setenv.bat`, указав полный путь к каталогу CDE и Java Runtime Environment (JRE) следующим образом:

```
setenv.bat -CDE_HOME C:\CDE -JRE_HOME C:\Java\jdk1.7.0_xx\jre
```
5. В каталоге `C:\CDE\bin` запустите файл `updateCDEProperty.bat`.
 - a. Укажите значения параметров `bo.username` и `bo.password`.
 - b. Введите короткое имя сервера SHR в поле `bo.server=`.
Например, `bo.server=shrdev`
6. Установите SAP Sybase IQ Network Client 15.4.
 - a. Скопируйте файл `SybaseIQ32_odbcDriver.zip` в папку `C:\CDE\bin`.
 - b. Извлеките содержимое ZIP-файла в системе Windows.
 - c. Запустите файл `setup.exe` в папке `C:\CDE\bin\SybaseIQ32_odbcDriver`.
7. В Windows нажмите кнопку **Пуск** запустите **Sybase > Sybase IQ 15.4 > ODBC Administrator 32-bit**. Откроется диалоговое окно ODBC Data Source Administrator.
8. Выберите вкладку **System DSN** и нажмите кнопку **Add**. Откроется диалоговое окно **Create New Data Source**.
9. В параметрах строки **Name** прокрутите окно и выберите **Sybase IQ**, а затем нажмите **Finish**. Откроется диалоговое окно ODBC Configuration for SQL Anywhere.
10. На вкладке **ODBC** в поле **Data source name** введите BSMR.
11. На вкладке **Login** выполните следующие действия:

- a. Для параметра **User ID** укажите значение `pmdb_admin`.
 - b. Для параметра **Password** введите пароль для схемы PMDB.
 - c. В поле **Action** выберите **Connect to a running database on another computer**.
 - d. В поле **Host** введите полное доменное имя сервера SHR.
 - e. В поле **Port** введите 21424.
 - f. В поле **Server name** введите короткое имя сервера SHR.
12. На вкладке **ODBC** нажмите **Test Connection** и после успешного завершения теста нажмите **OK**.
13. В Windows нажмите кнопку **Пуск** и запустите **Business Objects XI 3.1 > Business Objects Enterprise Client Tools > Designer**. Откроется диалоговое окно User Identification.
- a. В поле **Server name** введите короткое имя сервера SHR.
 - b. В поле **User Name** введите Administrator.
 - c. В поле **Password** введите пароль администратора SAP BusinessObjects. По умолчанию пароль не требуется.
 - d. В поле **Authentication** выберите Enterprise.
 - e. Нажмите кнопку **OK**. Откроется диалоговое окно **Universe Designer**.
14. Чтобы создать подключение к базе данных SHR, выберите **Tools** в меню и выберите **Connections**. Откроется диалоговое окно **Wizard Connection**.
- a. Нажмите кнопку **Add** («Добавить»). Откроется мастер **Define a new connection**.
 - b. Нажать кнопку **Next** («Далее»).
 - c. В поле **Connection Type** выберите **Secured**.
 - d. В поле **Connection Name** введите MA.
 - e. На уровне сети выберите **Sybase > Sybase IQ 15 > ODBC Drivers**.
 - f. Нажать кнопку **Next** («Далее»).
 - g. В поле **Authentication Mode** выберите **Use specified username and password**.
 - h. Оставьте поля **User name** и **Password** пустыми.
 - i. В поле **Data source name** введите BSMR.
 - j. Выберите **Next > Next > Finish**.
 - k. В диалоговом окне **Wizard Connection** нажмите **Finish**.
- Будет создано новое подключение к BusinessObjects с именем MA.
15. Запустите **командную строку** и выполните следующую команду для создания структуры каталогов пакета содержимого:
- ```
createCPFolders.bat -package cpname -subpackage subpackagename.ap -type application
```
16. Перейдите в каталог `%CDE_HOME%\workspace \package\subpackagename.ap` и выполните следующие команды:
- `ant createManifestTemplate`
  - `ant`
17. Система Universe создана и помещена в папку `%CDE_HOME%\workspace`



`\package\subpackagename.ap\dist.`

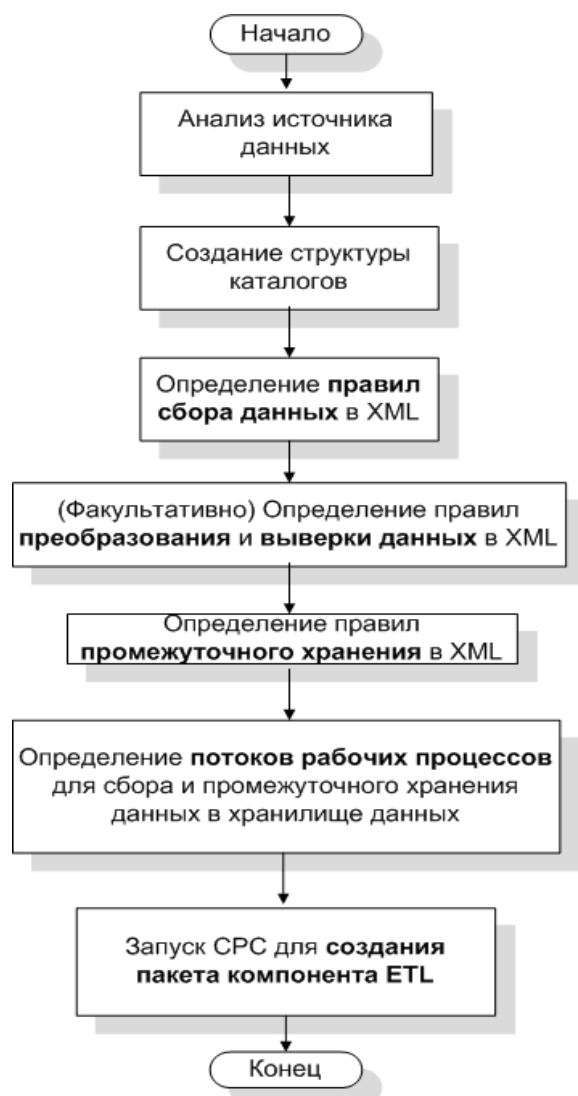
18. Экспорт системы Universe на сервер SAP BusinessObjects См. дополнительные сведения в разделе [Экспорт системы Universe в репозиторий SAP BusinessObjects](#).



## Глава 4: Создание пакета компонента ETL

В этой главе описаны этапы создания пакета компонента ETL, предназначенного для имеющегося пакета компонента «Сфера деятельности» пакета содержимого RetailPOS.

Создание пакета компонента ETL включает следующие этапы:



**Примечание.** В этой главе не приводится описание выверки данных.

## Предварительные требования

### Создание и установка пакета компонента «Сфера деятельности»

Выполните действия, описанные в разделе "[Создание и установка компонента «Сфера деятельности»](#)" на [странице 22](#), для создания модели сферы деятельности и используйте CPCS для создания пакета компонента «Сфера деятельности».

## Анализ источника данных

Как описано в главе "[Архитектура и разработка пакетов содержимого](#)" на [странице 14](#)», прежде чем приступить к созданию компонента ETL, необходимо указать источник данных, который содержит метрики, пригодные для ввода в модель данных сферы деятельности.

В данном руководстве в качестве источника данных рассматривается простая база данных, создаваемая с использованием программы PostgreSQL и содержащая данные, пригодные для ввода в модель данных пакета содержимого RetailPOS. На носителе SHR имеются образцы файлов и сценариев для создания базы данных и загрузки данных в таблицы. В разделе "[Создание источника данных для компонента ETL](#)" описано, как создать базу данных PostgreSQL, таблицы базы данных и вставить данные в таблицы.

## Создание структуры каталогов

Чтобы создать структуру каталогов для исходных файлов компонента ETL, введите в командной строке и выполните следующую команду:

```
<installation_directory>\CDE\bin>CreateCPFolders.bat -package RetailPOS -subpackage RetailPOSETL -type etl
```

Здесь:

- `<installation_directory>` — это каталог, в который выполнена установка SHR
- `RetailPOS` — это имя разрабатываемого пакета содержимого
- `RetailPOSETL` — это имя компонента ETL в RetailPOS

Каталог будет содержать шаблоны программы SHR, которые вы будете использовать для создания исходных файлов компонента ETL.

SHR содержит образцы исходных файлов для пакета содержимого пункта розничной торговли, которые размещены в указанной ниже папке. Эти файлы можно использовать для справки при создании своего собственного пакета содержимого.

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\
```

## Определение политики сбора данных в XML

Для сбора данных из таблиц базы данных `retailpos` необходимо определить политику сбора данных в XML. Используйте шаблон политики сбора данных `DB_collection_`

template.xml, который находится в следующей папке:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSETL.ap\source\etl\collection\
```

См. для справки образец RetailPOS\_DB\_Collection\_Policy.xml в следующей папке:

```
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS\RetailPOSETL.ap\source\etl\collection\
```

Чтобы ознакомиться с содержимым образца XML-файла политики сбора данных, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите файл RetailPOS\_DB\_Collection\_Policy.xml. Дважды щелкните, чтобы открыть файл XML в окне браузера.

## Определение правил преобразования данных

В примере с пунктом розничной торговли правило преобразования данных используется для добавления названия города, области, страны и почтового индекса в столбец адреса и для использования пробела в качестве разделителя значений столбцов в файлах .csv.

См. для справки образец RetailPOS\_transformation.xml, путь `%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS\RetailPOSETL.ap\doc`

Чтобы ознакомиться с содержимым образца XML-файла с правилом преобразования данных, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите файл RetailPOS\_transformation.xml. Дважды щелкните, чтобы открыть файл XML в окне браузера.

## Определение правил промежуточного хранения

Правила промежуточного хранения, указанные в файлах XML, используются для сопоставления имен столбцов в исходных файлах .csv с именами столбцов в соответствующих физических таблицах базы данных, называемых таблицами промежуточного хранения. Используйте шаблон правил промежуточного хранения, который находится в папке RetailPOS\RetailPOSDomain.ap\source\stagerule\_templates, для записи XML-файлов с правилами промежуточного хранения для каждого измерения: товар, продажи и магазин.

См. для справки образцы XML-файлов с правилом промежуточного хранения в папке `%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS\RetailPOSETL.ap\source\etl\stage_rules`

Для пакетов содержимого OOTB шаблоны правил промежуточного хранения находятся в папке `%CDE_HOME%\cplib\<ContentPack>\<SubPackage.ap>\source\stagerule_templates`

Например, синтаксис пакета содержимого System Management — `C:\HP-SHR\CDE\cplib\SystemManagement\CoreSystemManagement.ap\source\stagerule_templates`

Чтобы ознакомиться с содержимым образца XML-файла правил промежуточного хранения, щелкните значок в этом документе PDF: **Вложения: просмотр вложенных файлов** и выберите следующие файлы:

- Stage\_product\_stagerule.xml: правило промежуточного хранения данных для товара
- Stage\_retail\_sales\_stagerule.xml: правило промежуточного хранения данных о продажах;
- Stage\_store\_stagerule.xml: правило промежуточного хранения данных о магазине.

Дважды щелкните для открытия файлов XML в окне браузера.

## определение потоков рабочих процессов

Потоки рабочих процессов, определенные в файлах XML, используются для сбора данных из файлов .csv и их промежуточного хранения в хранилище данных.

При сборе данных топологии сборщик SHR создает файл \*relations\*.csv для каждого представления топологии в дополнение к типу ЭК, указанному в политике сбора данных RTSM. Вы должны создать правила промежуточного хранения этих данных и потоки рабочих процессов для загрузки файлов \*relations\*.csv.

Файлы \*relations\*.csv содержат данные топологии, которые должны быть загружены в таблицу моста ЭК. Чтобы создать определение правила промежуточного хранения для моста ЭК, воспользуйтесь шаблонами правил для моста ЭК в папке %CDE\_HOME%\cplib\Core\Core.ap\source\stagerule\_templates[Core\_0\_Stage\_K\_CI\_Bridge\_0\_stagerule.xml].

В дополнение к созданию правил промежуточного хранения необходимо задать поток рабочего процесса для активации потока данных. См. синтаксис для создания потоков ABC в документе *HP Service Health Reporter Content Development - Reference Guide*.

**Примечание.** На этапе COLLECT воспользуйтесь значением «Имя представления», указанным в политике сбора данных в качестве категории и значением «Связи» в качестве типа.

Используйте шаблон потоков рабочих процессов, который находится в папке %CDE\_HOME%\samples\RetailPOS\_Demo\_Content\_Pack\source\RetailPOS\RetailPOSETL.ap\source\orchestration\stream\_definitions, для записи XML-файлов потоков рабочих процессов — по одному файлу для каждого измерения: товар, продажи и магазин.

См. для справки образцы XML-файлов с потоками рабочих процессов в папке %CDE\_HOME%\samples\RetailPOS\_Demo\_Content\_Pack\RetailPOS\RetailPOSDomain.ap\source\orchestration\stream\_definitions

Чтобы ознакомиться с содержимым XML-файлов потоков рабочих процессов ETL, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и выберите следующие файлы:

- Dimension\_Product\_ETL\_stream.xml: XML-поток рабочих процессов для измерения «Товар»;
- Dimension\_Store\_ETL\_stream.xml: XML-поток рабочих процессов для измерения «Магазин»;
- Fact\_Retail\_Sales\_ETL\_stream.xml: XML-поток рабочих процессов для фактов.

Дважды щелкните для открытия файлов XML в окне браузера.

## Создание компонента ETL

Чтобы создать пакет компонента ETL с помощью CPC, выполните следующие действия:

1. Воспользуйтесь командой `cd` для перехода в следующий каталог:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSETL.ap
```

2. Выполните следующую команду:

```
ant
```

Пакет компонента ETL создается в следующем каталоге:

```
%CDE_HOME%\workspace\RetailPOS\RetailPOSETL.ap\dist
```

3. Перейдите в каталог `%CDE_HOME%\workspace\RetailPOS\RetailPOSETL.ap\dist`
4. Скопируйте `RetailPOS` в `%PMDb_HOME%\packages`. Таким образом пакет становится доступен для установки в диспетчере развертывания на административной консоли.

## Установка компонента ETL

Для установки пакетов компонентов пакета содержимого в SHR используется диспетчер развертывания, доступный на административной консоли. Используйте диспетчер развертывания для установки пакета компонента ETL — в примере с пунктом розничной торговли этот пакет называется `RetailPOSETL`.

См. указания по установке компонентов пакета содержимого с помощью диспетчера развертывания в *интерактивной справке для администраторов HP Service Health Reporter*.

## Работа с компонентом ETL

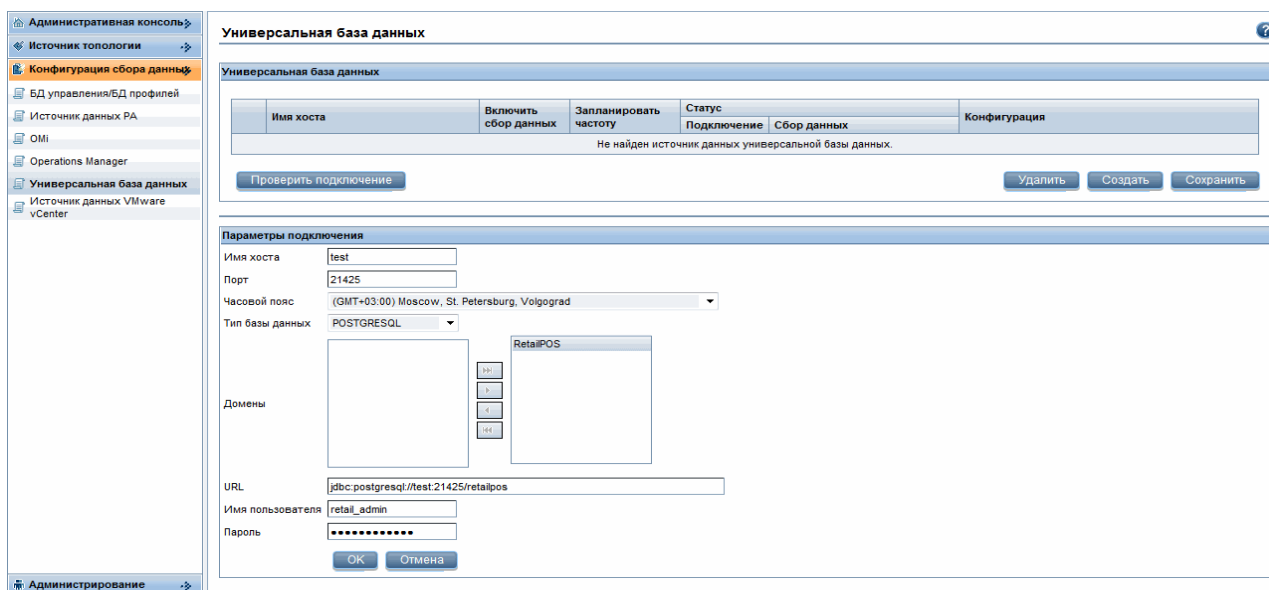
### Настройка универсальной базы данных

После установки компонентов «Сфера деятельности» и ETL пакета содержимого `RetailPOS` необходимо настроить подключение к базе данных `retailpos` для сбора данных в соответствии с политикой сбора данных в файле XML. Настройка подключения выполняется на странице «Универсальная база данных» административной консоли. Выполните следующие действия:

1. В административной консоли выберите пункты **Конфигурация сбора данных - > Универсальная база данных**. Откроется страница «Универсальная база данных».
2. Нажмите кнопку **Создать**. Откроется диалоговое окно «Параметры подключения».
3. Введите или выберите следующие значения:

| Поле     | Описание                                                                                                     |
|----------|--------------------------------------------------------------------------------------------------------------|
| Имя узла | Введите IP-адрес или полное доменное имя сервера, на котором вы создали базу данных <code>retailpos</code> . |

| Поле             | Описание                                                                                                         |
|------------------|------------------------------------------------------------------------------------------------------------------|
| vCenter          | Введите номер порта для запросов к серверу базы данных.                                                          |
| Часовой пояс     | Выберите часовой пояс, в соответствии с которым настраивается экземпляр базы данных.                             |
| Тип базы данных  | Выберите POSTGRESQL.                                                                                             |
| Домен            | Выберите RetailPOS.                                                                                              |
| URL-адрес        | Введите jdbc:postgresql://<server>:<port>/retailpos                                                              |
| Имя пользователя | Введите имя пользователя универсальной базы данных. В данном примере используется имя пользователя retail_admin. |
| Пароль           | Введите пароль пользователя универсальной базы данных. В данном примере используется пароль retail admin.        |



## Проверка компонента ETL

После установки пакета компонента ETL и запуска службы HP\_PMDB\_Platform\_Timer необходимо войти в пользовательский интерфейс администрирования, чтобы проверить статус потоков рабочих процессов компонента ETL. Выполните следующие действия:

1. В интерфейсе администрирования выберите пункт **Внутренний мониторинг->Обработка данных**.
2. На вкладке **Сведения о потоках** просмотрите статус потоков пакета содержимого RetailPOSETL. В случае успешного завершения у всех потоков будет указан статус OK.

В примере с пунктом розничной торговли компонент ETL содержит следующие потоки рабочих процессов, при этом каждый поток содержит один или несколько этапов.



- Поток рабочих процессов RetailPOSETL@Retail\_Sales\_ETL используется для перемещения фактов о продажах в таблицы промежуточного хранения.
- Поток рабочих процессов RetailPOSETL@Product\_ETL используется для перемещения измерения товара в таблицы промежуточного хранения.
- Поток рабочих процессов RetailPOSETL@Store\_ETL используется для перемещения измерения магазина в таблицы промежуточного хранения.
- Поток рабочих процессов RetailPOSETL@Promotion\_ETL используется для перемещения измерения рекламной акции в таблицы промежуточного хранения.

Как показано на приведенном ниже изображении, успешное завершение потоков обозначается зеленым цветом.

| Сведения о потоке для компонента пакета содержимого : RetailPOSETL |                              |             |                     |
|--------------------------------------------------------------------|------------------------------|-------------|---------------------|
| Имя потока                                                         | Статус шага (Выполнен/Всего) | Статус шага | Время начала        |
| RetailPOSETL@Promotion_ETL                                         | 1/1                          | SUCCESS     | 01.11.2012 17:00:18 |
| RetailPOSETL@Retail_Sales_ETL                                      | 1/1                          | SUCCESS     | 01.11.2012 17:00:17 |
| RetailPOSETL@Product_ETL                                           | 1/1                          | SUCCESS     | 01.11.2012 17:00:18 |
| RetailPOSETL@StoreETL                                              | 2/2                          | SUCCESS     | 01.11.2012 17:00:17 |

## Просмотр отчетов

После установки пакетов компонентов «Сфера деятельности» и «Отчеты» и загрузки данных в хранилище данных отчет отображается в интерфейсе SAP BusinessObjects InfoView. См. "[Просмотр отчетов в SAP BusinessObjects InfoView](#)" на странице 46:

# Глава 5: Создание пакета содержимого — упрощенный метод

В этом разделе описывается упрощенный метод создания компонентов сферы деятельности, ETL, и отчетов пакета содержимого со следующими источниками данных:

- Универсальная база данных  
SHR собирает данные только из баз данных, поддерживающих Java Database Connectivity (JDBC).
- HP Operations agent or HP Performance Agent
- Файлы CSV

## Анализ источника данных

Необходимо указать источник данных, содержащий метрики, пригодные для ввода в модель данных сферы деятельности. См. сведения о разработке модели данных в разделе ["разработка модели данных " на странице 24"](#).

## Универсальная база данных как источник данных

Если в качестве источника данных SHR используется универсальная база данных (Microsoft SQL, Oracle, Sybase IQ и др.), выполните следующие действия для создания компонентов сферы деятельности, отчетов и ETL пакета содержимого:

1. В командной строке перейдите в каталог `%CDE_HOME%\bin`.
2. Выполните следующую команду:  

```
createcpfolders.bat -package RetailPOS -type all
```

Будут созданы папки для компонентов пакета содержимого (сфера деятельности, ETL и отчеты).
3. Создайте следующие входные файлы в соответствии с вложенными образцами для RetailPOS и сохраните их в папке `%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source`:

**Примечание.** Чтобы ознакомиться с содержимым образцов вложений, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и дважды щелкните файл, чтобы открыть файл XML в окне браузера.

- a. Файл свойств подключения к базе данных (`dbconfig.properties`)  
Этот файл содержит учетные данные источника данных (универсальной базы данных), из которой пакет содержимого извлекает данные. Этот файл также

содержит учетные данные базы данных SHR (Sybase IQ), необходимые для создания SAP BusinessObjects Universe.

b. XML-файл запроса базы данных (*RetailPOS\_Query.xml*)

Этот файл содержит SQL-запросы для получения данных из источника данных (универсальная база данных). Он также содержит определения таблиц и столбцов в SHR для сохранения данных.

SHR содержит JDBC-драйверы для БД Microsoft SQL, Oracle, PostgreSQL, и Sybase IQ в файле CDE.zip. Если вы используете другую базу данных, скопируйте соответствующий JDBC-драйвер в систему SHR и среду разработки содержимого.

4. Используя файл свойств подключения к базе данных и XML- файл запроса базы данных в качестве входных данных, выполните следующую команду для создания файла сопоставления (ModelMapper.CSV):

```
createMappingFile.bat -useDB -passwords "mssql.password=<password>" -dbQueryXml
%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS_Query.xml -
config %CDE_HOME%\samples\RetailPOS_Demo_Content_
Pack\Source\dbconfig.properties
```

Проверьте значения в CSV-файле сопоставления и при необходимости измените его вручную в соответствии с требованиями бизнеса.

5. Выполните следующую команду для создания пакета содержимого:

```
createendtoendcp.bat -mappingfile "%CDE_HOME%\workspace\RetailPOS\RetailPOS_
MappingFile2394283040045872.csv" -collectionpolicyxml "%CDE_
HOME%\workspace\RetailPOS\RetailPOS_CollectionPolicy.xml" -configfile "%CDE_
HOME%\samples\RetailPOS_Demo_Content_Pack\Source\dbconfig.properties" -password
"db.password=<password>"
```

Компоненты сферы деятельности, ETL и отчетов созданы.

## CSV-файлы как источник данных

Если в качестве источника данных SHR используется файл CSV, выполните следующие действия для создания компонентов сферы деятельности, отчетов и ETL пакета содержимого:

1. В командной строке перейдите в каталог `%CDE_HOME%\bin`.
2. Выполните следующую команду:

```
createcpcfolders.bat -package RetailPOS -type all
```

Будут созданы папки для компонентов пакета содержимого (сфера деятельности, ETL и отчеты).

3. Создайте следующие входные файлы в соответствии с вложенными образцами для RetailPOS и сохраните их в папке `%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source`:

**Примечание.** Чтобы ознакомиться с содержимым образцов вложений, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и дважды щелкните файл, чтобы открыть файл XML в окне браузера.

- a. Файл свойств подключения CSV (*dbconfig.properties*)  
Этот файл содержит учетные данные базы данных SHR (Sybase IQ), необходимые для создания SAP BusinessObjects Universe.
  - b. CSV-файл политики (*RetailPOS\_Query.xml*)  
Этот файл содержит SQL-запросы для получения данных из источника данных (универсальная база данных). Он также содержит определения таблиц и столбцов в SHR для сохранения данных.
4. Используя CSV-файл политики и XML- файл политики CSV в качестве входных данных, выполните следующую команду для создания файла сопоставления (ModelMapper.CSV):

```
createMappingFile.bat -useCSV -csvXml %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\RetailPOS_Query.xml -config %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\dbconfig.properties -inputLocation %CDE_HOME%\inputcsvs-outputLocation %CDE_HOME%\workspace\RetailPOS
```

Проверьте значения в CSV-файле сопоставления и при необходимости измените его вручную в соответствии с требованиями бизнеса.

5. Выполните следующую команду для создания пакета содержимого:

```
createendtoendcp.bat -mappingfile "%CDE_HOME%\workspace\RetailPOS\RetailPOS_MappingFile2394283040045872.csv" -configfile "%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\dbconfig.properties" -password "mssql.password=<password>"
```

Компоненты сферы деятельности, ETL и отчетов созданы.

## Агент HP Performance как источник данных

Если в качестве источника данных SHR используется файл CSV, выполните следующие действия для создания компонентов сферы деятельности, отчетов и ETL пакета содержимого:

1. В командной строке перейдите в каталог *%CDE\_HOME%/bin*.
2. Выполните следующую команду:  

```
createcpfolders.bat -package RetailPOS -type all
```

  
Будут созданы папки для компонентов пакета содержимого (сфера деятельности, ETL и отчеты).
3. Создайте следующие входные файлы в соответствии с вложенными образцами для RetailPOS и сохраните их в папке *%CDE\_HOME%\samples\RetailPOS\_Demo\_Content\_Pack\Source*:

**Примечание.** Чтобы ознакомиться с содержимым образцов вложений, нажмите в этом документе PDF на значке **Вложения: просмотр вложенных файлов** и дважды щелкните файл, чтобы открыть файл XML в окне браузера.

- a. Файл свойств подключения (*dbconfig.properties*)  
Этот файл содержит сведения об имени хоста, порте и имени пользователя и имени хоста RTSM, и также об имени хоста агента HP Performance Agent, в котором

необходимо разработать агент. Этот файл также содержит учетные данные базы данных SHR, необходимые для создания SAP BusinessObjects Universe.

b. XML-файл политики топологии (*TopologyRTSM\_OM.XML*)

Этот файл содержит имена представлений, типы ЭК и атрибуты, которые должны собираться или исключаться ЭК. Кроме того, он содержит имена классов НРОМ и сведения об источниках данных. В случае политики сбора данных НРОМ имена атрибутов создаются для содержимого ООВ. Для новых типов ЭК атрибуты должны соответствовать атрибутам, указанным в политике сбора данных СМДВ.

c. XML-файл политики агента HP Performance Agent (*Fact\_PA.XML*)

Этот файл содержит источник данных и классы HP Performance Agent, из которых должны быть собраны данные.

4. Используя файлы, указанные ранее (*ModelMapper.CSV*) в качестве входных данных, выполните следующую команду для создания файла сопоставления (*ModelMapper.CSV*):

```
createMappingFile.bat -usePA -paXml %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\Fact_PA.xml -topologyXml %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\TopologyRTSM_OM.xml -config %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\dbconfig.properties -outputLocation %CDE_HOME%\workspace\RetailPOS -passwords <RTSM password>
```

Проверьте значения в CSV-файле сопоставления и при необходимости измените его вручную в соответствии с требованиями бизнеса.

5. Выполните следующую команду для создания пакета содержимого:

```
createendtoendcp.bat -mappingfile "%CDE_HOME%\workspace\RetailPOS\RetailPOS_MappingFile2394283040045872.csv" -topologyxml %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\TopologyRTSM_OM.xml -paXml %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\Fact_PA.xml -configfile %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\Source\dbconfig.properties
```

Компоненты сферы деятельности, ETL и отчетов созданы.

## Создание пакетов компонента «Сфера деятельности» и ETL

1. Перейдите в каталог `%CDE_HOME%\workspace\RetailPOS\RetailPOSDomainCP.ap\dist` (для домена) и `%CDE_HOME%\workspace\RetailPOS\RetailPOSETLCP.ap\dist` (для ETL)
2. Скопируйте сферу деятельности RetailPOS и ETL в каталог `%PMDB_HOME%\packages`.  
Благодаря этому пакет становится доступен для установки в диспетчере развертывания.

**Примечание.** Нет необходимости в том, чтобы приложение SHR было установлено на компьютере, на котором вы создаете пакет компонента «Сфера деятельности». Если вы создали пакет компонента «Сфера деятельности» на другом компьютере, необходимо скопировать его на компьютер SHR в папку `%PMDB_HOME%\packages`.

## Установка пакетов компонента «Сфера деятельности» и ETL

Для установки пакетов компонентов пакета содержимого в SHR используется диспетчер развертывания, доступный на административной консоли. См. указания по установке компонентов пакета содержимого в *интерактивной справке для администраторов HP Service Health Reporter*.

## Просмотр отчетов SHR

Для просмотра отчетов на основе данных, собранных с помощью созданного пакета содержимого, выполните действия, описанные в разделах ниже в следующем порядке:

1. ["Экспорт системы Universe в репозиторий SAP BusinessObjects" на странице 34](#)
2. ["создание отчетов Web Intelligence" на странице 36](#)
3. ["Экспорт отчетов в SHR" на странице 39](#)
4. ["Просмотр отчетов в SAP BusinessObjects InfoView" на странице 46](#)

# Приложение А

## Создание источника данных для компонента ETL

Для создания примера базы данных PostgreSQL в SHR используются следующие файлы и сценарии:

| Файлы/сценарии, встроенные в SHR                                                                                                                                                                              | Расположение файла/сценария                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| <p>Сценарии используются, чтобы скопировать в таблицы базы данных следующие файлы .csv:</p> <p>RetailPOS_Product.csv</p> <p>RetailPOS_Promotion.csv</p> <p>RetailPOS_Sales.csv</p> <p>RetailPOS_Store.csv</p> | <p>%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS_DB_Creation_Scripts\RetailPOS_CSV</p> |
| <p>RetailPOS_CreateDatabase.sql</p> <p>С помощью этого сценария SQL создается база данных PostgreSQL с именем RetailPOS для пользователя с именем retail_admin.</p>                                           | <p>%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS_DB_Creation_Scripts</p>               |
| <p>RetailPOS_CreateTables.sql</p> <p>Этот сценарий SQL используется для создания таблиц в базе данных RetailPOS.</p>                                                                                          | <p>%CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS_DB_Creation_Scripts</p>               |

| Файлы/сценарии, встроенные в SHR                                                                                                                                                                                                | Расположение файла/сценария                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| RetailPOS_PopulateTables.sql<br><br>С помощью этого сценария SQL выполняется копирование файлов .csv из папки %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS_DB_Creation_Scripts\RetailPOS_CSV в таблицы базы данных. | %CDE_HOME%\samples\RetailPOS_Demo_Content_Pack\RetailPOS_DB_Creation_Scripts |

**Предварительные требования** Прежде чем приступить к созданию базы данных PostgreSQL, выполните следующие действия:

Загрузите и установите программное обеспечение PostgreSQL с веб-сайта <http://www.postgresql.org/>. PostgreSQL можно установить в любой системе, необязательно там, где установлено приложение SHR.

Скопируйте следующие файлы на диск C:\ в системе, где вы установили PostgreSQL.

- RetailPOS\_CSV
- RetailPOS\_CreateDatabase.sql
- RetailPOS\_CreateTables.sql
- RetailPOS\_PopulateTables.sql

## Разработка содержимого для пакетов содержимого, включающих только компоненты приложения и ETL

Чтобы разработать настраиваемый пакет содержимого только с компонентами приложения и ETL (без сферы деятельности), необходимо повторно запустить сборку CPCS для восстановления пакета содержимого с помощью действительной системы Universe SAP BusinessObjects. В противном случае будут возникать такие ошибки, как «Unable to parse object XYZ».

Используйте следующий рабочий процесс для создания пакета содержимого. (для преобразования в блок-схему)

Создание структуры каталогов компонента «Отчеты».



Написание документа XML-модели

Создание XML-файла манифеста

Создание подключения к SAP BusinessObjects

Использование CPC для создания системы SAP BusinessObjects Universe

Установка пакета содержимого (с помощью административной консоли SHR > Диспетчер развертывания)

Запуск сборки CPC для создания пакета содержимого

Экспорт системы Universe в репозиторий SAP BusinessObjects

Экспорт отчетов в SHR

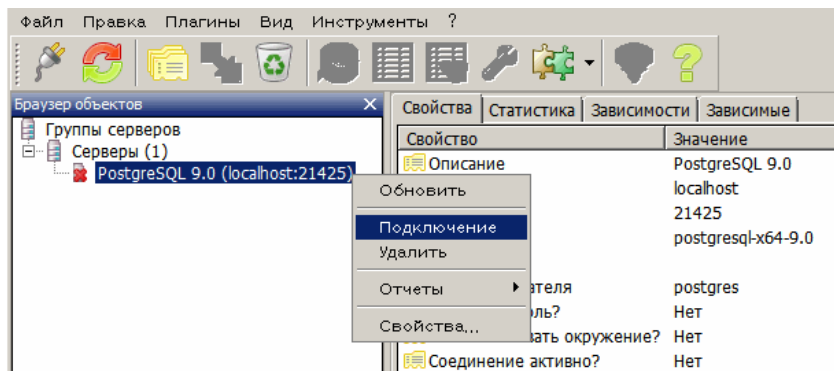
Запуск сборки CPC для восстановления пакета содержимого

Установка пакета содержимого (с помощью административной консоли SHR > Диспетчер развертывания)

## Создание базы данных PostgreSQL

Чтобы создать базу данных с именем RetailPOS, выполните следующие действия:

1. Выполните вход от имени администратора в систему, где установлена программа PostgreSQL.
2. Запустите программу PostgreSQL **pgAdmin III**.
3. Подключитесь к пользователю по умолчанию **postgres**, используя настроенный пароль.



В окне Object browser показаны базы данных, доступные пользователю postgres.

4. Закройте редактор SQL-запросов и выберите **Файл ->Открыть**. Перейдите в папку на диске C:\, в которую вы скопировали сценарий RetailPOS\_CreateDatabase.sql, и нажмите кнопку **Открыть**.
5. Нажмите **Execute pgScript** для запуска сценария RetailPOS\_CreateDatabase.sql.  
С помощью этого сценария создаются база данных retailpos и ее владелец — пользователь retail\_admin.
6. Закройте редактор SQL-запросов и нажмите **Refresh**.  
База данных retailpos с пользователем retail\_admin отобразятся в окне Object browser.

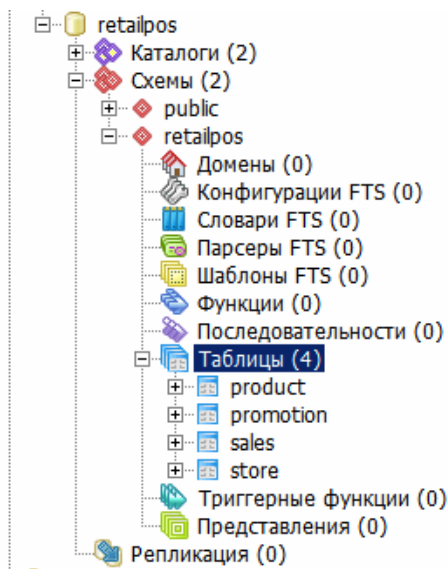
## Создание таблиц базы данных

Чтобы создать таблицы в базе данных retailpos, выполните следующие действия:

1. В окне Object browser выберите **retailpos** из списка баз данных и откройте редактор SQL-запросов.
2. Закройте редактор SQL-запросов и выберите **Файл ->Открыть**. Перейдите в папку на диске C:\, в которую вы скопировали сценарий RetailPOS\_CreateTables.sql, и нажмите кнопку **Открыть**.
3. Нажмите **Execute pgScript** для запуска сценария RetailPOS\_CreateTables.sql.

Будут созданы следующие таблицы, как показано на приведенном ниже рисунке:

- retailpos.product
- retailpos.store
- retailpos.promotion
- retailpos.sales



## Вставка данных в базу данных

Чтобы вставить данные из файлов .csv в таблицы, выполните следующие действия:

1. Закройте редактор SQL-запросов и выберите **Файл ->Открыть**. Перейдите в папку на диске C:\, в которую вы скопировали сценарий RetailPOS\_PopulateTables.sql, и нажмите кнопку **Открыть**.
2. Нажмите **Execute pgScript** для запуска сценария RetailPOS\_PopulateTables.sql.

С помощью этого сценария файлы .csv вставляются из файла RetailPOS\_CSV в таблицы базы данных.

# Часть II

Разработка содержимого — справочное руководство

## Обзор

В этом документе представлен подробный обзор синтаксиса исходных XML-файлов, которые разработчик содержимого должен создать для построения пакета содержимого с помощью CPCS.

См. сведения о встроенных пространствах (Universe) SHR BusinessObjects в разделе *Проверка установки* документа *Руководство по установке и настройке HP Service Health Reporter*.

См. сведения об интерфейсе промежуточного хранения и документацию по моделям для всех встроенных компонентов домена в документе интерфейса промежуточного хранения и модели, который входит в каждый компонент домена.

Для готовых пакетов содержимого SHR документ интерфейса промежуточного хранения и модели доступен в каждом пакете содержимого. Например, для домена «Управление системой» документ интерфейса промежуточного хранения и модели находится по следующему пути:

```
%PMDB_HOME%\packages\SystemManagement\CoreSystemManagement.ap\doc (Windows)
```

```
$PMDB_HOME/packages/SystemManagement/CoreSystemManagement.ap/doc (Linux)
```

Для вновь разработанного настраиваемого содержимого документация по интерфейсу промежуточного хранения и модели автоматически создается CPCS. Например, для домена «Управление системой» документ интерфейса промежуточного хранения и модели находится по следующему пути:

```
%CDE_HOME%\cplib\SystemManagement\CoreSystemManagement.ap\doc (Windows)
```

```
$CDE_HOME/cplib/SystemManagement/CoreSystemManagement.ap/doc (Linux)
```

См. дополнительные сведения о матрице хранилища данных SHR в HTML-документе «Матрица хранилища данных SHR», доступном по следующему пути:

```
%CDE_HOME%\doc\bus_matrix (Windows) или $CDE_HOME\doc\bus_matrix (Linux).
```

**Примечание.** HP представляет поддержку пакетов содержимого, созданных или измененных в CPCS или иным способом в рамках стандартных соглашений о поддержке и обслуживании продуктов HP. Однако любые дефекты, обнаруженные в базовом средстве HP, устраняются HP.

## Предварительные условия

Предполагается, что читатель знаком со следующими материалами.

| Предварительные требования                                                               | Справочная документация                                                         |
|------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Основные понятия разработки содержимого SHR                                              | Разработка содержимого — руководство по началу работы.                          |
| Концепция хранилища данных                                                               | Ресурсы по концепции хранилища данных можно найти в Интернете.                  |
| Основы XML и создание документов XML                                                     | Ресурсы, связанные с концепцией XML, можно найти в Интернете.                   |
| SQL                                                                                      | Ресурсы, связанные с SQL, можно найти в Интернете.                              |
| Понимание HP Run-time Service Model (RTSM) и HP Configuration Management Database (CMDB) | Руководство по моделированию HP BSM 9.2, руководстве по администрированию RTSM. |

## Определение терминов

| Термин | Определение                                    |
|--------|------------------------------------------------|
| CPC    | Среда разработки содержимого                   |
| ЭК     | Элемент конфигурации                           |
| CIUID  | Уникальный идентификатор элемента конфигурации |
| CMDB   | База данных управления конфигурацией           |
| DBMS   | Система управления базами данных               |
| ETL    | Извлечение, преобразование и загрузка          |
| OA     | HP Operations Agent                            |
| OM     | HP Operations Manager;                         |
| PA     | HP Performance Agent                           |
| RTSM   | Модель обслуживания во время выполнения        |
| SPI    | Интеллектуальные подключаемые модули HP:       |
| XML    | Xtensible Markup Language                      |

# Глава 6: Определение XML-модели

XML-модель определяет модель хранилища данных для определенной модели.

## Структура XML-модели

```
<schema>
 <relational >
 <dimensionTable >
 <column />
 </dimensionTable >
 <bridgeTable >
 <column />
 </bridgeTable >
 <factTable >
 <column />
 </factTable >
 </relational >
 <logical >
 <dimension >
 <memberGroup >
 <member />
 <calculatedMember />
 </memberGroup >
 <hierarchy >
 <level >
 <group />
 </level >
 </hierarchy >
 </dimension >
 <cube >
 <measures >
 <measure />
 <calculatedMeasure />
```

```

</ measures >
< dimensionRef />
</ cube >
< cubeRef />
< cubeRegion >
< derivedMeasures >
< measureRef />
</ derivedMeasures >
< aggregRef />
</ cubeRegion >
</ logical >
< aggregates >
< aggregate >
< aggregateMeasure />
< calculatedMeasure />
< aggLevel />
</ aggregate >
< forecast >
< aggregateMeasure />
< calculatedMeasure />
< aggLevel />
</ forecast >
</ aggregates >
</ schema >

```

Элемент XML: **schema**

Корневой элемент определения модели. Схема представляет модель базы данных домена. Она включает реляционную модель данных (таблицы базы данных), логическое представление реляционной модели данных и определение сведения данных. Схема также может ссылаться на реляционные и логические элементы другой схемы.

Атрибут	Обязательный	Описание
name	Да	Имя схемы
version	Да	Версия схемы
caption	Да	Метка схемы.

Дочерние элементы	Множество	Описание
<a href="#">relational</a>	0..1	Раздел реляционной схемы
<a href="#">logical</a>	0..1	Раздел логической схемы
<a href="#">aggregation</a>	0..1	Раздел сведения

Элемент XML: **relational**

#### Определение реляционной схемы

Дочерние элементы	Множество	Описание
<a href="#">factTable</a>	0..n	Список таблиц фактов в реляционной модели
<a href="#">dimensionTable</a>	0..n	Список таблиц измерений в реляционной модели
<a href="#">bridgeTable</a>	0..n	Список мостовых таблиц в реляционной модели

## Элемент XML: factTable

Элемент factTable представляет физическую таблицу хранилища данных, содержащую численные измерения производительности

Атрибут	Обязательный	По умолчанию	Описание
name	Да		Уникальное имя таблицы фактов
caption	Да		Метка таблицы фактов.
version	Да		Версия таблицы модели
description	Нет		Краткое описание таблицы фактов
type	Да		Уровень детализации таблицы фактов. Допустимые значения <ul style="list-style-type: none"> <li>transactionGrain : Таблица фактов с детализацией на уровне транзакций содержит данные транзакций (одна строка на событие транзакции)</li> <li>accumulatedSnapShot : Таблица фактов с накопительным снимком содержит данные жизненного цикла, которые обновляются при каждом изменении состояния (одна строка на жизненный цикл)</li> <li>periodicSnapShot : Таблица фактов с периодическим снимком содержит</li> </ul>

Атрибут	Обязательный	По умолчанию	Описание
			данные производительности с регулярными интервалами (одна строка на период)
subType	Нет		Подтип детализации таблицы фактов. Пример rate:5-minute event:as-pollled  <b>Примечание.</b> Если установлен подтип event:as-pollled, дублированные строки в таблице подтипов не будут удаляться автоматически
isExternal	Нет	false	Обработка внешней таблицы (или представления): если выбрано значение true, схема таблицы указана (элементы <a href="#">attribute</a> ), но таблица не создается и управляется платформой.

Дочерние элементы	Множество	Описание
<a href="#">column</a>	0..n	Список столбцов таблицы

**Примечание.** Определение таблицы фактов должно включать определение следующих двух столбцов

dsi\_key\_id\_: Столбец внешнего ключа для основной таблицы измерений

ta\_period: Столбец периода

#### Элемент XML: **dimensionTable**

Элемент dimensionTable представляет физическую таблицу хранилища данных, содержащую данные и атрибуты измерений.

**Примечание.** Основной ключ (суррогатный ключ) таблицы измерений по умолчанию создается платформой.

Атрибут	Обязательный	По умолчанию	Описание
name	Да		Уникальное имя таблицы измерений
caption	Да		Метка таблицы измерений.



Атрибут	Обязательный	По умолчанию	Описание
version	Да		Версия таблицы модели
description	Нет		Краткое описание таблицы измерений
type	Да		Тип таблицы измерений
conformedTo	Нет		<p>Главная таблица измерений, которой должна соответствовать эта таблица измерений.</p> <p>Таблица измерений всегда содержит подмножество строк из главной таблицы измерений.</p> <p>Столбец бизнес-ключей, заданный в этой таблице, должен соответствовать столбцам бизнес-ключей, заданным в главной таблице измерений.</p> <p>Главная таблица измерений может быть задана в той же схеме или использовать зависимую схему. Если используется зависимая схема, главная таблица измерений должна включать имя зависимой схемы как префикс, отделенный двоеточием ":"</p>
isExternal	Нет	false	Обработка внешней таблицы (или представления): если выбрано значение true, указывается только схема таблицы (элементы <a href="#">attribute</a> ), но таблица не создается и управляется платформой.

Дочерние элементы	Множество	Описание
<a href="#">column</a>	0..n	Список столбцов таблицы

#### Элемент XML: **bridgeTable**

Элемент `bridgeTable` представляет физическую таблицу хранилища данных, которая действует как мостовая таблица переменной глубины и содержит отношения родительский-дочерний между ЭК.

Атрибут	Обязательный	По умолчанию	Описание
name	Да		Уникальное имя мостовой таблицы
caption	Да		Метка мостовой таблицы.

Атрибут	Обязательный	По умолчанию	Описание
version	Да		Версия таблицы модели
description	Нет		Краткое описание таблицы
type	Да		Тип таблицы (мостовая)
isExternal	Нет	false	Обработка внешней таблицы (или представления): если выбрано значение true, схема таблицы указана (элементы <a href="#">attribute</a> ), но таблица не создается и управляется платформой.
subType	Нет	variable	Тип переменной таблицы. Допустимые значения variable — переменная глубина fixed — фиксированная глубина (в настоящее время не поддерживается)

Дочерние элементы	Множество	Описание
<a href="#">column</a>	0..n	Список столбцов таблицы

**Примечание.**

- Мостовая таблица с переменной глубиной должна включать два столбца бизнес-ключей, ссылающихся на одну таблицу измерений, представляющую отношение родительский-дочерний. Мостовая таблица фиксированной глубины может включать только одно такое отношение.
- Мостовая таблица с переменной глубиной может включать любое количество столбцов бизнес-ключей в дополнение к двум столбцам, вовлеченным в отношение родительский-дочерний.

**Элемент XML: [column](#)**

Элемент [column](#) определяет физические столбцы таблицы

Атрибут	Обязательный	По умолчанию	Описание
name	Да		Имя столбца (уникальное для таблицы)
caption	Да		Метка столбца.
dataType	Да, если это не столбец ссылки	Целое число в случае столбца	Тип данных столбца (независимый от БД). Допустимые значения String

Атрибут	Обязательный	По умолчанию	Описание
		ссылки	Time Date Integer Float Double varbinary
subType	Нет		Технологические сведения о типе
size	Нет	255 для типа String	Размер типа данных
reference	Нет		Столбец внешнего ключа. Если указанная таблица не задана локально в схеме, она должна включать имя схемы как префикс, отделенный двоеточием ":"
description	Нет		Описание столбца
default	Нет		Значение столбца по умолчанию. Примечание. Строковые значения должны быть указаны с одинарными кавычками Пример: default="default-value"
notNull	Нет	false	Разрешено ли значение null
businessKey	Нет	false	Бизнес-ключ (уникальный)

### Логическая модель

Логическая модель включает логические кубы, показатели, измерения, атрибуты измерений, иерархии измерений и уровни иерархий. Определение логической модели используется при создании пространства ВО для создания отчетов.

Элемент XML: **logical**

### Определение логической схемы

Дочерние элементы	Множество	Описание
<a href="#">dimension</a>	0..n	Список глобальных измерений логической модели
<a href="#">cube</a>	0..n	Список кубов логической модели
<a href="#">cubeRef</a>	0..n	Определяет ссылку на куб, заданный в зависимой схеме

Дочерние элементы	Множество	Описание
<a href="#">cubeRegion</a>	0..n	Определяет куб, полученный из исходного куба, заданного в зависимой схеме как его подмножество.

Элемент XML: **dimension**:

Элемент `dimension` определяет набор уникальных атрибутов, которые оценивают данные фактов и иерархии измерений.

Атрибут	Обязательный	Описание
<code>name</code>	Да	Уникальное имя измерения
<code>caption</code>	Да	Метка измерения
<code>description</code>	Нет	Описание измерения

Дочерние элементы	Множество	Описание
<a href="#">hierarchy</a>	1..n	Иерархии измерений
<a href="#">memberGroup</a>	0..n	Группа участников

Элемент XML: **hierarchy**

Элемент иерархии определяет иерархию измерений. Иерархия измерений определяет, как организованы данные на разных уровнях сведения

Атрибут	Обязательный	По умолчанию	Описание
<code>name</code>	Да		Уникальное имя иерархии
<code>caption</code>	Да		Метка иерархии.
<code>description</code>	Нет		Описание иерархии
<code>createLevels</code>	Нет	<code>true</code>	Когда элементу <code>CreateLevels</code> присвоено значение <code>false</code> , иерархия не будет создана в пространстве <code>BO</code> . Эта функция часто используется для одноуровневых иерархий.

Дочерние элементы	Множество	Описание
<a href="#">level</a>	1..n	Уровень иерархии

Элемент XML: **level**

Элемент `level` определяет уровень иерархии. Уровень иерархии представляет позицию в иерархии. Каждый уровень более детален, чем его родительский уровень. Например, иерархия измерений времени может иметь такие уровни, как день, месяц, квартал и год.

Атрибут	Обязательный	Описание
name	Да	Уникальное имя уровня иерархии
caption	Да	Метка уровня иерархии
таблица	Да	Таблица измерений, используемая для уровня
levelCaption	Да	<p>Определяет название уровня. Название уровня используется для отображения уровня детализации в отчетах. Оно должно быть разборчиво для человека и уникально идентифицировать участников определенного уровня.</p> <p>Примечание. В пространстве BO levelCaption используется как для оперативного сведения, так и для отображения уровня в иерархиях.</p>

Дочерние элементы	Множество	Описание
<a href="#">member</a>	0..n	Атрибут уровня иерархии (из таблицы измерений, заданной для уровня)
<a href="#">calculatedMember</a>	0..n	Рассчитывается из других столбцов таблицы измерений
<a href="#">group</a>	0..n	Группа участников

#### Элемент XML: **member**

Элемент `member` определяет участника уровня. Участник уровня сопоставляет логическое имя участника с физическим столбцом, заданным в таблице измерений.

Атрибут	Обязательный	Описание
name	Да	Имя участника.
column	Да	Имя атрибута (как указано в таблице измерений)
caption	Да	Метка участника
description	Нет	Описание участника

#### Элемент XML: **calculatedMember**

Элемент `calculatedMember` определяет рассчитанного участника. Рассчитанный участник сопоставляет логическое имя участника и SQL-выражение, включающее столбцы из политики измерений

Атрибут	Обязательный	Описание
name	Да	Имя участника.

Атрибут	Обязательный	Описание
caption	Да	Метка участника.
formula	Да	Стандартное SQL-выражение, включающее столбцы из политики измерений Пример: \${member1} + \${ member2} (\${member1} + \${member2})/\${member3}
description	Нет	Описание рассчитанного участника
dataType	Нет	Тип данных рассчитанного участника, по умолчанию String

Элемент XML: **group**

Элемент group ссылается на группу участников, идентифицированную по имени группы участников.

Атрибут	Обязательный	Описание
name	Да	Имя группы участников.

Элемент XML: **memberGroup**

Элемент memberGroup группирует участников под именем. Участники, заданные в группе, будут вставлены в элемент level как есть.

Атрибут	Обязательный	Описание
name	Да	Имя группы участников.

Дочерние элементы	Множество	Описание
<a href="#">member</a>	0..n	Атрибут уровня иерархии (из таблицы измерений, заданной для уровня)
<a href="#">calculatedMember</a>	0..n	Рассчитывается из других столбцов таблицы измерений

Элемент XML: **cube**

Элемент cube определяет куб данных. Куб — это абстрактное представление многомерного набора данных. Куб содержит такие показатели бизнеса, как продажи с отбором по измерениям, таким как продукт, магазин и город. Куб представляет собой схему «звезда» (снежинка) в реляционной модели. Схема «звезда» включает таблицу фактов в центре и одну или несколько таблиц измерений, на которую ссылается таблица фактов. Схема «снежинка» — это расширение схемы «звезда», которое подразумевают нормализацию измерений в несколько связанных таблиц измерений.

Атрибут	Обязательный	Описание
name	Да	Уникальное имя куба
caption	Да	Метка куба.
description	Нет	Описание куба

Дочерние элементы	Множество	Описание
<a href="#">measures</a>	1	Показатели куба (из таблицы фактов)
<a href="#">dimension</a>	0..n	Локальные измерения, используемые для куба.
<a href="#">dimensionRef</a>	0..n	Ссылка на глобальное измерение, используемое для куба.

Элемент XML: **dimensionRef**

Элемент dimensionRef ссылается на элемент глобального измерения

Атрибут	Обязательный	Описание
name	Да	Псевдоним измерения (scope=cube)
reference	Да	Имя измерения по ссылке. Примечание. Если ссылка на измерение осуществляется из зависимой схемы, имя измерения должно включать имя схемы как префикс, отделенный двоеточием ":"

Элемент XML: **measures:**

Элемент measures определяет список элементов measure

Атрибут	Обязательный	Описание
factTable	Да	Таблица фактов, на которую ссылается реляционный раздел
caption	Нет	Название показателей

Дочерние элементы	Множество	Описание
<a href="#">measure</a>	1..n	Показатели куба (из таблицы фактов)
<a href="#">calculatedMeasure</a>	0..n	Расчетные показатели куба (из measures)

Элемент XML: **показатель**

Элемент measure сопоставляет логическое имя показателя с именем столбца таблицы фактов

Атрибут	Обязательный	Описание
name	Да	Имя участника.
column	Да	Указанное имя столбца из таблицы фактов
caption	Да	Метка участника.
aggregation	Да	Правило сведения по умолчанию, используемое для детализации и сведения иерархии. Допустимые значения: sum, min, max, average, count и none.
description	Нет	Описание показателя

Элемент XML: **calculatedMeasure**

Элемент calculatedMeasure сопоставляет логическое имя показателя с SQL-выражением, включающим показатели

Атрибут	Обязательный	Описание
name	Да	Имя показателя
caption	Да	Метка расчетного показателя.
formula	Да	Стандартное SQL-выражение, включающее показатели. Пример: \${measure1} + \${measure2} (\${measure1} + \${measure2})/\${measure3}
description	Нет	Описание рассчитанного показателя
dataType	Нет	Тип данных рассчитанного показателя

Элемент XML: **cubeRef**

Элемент cubeRef ссылается на куб и соответствующие сводные показатели, заданные в зависимой схеме

Атрибут	Обязательный	Описание
name	Да	Уникальное имя куба
reference	Да	Куб, указанный в родительской/зависимой схеме

Элемент XML: **cubeRegion**

Элемент cubeRegion определяет куб, полученный из исходного куба, заданного в зависимой схеме, как его подмножество

Атрибут	Обязательный	Описание
name	Да	Уникальное имя куба



Атрибут	Обязательный	Описание
reference	Да	Куб, указанный в родительской схеме

Дочерние элементы	Множество	Описание
<a href="#">derivedMeasures</a>	1..1	Показатели для куба, полученные из исходного куба
<a href="#">aggrefRef</a>	0..n	Ссылка на сводный показатель, используемый для исходного куба для наследования.

Элемент XML: **derivedMeasures**:

Элемент `derivedMeasures` определяет набор показателей, полученных из исходного куба, заданного в зависимой схеме как его подмножество.

Атрибут	Обязательный	Описание
caption	Нет	Метка производных показателей

Дочерние элементы	Множество	Описание
<a href="#">measureRef</a>	1..n	Показатели для куба, полученные из исходного куба

Элемент XML: **aggrefRef**

Элемент `aggrefRef` определяет набор сводных показателей, полученных из исходного куба, заданного в зависимой схеме как его подмножество.

Атрибут	Обязательный	Описание
name	Да	Имя сводного показателя
reference	Да	Сводный показатель, указанный в родительской схеме

Элемент XML: **measureRef**

Элемент `measureRef` определяет набор показателей/расчетных показателей, полученных из исходного куба, заданного в зависимой схеме как его подмножество.

Атрибут	Обязательный	Описание
name	Да	Имя показателя/расчетного показателя для куба, полученные из исходного куба
caption	Нет	Метка показателя

Элемент XML: **aggregates**

Элемент `aggregates` содержит список элементов `aggregate` и `forecast`.

Дочерние элементы	Множество	Описание
<a href="#">aggregate</a>	0..n	Список сводных таблиц
<a href="#">forecast</a>	0..n	Список таблиц сведения прогнозов

Элемент XML: **aggregate**

Элемент `aggregate` определяет правила, которые применяются к таблице фактов перед сведением

Атрибут	Обязательный	По умолчанию	Описание
<code>name</code>	Да		Имя сводной таблицы.
<code>caption</code>	Да		Метка сводной таблицы.
<code>cube</code>	Да		Имя куба (локального) для которого выполняется сведение. (указывается как элемент <code>cube</code> )
<code>source</code>	Да		Исходная таблица для сведения
<code>isExternal</code>	Нет	<code>false</code>	Если выбрано значение “true”, сводная таблица не будет создана
<code>type</code>	Нет	<code>trend_sum</code>	Тип сведения. Допустимые значения <ul style="list-style-type: none"> <li><code>trend_sum</code> — предварительное сведение не выполняется (будет создана сводная таблица)</li> <li><code>OLAP</code> — не выполняет оперативное сведение (сводная таблица не будет создана)</li> </ul>

Дочерние элементы	Множество	Описание
<a href="#">aggMeasure</a>	1..n	Сводный показатель
<a href="#">calculatedMeasure</a>	1..n	Сводные показатели для сводной таблицы
<a href="#">aggLevel</a>	0..n	Уровень, на котором выполняется сведение Порядок имеет значение (порядок Groupby)

Элемент XML: **aggMeasure**

Элемент `aggMeasure` определяет столбец сводных показателей в сводной таблицы и тип сведения, выполняемый для исходного столбца

Атрибут	Обязательный	Описание
<code>name</code>	Да	Имя сводного показателя. При создании сводного

Атрибут	Обязательный	Описание
		<p>столбца к функции сведения будет добавлено значение атрибута name в качестве префикса. Например, если имя — “cpu_util”, а сводная функция — “avg”.</p> <p>Примечание. Если к значению атрибута name добавляется префикс “!”, то созданный сводный столбец будет таким же, как значение атрибута name.</p>
source	Да	Имя столбца из исходной таблицы, для которого выполняется сведение.
caption	Да	Название сводного показателя
aggregation	Да	<p>Функция сведения. Допустимые значения</p> <ul style="list-style-type: none"> <li>Если aggMeasure является частью элемента aggregate, поддерживаемые функции: avg, min, max, cnt, tot, med, std, slope, wav, perXX, lst, nlst.</li> <li>Если aggMeasure является частью элемента forecast, поддерживаемые функции: avg, min, max, cnt, tot, med, std, slope, wav, perXX, fXX, DTT [XX].</li> <li>Если тип сведения — OLAP, поддерживаемые функции: min, max, avg, sum.</li> </ul> <p>См. список описаний этих функций в таблице <a href="#">“Поддерживаемые функции в элементах Aggregate и Forecast” на странице 160</a> в приложении.</p>
description	Нет	Описание сводного показателя

#### Элемент XML: **calculatedMeasure**

Элемент calculatedMeasure определяет столбец сводных показателей в сводной таблицы и тип сведения, выполняемый для в результате SQL-выражения.

Атрибут	Обязательный	Описание
name	Да	Имя показателя
caption	Да	Метка расчетного показателя.
aggregation	Да	<p>Функция сведения. Допустимые значения</p> <ul style="list-style-type: none"> <li>Если тип сведения: trend_sum tot,min,max,avg,lst,per95,Per90,wav(),dtt[100],f30,f60,f90</li> <li>Если тип сведения: OLAP min,max,avg,sum</li> </ul>
formula	Да	Стандартное SQL-выражение, включающее столбцы из

Атрибут	Обязательный	Описание
		таблицы фактов/измерений
description	Нет	Описание рассчитанного показателя
dataType	Нет	Тип данных сводного показателя, значение по умолчанию Double

#### Элемент XML: **aggLevel**

Элемент `aggLevel` представляет уровень измерения, для которого выполняется сведение

Атрибут	Обязательный	Описание
dimension	Да	Имя исходного измерения для указанного куба (задается в элементе <a href="#">dimension</a> ).
hierarchy	Да	Имя исходной иерархии для указанного измерения (задается в элементе <a href="#">hierarchy</a> ).
level	Да	Имя уровня сведения для указанной иерархии (задается в элементе <a href="#">level</a> ).

#### Элемент XML: **forecast**

Элемент `forecast` определяет прогноз для факта. Прогноз — это расширение сводного показателя (для сведения прогноза) с дополнительным атрибутом `baselinedays`

Атрибут	Обязательный	Описание
baselinedays	Да	Количество дней, используемое для базового показателя. (например, 42-дневный базовый период прогноза)

## Сопоставление компонентов между XML-моделью и пространством VO

XML-модель используется как входные данные при формировании пространства VO.

Платформа CPC в SHR обрабатывает XML-модель и формирует пространство VO.

Пространство VO — это файл, который служит семантическим уровнем между внутренней базой данной и зависимыми отчетами. Файл пространства VO содержит следующие компоненты

- Классы, класс — это группа объектов пространства. Он предоставляет структуру для пространства. Обычно класс содержит групповые объекты показателей или группу из соответствующего измерения и детальных объектов. Класс может включать подклассы, что позволяет группировать объекты в более детальные поднаборы.
- Объекты — объекты обозначают столбцы, заданные в таблицах и представлениях баз данных

- Таблицы — физические таблицы хранилища данных
- Запросы Join — SQL-выражения, описывающие связь таблиц
- Контексты — группы связанных таблиц с соответствующими запросами join
- Иерархия — упорядоченная последовательность объектов измерений, которая позволяет пользователям анализировать данные на определенном уровне и выполнять детализацию на более низком уровне для получения более подробных данных

В таблице ниже представлено сопоставление элементов XML-модели в компонентами пространства BO.

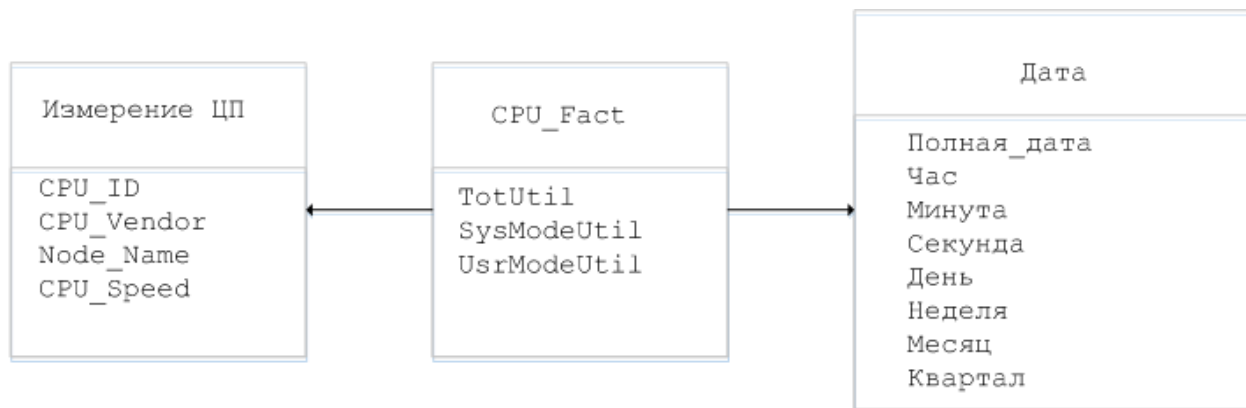
Элемент XML-модели	Компонент пространства BO
<a href="#">cube</a>	Класс, содержащий объекты показателей
<a href="#">dimension</a>	Класс, содержащий объекты измерений
<a href="#">hierarchy</a>	Иерархия
Атрибут <a href="#">levelCaption</a> в элементе <a href="#">level</a>	Объект измерения
Элементы <a href="#">member</a> , заданные в элементе <a href="#">level</a>	Детальные объекты в объекте измерения
Элемент <a href="#">memberGroup</a> , указанный в элементе <a href="#">level</a>	Детальные объекты в объекте измерения
<a href="#">factTable</a> , <a href="#">dimensionTable</a> , <a href="#">bridgeTable</a> , <a href="#">aggregate</a>	Таблицы

Запросы join и контексты пространства BO создаются из ссылок внешних ключей, заданных [реляционном разделе](#), и уровня сведения, заданного в разделе [сведения](#).

## Примеры определения модели

### Пример файла XML-модели

Рассмотрим простую модель измерений для управления производительностью ЦП



Эта модель измерений содержит факт CPU и три измерения, которые служат для отбора факта CPU.

Эквивалентный XML-файл определения модели как правило будет содержать следующее

Реляционный раздел

- [Определение таблицы фактов](#) для факта CPU
- [Определение таблицы измерений](#) для измерения CPU
- [Определение таблицы измерений](#) для измерения Date

Логический раздел

- Определение [измерения](#) и [иерархии](#) для измерения CPU
- Определение [измерения](#) и [иерархии](#) для измерения Date
- [Определение куба](#) и определение ссылки измерения для факта CPU

Сводный раздел

- Почасовое [определение сведения](#) для факта CPU
- Ежедневное [определение сведения](#) для факта CPU
- Прогнозируемое [определение сведения](#) для факта CPU

См. модель измерений, описанную выше, в файле определения модели **CPU\_Performance\_Management.xml** в разделе вложений

## Ссылки на элементы в зависимом файле XML-модели

XML-модели может ссылаться на элементы, заданные в другой зависимой XML-модели по имени схемы зависимой XML-модели в качестве префикса и имени элемента. Элементы, для которых допускаются ссылки: dimensionTable, dimension и cube

### Model1.xml

```
<schema name="schema1">
 <relational>
 <dimensionTable name="dimension_table1".....>
```

```

.....
</dimensionTable>
<dimensionTable name="dimension_table2".....>
.....
</dimensionTable>
<factTable name="fact_table1".....>
.....
</factTable>
</relational>
<logical>
<dimension caption="Dimension1" name="Dimension1">
.....
<hierarchy name="Dimension1Hierarchy".....>
 <level name="Dim1Level1".....>
.....
 </level>
</hierarchy>
</dimension>
<cube name="Fact1" caption="Fact1" description="Fact1 description">
.....
</cube>
</logical>
</schema>

```

**Model2.xml**

```

<schema name="schema2">
<relational>
 <dimensionTable name="dimension_table3".....>
 <! -- Referring a dimension table defined in dependent schema-->
 <column name="dim_attribute1" reference="schema1:dimension_table2"
..... />
 </dimensionTable>
 <factTable name="fact_table2".....>
.....
</factTable>

```

```

</relational>
<logical>
 <dimension caption="Dimension2" name="Dimension2">

 <hierarchy name="Dimension2Hierarchy".....>

 </hierarchy>
 </dimension>
 <cube name="Fact2" caption="Fact2" description="Fact2 description">

 <! -- Referring a dimension defined in dependent schema-->
 <dimensionRef name="Dimension1" reference="schema1:Dimension1" />
 </cube>
 <! -- Referring a cube defined in dependent schema-->
 <cubeRef name="Fact1" reference="schema1:Fact1"/>
</logical>
</schema>

```

## Задание иерархий и уровней

Рассмотрим измерение Date. Иерархия измерения Date, которая включает данные на уровне лет, месяцев, дней и часов и может быть определена следующим образом

```

<dimension caption="DATETIME" name="DATETIME">
 <hierarchy caption="DATETIME Hierarchy" name="DATETIMEH"
 description="DATETIME hierarchy ">
 <level caption="Year" name="Year" table="DATETIME"
 levelCaption="Year">
 <member name="Year" column="TIME_YEAR_NUMBER"
 caption="Year" description="Year" />
 </level>
 <level caption="Month" name="Month" table="DATETIME"
 levelCaption="Month">
 <member name="Month" column="TIME_MONTH_NAME"

```



```
 caption="Month" description="Month Name"/>
</level>
 <level caption="Day" name="Day" table="DATETIME"
 levelCaption="Day">
 <member name="Day" column="TIME_DAY_MONTH_NUMBER"
 caption="Day" description="Day" />
<member name="Day Name" column="TIME_DAY_NAME"
 caption="Day Name" description="Day Name" />
 </level>
 <level caption="Hour" name="Hour" table="DATETIME"
 levelCaption="Hour">
<member name="Hour" column="TIME_HOUR_ID" caption="Hour"
 description="Hour" />
 </level>
</hierarchy>
</dimension>
```

Политики сбора данных ETL

# Глава 7: Определение XML политики сбора данных RTSM

XML политики сбора данных RTSM определяет типы ЭК и атрибуты представления, которые должны собираться из RTSM. SHR использует политику сбора данных RTSM для извлечения топологии и данных измерений для домена с компьютера BSM.

## Структура XML

```
< etldefinition >
 < views >
 < view >
 < citype >
 < aliassource >
 < aliastarget />
 </ aliassource >
 < ciattribute />
 < c i r e f >
 < ciattribute />
 </ c i r e f >
 </ ciattribute >
 </ view >
 </ view >
 </ etldefinition >
```

Элемент XML: **etldefinition**

etldefinition — это корневой элемент определения политики сбора данных CMDB. Этот элемент определяет тип сбора данных и домен, для которого он применяется

Атрибут	Обязательный	Описание
type	Да	Определяет шаг ETL. Должен иметь значение “collect” для политики сбора данных CMDB
collector	Да	Тип сборщика. Должен иметь значение “CMDB” для политики сбора данных CMDB
domain_name	Да	Домен, для которого выполняется сбор данных

Атрибут	Обязательный	Описание
contentpack_name	Нет	Имя пакета содержимого для домена
description	Нет	Краткое описание политики сбора данных

Дочерние элементы	Множество	Описание
<a href="#">views</a>	1..1	Определение представлений CMDB

Элемент XML: **views**

#### Определение представлений CMDB

Атрибут	Обязательный	Описание
description	Нет	Описание группирования представлений

Дочерние элементы	Множество	Описание
<a href="#">view</a>	1..n	Список представлений CMD, для которых выполняется сбор данных

Элемент XML: **view**

Элемент view определяет представление CMDB

Атрибут	Обязательный	Описание
name	Да	Имя представления CMDB
dumplocation	Нет	dumplocation
description	Нет	Краткое описание представления

Дочерние элементы	Множество	Описание
<a href="#">citype</a>	1..n	Список типов ЭК для сбора

Элемент XML: **citype**

Элемент citype определяет тип ЭК. Тип ЭК — это именованная группа элементов конфигурации (ЭК — управляемый компонент, например узел, программное обеспечение).

Атрибут	Обязательный	Описание
name	Да	Имя типа ЭК, указанное в RTSM (BSM Modelling Studio > Диспетчер типов ЭК).
alias	Нет	Псевдоним типа ЭК

Атрибут	Обязательный	Описание
filter	Нет	Условие фильтрации строк. См. полный перечень поддерживаемых фильтров и их синтаксис в <a href="#">приложении А</a>
type	Нет	Указывает тип ЭК: измерение ОС/источник данных/оба. Допустимые значения <ul style="list-style-type: none"> <li>dimension — ЭК этого типа считаются измерениями, но не считаются источниками данных (создается CSV-файл измерений, но данные не вводятся в таблицу словаря источников данных SHR)</li> <li>datasource — ЭК этого типа считаются источниками данных, но не считаются измерениями (данные вводятся в таблицу словаря источников данных SHR, но CSV-файл измерений не создается)</li> <li>both — ЭК этого типа считаются источниками данных и измерениями (данные вводятся в таблицу словаря источников данных SHR, и создается CSV-файл измерений) Значение по умолчанию: “both”.</li> </ul>
description	Нет	Краткое описание типа ЭК

Дочерние элементы	Множество	Описание
<a href="#">aliassource</a>	0..n	Псевдонимы имени CSV-файла
<a href="#">ciattribute</a>	1..n	Определение атрибута ЭК
<a href="#">ciref</a>	1..n	Определение ссылки ЭК

#### Элемент XML: **aliassource**

Элемент `aliassource` используется для определения псевдонимов CSV-файла для собираемых данных, что позволяет иметь несколько копий одних и тех же данных в виде нескольких файлов. Это позволяет выполнять несколько независимых операций для одних и тех же данных.

**Примечание.** Если элемент `aliassource` не задан, имя созданного файла будет соответствовать шаблону

\* `viewName _0_ citypeName _0_*.csv`

Атрибут	Обязательный	Описание
description	Нет	Краткое описание источника псевдонима

Дочерние элементы	Множество	Описание
<a href="#">aliastarget</a>	1..n	Шаблоны имен целевых CSV-файлов

#### Элемент XML: **aliastarget**

Псевдонимы целевого CSV-файла задаются с помощью элемента `aliastarget`.

Атрибут	Обязательный	Описание
<code>type</code>	Да	Пользовательский псевдоним типа ЭК
<code>category</code>	Да	Пользовательский псевдоним представления
<code>description</code>	Нет	Описание псевдонима

**Примечание.** Значения атрибутов `type` и `category` будут использоваться в имени созданного CSV-файла. Шаблон имени файла: `*category_0_type_0*.csv`.

Если два элемента `citype` указывают одинаковые тип и категорию в поле `aliastarget`, данные из обоих типов ЭК будут объединены в одном файле.

Элемент XML: **`ciref`**

Элемент `ciref` используется для указания атрибутов ЭК из другого типа ЭК.

**Примечание.** Тип ЭК может ссылаться только на атрибуты ЭК типов ЭК, которые заданы в одной иерархии с текущим типом ЭК.

Атрибут	Обязательный	Описание
<code>name</code>	Да	Имя типа ЭК по ссылке
<code>relationName</code>	Нет	Представляет тип отношения между родительским ЭК и ЭК по ссылке. Когда значение <code>relationName</code> , заданное в представлении RTSM (BSM Modelling Studio) указывается в политике сбора данных, только экземпляры ЭК, связанные с родительским ЭК с определенным значением <code>relationName</code> , собираются сборщиком SHR RTSM.
<code>description</code>	Нет	Описание ссылки

Дочерние элементы	Множество	Описание
<code>ciattribute</code>	1..n	Определение атрибута ЭК

Элемент XML: **`ciattribute`**

Элемент `ciattribute` определяет атрибут ЭК.

Атрибут	Обязательный	Описание
<code>name</code>	Да	Имя атрибута, указанное в RTSM (BSM Modelling Studio > Диспетчер типов ЭК).
<code>datatype</code>	Да	Тип данных атрибута Примечание: Не используется

Атрибут	Обязательный	Описание
csvColumnName	Нет	Имя целевого столбца CSV для атрибута ЭК. Если этот элемент не задан, в качестве имени столбца CSV используется имя атрибута.
description	Нет	Описание атрибута

## Примеры политики сбора данных RTSM

### Пример XML-файла политики сбора данных RTSM

Рассмотрим задание политики сбора данных CMDB для базы данных Oracle на предприятии. Политика сбора данных CMDB может быть задана с помощью типов ЭК в представлении CMDB “ORA\_Deployment”, предоставленном ORACLE DBSPI. См. пример файла политики сбора данных CMDB uCMDB\_Collection\_Policy.xml в разделе вложений для базы данных Oracle. Политика сбора данных определяет типы ЭК nt, unix и oracle, при этом тип ЭК oracle ссылается на атрибуты в типах ЭК nt и unix.

## Структура XML политики сбора данных SiteScope API

```
< etldefinition >
< category >
< monitor >
< aliassource >
 < aliastarget />
</ aliassource >
< counter >
< csvcolumn >
</ counter >
</ etldefinition >
```

#### Элемент XML: **etldefinition**

etldefinition — это корневой элемент определения политики сбора данных SiteScope API. Этот элемент определяет тип сбора данных и домен, для которого он применяется

Атрибут	Обязательный	Описание
type	Да	

Атрибут	Обязательный	Описание
		Определяет шаг ETL. Должен иметь значение "SIS" для политики сбора данных SiteScope
collector	Да	Тип сборщика. Должен иметь значение "SIS" для политики сбора данных SiteScope

Элемент XML: **Категория**

Дочерние элементы	Описание
Категория	Логическое имя, назначенное группе мониторов для сбора.

Элемент XML: **Монитор**

Атрибут	Обязательный	Описание
type	Да	Тип монитора SiteScope для сбора. Это имя должно соответствовать имени в SiteScope.
summarizeinterval	Да	Интервал сведения, который будет использоваться для всех операций сведения для отдельных показателей.

Элемент XML: **aliassource**

Элемент **aliassource** используется для определения псевдонимов CSV-файла для собираемых данных, что позволяет иметь несколько копий одних и тех же данных в виде нескольких файлов. Это позволяет выполнять несколько независимых операций для одних и тех же данных.

Атрибут	Обязательный	Описание
aliasSourceType	Да	Тип и категория источника, для которого задается псевдоним

Элемент XML: **aliastarget**

Псевдонимы целевого CSV-файла задаются с помощью элемента **aliastarget**.

Атрибут	Обязательный	Описание
aliastargettype	Да	Тип псевдонима отдельного целевого объекта и категория источника

Элемент XML: **counter**

Элемент счетчика для XML-файла SiteScope

Атрибут	Обязательный	Описание
filter	да	Регулярное выражение Java, применяемое к атрибуту источника
source	да	Имя атрибута монитора, к которому следует применить регулярное выражение. Значение атрибута по умолчанию: name.

Дочерние элементы	Описание
csvcolumn	<p>Элемент, определяющий имя столбца CSV-файла</p> <p>Включает следующие атрибуты:</p> <ul style="list-style-type: none"> <li>• name: Имя целевого столбца</li> <li>• source: Имя исходного столбца</li> </ul>



# Глава 8: Определение XML политики сбора данных OM

XML политики сбора данных OM определяет используемый источник данных PA и класс PA, из которого собираются данные измерений. SHR подключается к компоненту Operations Manager, настроенному в Admin UI, чтобы получить управляемые узлы и выполнить обнаружение измерений на этих узлах.

## Структура XML

```
< etldefinition >
 < sncollection >
 < rule >
 < alias source >
 < aliastarget />
 </ alias source >
 < mapping />
 </ rule >
 </ sncollection >
</ etldefinition >
```

Элемент XML: **etldefinition**

etldefinition — это корневой элемент определения политики сбора данных OM. Этот элемент определяет тип сбора данных и домен, для которого он применяется

Атрибут	Обязательный	Описание
type	Да	Определяет шаг ETL. Должен иметь значение “collect” для политики сбора данных OM
collector	Да	Тип сборщика. Должен иметь значение “OM” для политики сбора данных SN/OM
domain_name	Да	Домен, для которого выполняется сбор данных
contentpack_name	Нет	Имя пакета содержимого для домена
description	Нет	Краткое описание политики сбора данных

Дочерние элементы	Множество	Описание
<a href="#">sncollection</a>	1..1	Группирование правил сбора SN

Элемент XML: **sncollection**

Определение представлений

Атрибут	Обязательный	Описание
name	Да	Имя коллекции SN
mappedby	Нет	Имя источника данных PA
description	Нет	Описание коллекции SN

Дочерние элементы	Множество	Описание
<a href="#">rule</a>	1..n	Список правил сбора данных SN

Элемент XML: **rule**

Определение правила сбора данных SN

Атрибут	Обязательный	Описание
citype	Да	Пользовательское имя типа ЭК. Имя типа ЭК аналогично имени типа ЭК в CMDB.
datasource	Да	Имя источника данных PA Если имя источника данных непостоянно (например, в SAP SPI), вы можете задать шаблон (регулярное выражение), соответствующий имени источника данных.  Синтаксис PATTERN:<Java_regex>  Пример PATTERN:ABC.*  Ограничение:  Если шаблон соответствует нескольким источникам данных, классы и метрики должны совпадать во всех источниках данных.
class	Да	Имя класса PA
type	Нет	Указывает тип ЭК: измерение ОС/источник данных/оба. Допустимые значения <ul style="list-style-type: none"> <li>dimension — ЭК этого типа считаются измерениями, но не считаются источниками данных (создается CSV-файл измерений, но данные не вводятся в таблицу словаря)</li> </ul>

Атрибут	Обязательный	Описание
		<p>источников данных SHR)</p> <ul style="list-style-type: none"> <li>datasource — ЭК этого типа считаются источниками данных, но не считаются измерениями (данные вводятся в таблицу словаря источников данных SHR, но CSV-файл измерений не создается)</li> <li>both — ЭК этого типа считаются источниками данных и измерениями (данные вводятся в таблицу словаря источников данных SHR, и создается CSV-файл измерений). Значение по умолчанию: both.</li> </ul>
filter	Нет	Условие фильтрации строк. См. полный перечень поддерживаемых фильтров и их синтаксис в <a href="#">приложении A</a>
description	Нет	Описание правила SN

Дочерние элементы	Множество	Описание
<a href="#">aliassource</a>	0..n	Псевдонимы имен файлов для собранных данных
<a href="#">mapping</a>	1..n	Список сопоставлений между столбцом CSV и метрикой PA

#### Элемент XML: **aliassource**

Элемент `aliassource` используется для определения псевдонимов CSV-файла для собираемых данных, что позволяет иметь несколько копий одних и тех же данных в виде нескольких файлов. Это позволяет выполнять несколько независимых операций для одних и тех же данных.

Примечание: Если элемент `aliassource` не задан, имя созданного файла будет соответствовать шаблону

\* `collectionName_0_ cityName_0_*.csv`

Атрибут	Обязательный	Описание
description	Нет	Описание псевдонима источника

Дочерние элементы	Множество	Описание
<a href="#">aliastarget</a>	1..n	Шаблоны имен целевых CSV-файлов

#### Элемент XML: **aliastarget**

Псевдонимы целевого CSV-файла задаются с помощью элемента `aliastarget`.

Атрибут	Обязательный	Описание
type	Да	Пользовательский псевдоним типа ЭК

Атрибут	Обязательный	Описание
категория	Да	Пользовательский псевдоним представления
description	Нет	Описание псевдонима

**Примечание.** Значения атрибутов “type” и “category” будут использоваться в имени созданного CSV-файла. Шаблон имени файла: \*category\_0\_type\_0\_\*.csv.

Если два элемента **rule** указывают одинаковые тип и категорию в поле **aliastarget**, данные из обоих типов ЭК будут объединены в одном файле.

#### Элемент XML: **mapping**

Атрибут	Обязательный	Описание
source	Да	Имя столбца CSV
metric	Да	Имя метрики PA
defaultvalue	Нет	Значение по умолчанию. Далее представлены переменные замещения для значения по умолчанию <ul style="list-style-type: none"> <li>• \$agentname</li> <li>• \$currenttime</li> </ul>
description	Нет	Описание сопоставления

## Примеры политики сбора данных OM

### Пример XML-файла политики сбора данных OM

Рассмотрим задание политики сбора данных SN для базы данных Oracle на предприятии. Политику сбора данных OM можно определить с помощью источников данных PA “DBSPI\_ORA\_GRAPH”, “CODA”. См. пример файла политики сбора данных OM `OM_Collection_Policy.xml` в разделе вложений для базы данных Oracle. Политика сбора данных определяет правила сбора данных PA для типов ЭК `nt`, `unix` и `oracle CI`.

# Глава 9: Определение XML политики сбора данных OA

XML политики сбора данных OA/PA определяет источник данных PA и класс PA, из которого собираются метрики.

## Структура XML

```
< etldefini t ion >
< do m ain >
 < datasource >
 < cla s s >
 < aliassou r ce >
 < ali a starget />
 </ alia s source >
 < met r ic />
 </ cl a s s >
</ da t asour c e >
</ dom a in >
</ etldefinit i on >
```

## Элемент XML: etldefinition

etldefinition — это корневой элемент определения политики сбора данных OA/PA. Этот элемент определяет тип сбора данных и домен, для которого он применяется

Атрибут	Обязательный	Описание
type	Да	Определяет шаг ETL. Должен иметь значение “collect” для политики сбора данных PA
collector	Да	Тип сборщика. Должен иметь значение “PA” для политики сбора данных PA/OA
domain_name	Да	Домен, для которого задан сбор данных, например управление системой, производительность сети.  Примечание: Значение используется для сопоставления обнаруженных узлов для домена

Атрибут	Обязательный	Описание
		(через политику сбора данных CМДВ/ОМ) с политиков сбора данных РА, заданной для домена
contentpack_name	Нет	Имя пакета содержимого для домена
description	Нет	Краткое описание политики сбора данных РА
Дочерние элементы	Множество	Описание
<a href="#">domain</a>	1..n	Домен сбора данных

## Элемент XML: domain

### Определение домена

Атрибут	Обязательный	Описание
description	Нет	Описание домена

Дочерние элементы	Множество	Описание
<a href="#">datasource</a>	1..n	Список источников данных РА, из которых собираются данные

### Элемент XML: **datasource**

Элемент datasource определяет источник данных РА, из которого собираются данные.

Атрибут	Обязательный	Описание
name	Да	<p>Имя источника данных РА. Пример CODA, SCOPE, SPI (Oracle, MSSQL, Microsoft Exchange и др.).</p> <p>Если имя источника данных непостоянно (например, в SAP SPI), вы можете задать шаблон (регулярное выражение), соответствующий имени источника данных.</p> <p>Синтаксис</p> <p>PATTERN: &lt;Java_regex&gt;</p> <p>Пример</p> <p>PATTERN: ABC.*</p> <p>Ограничение:</p> <p>Если шаблон соответствует нескольким источникам данных, классы и метрики должны совпадать во всех источниках данных.</p>

Атрибут	Обязательный	Описание
summarized	Нет	Булево значение, обозначающее тип данных (raw или summarized) для сбора. Если выбранное значение true, собираются данные summarized. Значение по умолчанию: true.
description	Нет	Краткое описание источника источнике данных PA

Дочерние элементы	Множество	Описание
<a href="#">class</a>	1..n	Список классов PA для сбора

Элемент XML: **class**

Элемент class определяет класс PA в указанном источнике данных, из которого собираются данные.

Атрибут	Обязательный	Описание
name	Да	Имя класса PA
summarized	Нет	Булево значение, обозначающее тип данных (raw или summarized) для сбора. Если выбранное значение true, собираются данные summarized. Значение по умолчанию: true.  Примечание: Это значение переопределяет атрибут “summarized”, заданный на уровне источника данных
description	Нет	Краткое описание класса

Дочерние элементы	Множество	Описание
<a href="#">metric</a>	1..n	Список метрик, собираемых из класса PA
<a href="#">aliassource</a>	0..1	Псевдонимы имен файлов для собранных данных

Элемент XML: **metric**

Элемент class определяет метрику в классе PA для сбора

Атрибут	Обязательный	Описание
name	Да	Имя метрики, заданной в классе PA
datatype	Да	Тип данных метрики.  Примечание. Это значение не используется в настоящее время
identity	Да	Столбец ключей  Примечание. Это значение не используется в настоящее время

### Элемент XML: **aliassource**

Элемент `aliassource` используется для определения псевдонимов CSV-файла для собираемых данных, что позволяет иметь несколько копий одних и тех же данных в виде нескольких файлов. Это позволяет выполнять несколько независимых операций для одних и тех же данных.

**Примечание.** Если элемент `aliassource` не задан, имя созданного файла будет соответствовать шаблону

`PAData source_0_PAclass_0_*.csv`

Атрибут	Обязательный	Описание
<code>type</code>	Нет	Имя класса PA
<code>category</code>	Нет	Имя источника данных PA
<code>description</code>	Нет	Описание псевдонима

Дочерние элементы	Множество	Описание
<code>aliastarget</code>	1..n	Шаблоны имен целевых CSV-файлов

### Элемент XML: **aliastarget**

Псевдонимы целевого CSV-файла задаются с помощью элемента **aliastarget**.

Атрибут	Обязательный	Описание
<code>type</code>	Да	Пользовательский псевдоним класса PA
<code>category</code>	Да	Пользовательский псевдоним источника данных PA
<code>description</code>	Нет	Описание псевдонима

**Примечание.** Значения атрибутов `type` и `category` будут использоваться в имени созданного CSV-файла. Шаблон имени файла: `*category_0_type_0_*.csv`.

Если два элемента `class` указывают одинаковые тип и категорию в поле `aliastarget`, данные из обоих типов ЭК будут объединены в одном файле.

## Примеры политики сбора данных OA

### Пример XML-файла политики сбора данных OA

Предположим, что нам нужна собрать метрики с агента Performance Agent. Метрики ЦП доступны:

- В классе CPU источника данных SCOPE
- В классе CPU источника данных CODA



См. пример файла политики сбора данных PA SystemManagement\_OA\_Collection\_Policy.xml в разделе вложений, она содержит определения для сбора метрик ЦП из Performance Agent

# Глава 10: Определение XML политики сбора данных БД

Политика сбора данных БД определяет таблицы и запросы для сбора данных из различных источников данных БД, таких как:

OM, OMi, ManagementDB, ProfileDB и другие базы данных с поддержкой JDBC.

## Структура XML

```
<etldefinition >
 <queries >
 <query >
 <tables >
 <table >
 <columns >
 <column />
 </columns >
 <condition />
 </table >
 </tables >
 <genericsql >
 <statement >
 </statement >
 </genericsql >
 <joinqueries >
 <joinquery >
 <aliassource >
 <aliastarget />
 </aliassource >
 <content ></content >
 </joinquery >
 </joinqueries >
 </query >
</queries >
```

</ etldefinition >

Элемент XML: **etldefinition**

etldefinition — это корневой элемент определения политики сбора данных БД. Этот элемент определяет тип сбора данных и домен, для которого он применяется.

Атрибут	Обязательный	Описание
type	Да	Определяет шаг ETL. Должен иметь значение “collect” для политики сбора данных БД
collector	Да	Тип сборщика. Должен иметь значение “DB” для политики сбора данных БД
domain_name	Да	Домен, для которого выполняется сбор данных
contentpack_name	Нет	Имя пакета содержимого для домена
description	Нет	Краткое описание политики сбора данных

Дочерние элементы	Множество	Описание
<a href="#">queries</a>	1..1	Группа спецификаций запроса

Элемент XML: **queries**

Определение запросов

Атрибут	Обязательный	Описание
name	Да	Имя группирования запросов
description	Нет	Описание группирования запросов

Дочерние элементы	Множество	Описание
<a href="#">query</a>	1..n	Список запросов

Элемент XML: **query**

Определение запроса

Атрибут	Обязательный	Описание
name	Да	Имя запроса
type	Да	Тип запроса. Значение определяет источник БД, для которого выполняется запрос. Допустимые значения OM — запрос выполняется для базы данных OM, настроенной в AdminUI OMI — запрос выполняется для базы данных OMI,

Атрибут	Обязательный	Описание
		настроенной в AdminUI PROFILE_DATABASE — запрос выполняется для баз данных профилей, настроенных в AdminUI generic — запрос выполняется для универсальных баз данных, настроенных в AdminUI
dbtype	Да	Тип БД. Допустимые значения ORACLE SYBASEIQ MSSQL  <b>Примечание.</b> Если для атрибута type установлено значение "PROFILE_DATABASE", dbtype может принимать значения "ORACLE" или "MSSQL"
description	Нет	Краткое описание запроса

Дочерние элементы	Множество	Описание
<a href="#">tables</a>	1..1	Группирование таблиц для запроса
<a href="#">joinqueries</a>	1..1	Группирование запросов join
<a href="#">genericsql</a>	1..n	Универсальные операторы SQL для выполнения

#### Элемент XML: **genericsql**

Элемент универсального SQL-запроса определяет SQL-запрос, который должен быть выполнен для источника данных (OM, OMI, универсальная база данных и др.)

Атрибут	Обязательный	Описание
name	Да	Имя универсального запроса, уникального идентифицирующего определенные объекты для сценариев с добавочным сбором данных.
timelag	Нет	Количество часов, на которое необходимо вернуться (с максимального времени последнего сбора) для сбора данных, поздно полученных от источника. Например, если установлена задержка 1 час и максимальное время последнего сбора составляет 21:00:00, в 22:00:00 данные будут собраны для периода с 20:00:00 по 22:00:00.
initialhistory	Нет	Объем исторических данных, собираемых в рамках первого сбора данных. Например, если для initialhistory выбрано значение 48, при перво сборе данных будут

Атрибут	Обязательный	Описание
		<p>собраны данные за 48 часов до текущего момента.</p> <p><b>Примечание.</b> SHR регистрирует максимальное время сбора данных для каждой операции сбора данных и использует его как начальную точку при последующем сборе данных.</p>
description	Нет	Описание запроса.
категория	Нет	Пользовательское имя. Обычно обозначает источник (OMi, OM и др.).
lastcollectioncol	Нет	<p>Обозначает столбец, указанный как значение этой переменной, используется для отслеживания времени последнего сбора.</p> <p><b>Примечание.</b> Только один столбец должен использоваться как столбец последнего сбора данных. Кроме того, в указанном столбце оператор where должен быть обновлен выражением <code>&lt;column_name&gt; &gt; \$FETCH_START_TIME</code>, где <code>&lt;column_name&gt;</code> — это значение переменной.</p>
timecoltypes	Нет	<p>Содержит список столбцов периода и соответствующий тип периода (UTC/datetime). Формат списка: <code>&lt;time_col1&gt;,&lt;time_col_type&gt;::&lt;time_col2&gt;,&lt;time_col_type&gt;</code></p> <p>Пример выражения: <code>start_time,UTC::end_time,datetime</code></p> <p><b>Примечание.</b> Чтобы создать настраиваемые отчеты на основе данных, собранных из HPOM, OMi или базы данных профиля, преобразуйте данные в столбцах даты и времени в UTC.</p>
alias	Нет	Псевдоним запроса. SQL-запрос, заданный в элементе <code>joinquery</code> , должен использовать имя псевдонима вместо имени запроса.
persisttype	Да	<p>Содержит режим сохранения данных, собранных из запроса. Может принимать два значения — File и Table.</p> <p>Режим файла обозначает, что данные, возвращенные при выполнении запроса, будут сохранены непосредственно в CSV-файл с использованием указанного типа и категории.</p> <p>Режим таблицы аналогичен табличному разделу запроса, в рамках которого собираемые данные загружаются в таблицы псевдонимов, используемые в</p>

Атрибут	Обязательный	Описание
		запросах join.
type	Нет	Пользовательское имя. Обычно обозначает тип данных (fact/dimension).

Дочерние элементы	Множество	Описание
<a href="#">statement</a>	1..1	Фактические операторы SQL для выполнения

Элемент XML: **statement**

Элемент statement содержит пользовательские SQL-запросы для выполнения.

Элемент XML: **tables**

Атрибут	Обязательный	Описание
description	Нет	Краткое описание группирования таблиц

Дочерние элементы	Множество	Описание
<a href="#">table</a>	1..n	Список таблиц для запроса

Элемент XML: **table**

Элемент table определяет физическую таблицу БД из источника (OM, OMI, универсальная база данных и др.)

Атрибут	Обязательный	Описание
name	Да	Имя физической таблицы в базе данных источника
alias	Нет	Псевдоним таблицы. SQL-запрос, заданный в элементе <a href="#">joinquery</a> , должен использовать псевдоним вместо физического имени, поскольку псевдоним используется как временное имя таблицы в базе данных SHR. Если этот атрибут не задан, атрибут name используется как временное имя таблицы.
type	Да	<p>Допустимые значения</p> <ul style="list-style-type: none"> <li><code>persist</code> — выполнение запроса для таблицы и сохранение результатов во временной таблице базы данных SHR. В этом случае должен быть задан элемент <a href="#">joinquery</a> для сброса данных из временной таблицы SHR в CSV-файл.</li> </ul> <p>Эта функция обычно используется для сокращения нагрузки на исходную БД путем выполнения простого запроса для этой БД и сброса данных во временную таблицу SHR с последующим выполнением сложных запросов join для временных таблиц, чтобы сбросить</p>

Атрибут	Обязательный	Описание
		<p>нужные файлы в CSV-файл для дальнейшей обработки.</p> <ul style="list-style-type: none"> <li>• <code>dumpcsv</code> — запрос таблицы и сохранение результатов в CSV-файл. Временные таблицы не будут созданы в базе данных SHR.</li> </ul> <p>Обычно этот параметр используется, когда исходная база данных может выполнить ресурсоемкий запрос.</p>
<code>initialhistory</code>	Нет	<p>Количество часов исторических данных для сбора в рамках первого сбора данных. Например, если для <code>initialhistory</code> выбрано значение 48, при первом сборе данных будут собраны данные за 48 часов до текущего момента.</p> <p>Примечание: SHR регистрирует максимальное время сбора данных для каждой операции сбора данных и использует его как начальную точку при последующем сборе данных.</p>
<code>timelag</code>	Нет	<p>Количество часов, на которое необходимо вернуться (с максимального времени последнего сбора) для сбора данных, поздно полученных от источника. Например, если установлена задержка 1 час и максимальное время последнего сбора составляет 21:00:00, в 22:00:00 данные будут собраны для периода с 20:00:00 по 22:00:00.</p>
<code>dstype</code>	Нет	<p>Допустимое значение “BAC_MANAGEMENT”. Это значение указывает на то, что запрос должен быть выполнен для базы данных управления вместо “PROFILE_DATABASE”.</p> <p>Этот атрибут следует задавать, только если для атрибута <code>type</code> запроса установлено значение “PROFILE_DATABASE”.</p>
<code>description</code>	Нет	Описание таблицы

Дочерние элементы	Множество	Описание
<code>columns</code>	0..1	Группа столбцов
<code>condition</code>	0..1	Условное выражение SQL для фильтрации данных

Элемент XML: **columns**

Атрибут	Обязательный	Описание
<code>description</code>	Нет	Описание группирования столбцов

Дочерние элементы	Множество	Описание
<code>column</code>	1..n	Список столбцов таблицы

Элемент XML: **column**

Определение столбца таблицы

Атрибут	Обязательный	Описание
<code>name</code>	Да	Имя столбца
<code>alias</code>	Нет	Псевдоним таблицы. SQL-запрос, заданный в элементе <code>joinquery</code> , должен использовать псевдоним вместо физического имени, поскольку псевдоним используется в качестве имен столбцов во временной таблице. Если этот атрибут не задан, атрибут <code>name</code> используется в качестве имен столбцов во временной таблице.
<code>lastColColumnType</code>	Нет	Указывает, что столбец является столбцом периода. Допустимые значения <ul style="list-style-type: none"> <li><code>utc</code></li> <li><code>datetime</code></li> </ul>
<code>lastcollectiontime</code>	Нет	Обозначает столбец используется для отслеживания времени последнего сбора. Когда сборщик данных выполняет ежечасный сбор данных, он запрашивает данные со времени последнего сбора данных до текущего момента.  Допустимые значения: <code>TRUE</code> и <code>FALSE</code> , но только один столбец должен иметь атрибут <code>lastcollectiontime="TRUE"</code> .  Данные, собранные в рамках последнего сбора данных, могут быть повторены в рамках текущего сбора, что приведет к дублированию данных в CSV-файлах. Однако дубликаты отбрасываются, когда CSV-файлы загружаются в хранилище данных SHR. Вы можете предотвратить сбор дублированных данных с помощью универсального запроса SQL в политике сбора данных.
<code>description</code>	Нет	Описание столбца

Элемент XML: **condition**

Элемент `condition` определяет условное SQL-выражение для фильтрации данных в исходной таблице.

**Примечание.**



- Если для элемента **column** указан атрибут **lastcollectiontime** = "true", то атрибут **condition** для этого столбца времени будут автоматически добавлен, начиная с ключевого слова "where", в созданный SQL-запрос. Таким образом ключевое слово "where" не должно указываться при задании условия фильтра для других столбцов.
- Если столбец не отмечен как **lastcollectiontime** = "true", условие фильтра, если таковое имеется, должно быть задано с ключевым словом "where"

Атрибут	Обязательный	Описание
expression	Да	Where условие фильтрации данных (синтаксис SQL)
description	Нет	Краткое описание условия

Элемент XML: **joinqueries**

Атрибут	Обязательный	Описание
description	Нет	Описание группирования запросов join

Дочерние элементы	Множество	Описание
<a href="#">joinquery</a>	1..n	Список запросов join

Элемент XML: **joinquery**

Элемент **joinquery** определяет запрос, который будет выполнен для временных таблиц, содержащих данные, полученные из источников БД (таких как OM, OMI и универсальная БД). Все значения во временных таблицах обрабатываются как тип данных "varchar".

Атрибут	Обязательный	Описание
type	Да	Пользовательское имя. Обычно обозначает тип данных (fact/dimension).
category	Да	Пользовательское имя. Обычно обозначает источник (OMi, OM и др.).
description	Нет	Описание запроса

**Примечание.** Значения атрибутов "type" и "category" будут использоваться в имени созданного CSV-файла. Шаблон имени файла: \*category\_0\_type\_0\_\*.csv

Дочерние элементы	Множество	Описание
<a href="#">aliassource</a>	0..n	Псевдонимы имени CSV-файла
<a href="#">content</a>	1..1	SQL-запрос

Элемент XML: **aliassource**

Элемент **aliassource** используется для определения псевдонимов CSV-файла для собираемых данных, что позволяет иметь несколько копий одних и тех же данных в виде

нескольких файлов. Это позволяет выполнять несколько независимых операций для одних и тех же данных.

**Примечание.** Если элемент `aliassource` не задан, имя созданного файла будет соответствовать шаблону `joinqueryCategory _0_ joinqueryType _0_*.csv`

Атрибут	Обязательный	Описание
<code>description</code>	Нет	Описание псевдонима

Дочерние элементы	Множество	Описание
<code>aliastarget</code>	1..n	Шаблоны имен целевых CSV-файлов

Элемент XML: **aliastarget**

Псевдонимы целевого CSV-файла задаются с помощью элемента `aliastarget`.

Атрибут	Обязательный	Описание
<code>type</code>	Да	Пользовательский псевдоним типа <code>joinquery</code>
<code>категория</code>	Да	Пользовательский псевдоним категории <code>joinquery</code>
<code>description</code>	Нет	Описание псевдонима

Элемент XML: **content**

Элемент `content` содержит пользовательский SQL-запрос, содержащий условия `join`. В рамках SQL-запросов можно выполнять только строковые операции; арифметические операции не разрешены.

## Примеры политики сбора данных БД

### Пример XML-файла политики сбора данных из универсальной БД

См. пример файла политики сбора данных из универсальной БД в файле `Generic_DB_Collection_Policy.xml` в разделе вложений. Он определяет запросы для сбора данных ЦП системы из базы данных Microsoft SCOM.



# Глава 11: Определение XML политики преобразования

Политика преобразования определяет группу наборов записей с условиями фильтрации. Для каждого набора записей, заданного в XML-файле преобразования, будет создан выходной CSV-файл после процесса преобразования, содержимое CSV-файла будет определяться записями в наборе записей.

## Структура XML

```
< etldefinition >
< recordSet >
< record />
</ recordSet >
</ etldefinition >
```

## Элемент XML: etldefinition

etldefinition — это корневой элемент определения политики преобразования. Этот элемент определяет группу наборов записей с правилами преобразования.

Атрибут	Обязательный	Описание
type	Да	Определяет шаг ETL. Должен иметь значение “TRANSFORM” для политики преобразования
contentpack_name	Нет	Имя пакета содержимого
description	Нет	Краткое описание политики преобразования

Дочерние элементы	Множество	Описание
recordSet	1..n	Список наборов записей

## Элемент XML: recordSet

Элемент recordSet определяет правила преобразования для исходного шаблона CSV.

Атрибут	Обязательный	Описание
name	Да	Имя набора записей. Имя должно быть уникальным

Атрибут	Обязательный	Описание
source_type	Да	Тип источника. Это может быть класс PA/тип ЭК/пользовательский псевдоним класса PA/пользовательский псевдоним типа ЭК. Тип источника используется для идентификации исходного/входного CSV-файла для заданного набора записей
source_category	Да	Исходная категория. Это может быть источник данных PA/представление RTSM/пользовательский псевдоним источника данных PA/пользовательский псевдоним представления RTSM.  Категория источника используется для идентификации исходного/входного CSV-файла для заданного набора записей
target_type	Да	Тип целевого объекта. Может совпадать со значением source_type или иметь пользовательское значение. Тип целевого объекта будет использоваться в имени входного CSV-файла
target_category	Да	Категория целевого объекта. Может совпадать со значением source_category или иметь пользовательское значение. Категория целевого объекта будет использоваться в имени входного CSV-файла
description	Нет	Описание набора записей
condition	Нет	Условие фильтрации строк. См. полный перечень поддерживаемых фильтров и их синтаксис в <a href="#">приложении A</a>
doPivot	Нет	Булево значение, которое указывает, является ли преобразование поворотным. Значение по умолчанию: false.

- Значения атрибутов source\_type и source\_category используются для идентификации исходного/входного файла для определения recordSet. Например, если элемент recordSet определяет source\_type как "CPU" и source\_category как "CODA", исходные CSV-файлы с шаблоном имени файла \*CODA\_0\_CPU\_0\_\*.csv будут отобраны для преобразования.
- Значения атрибутов target\_type и target\_category используются для формирования имени выходного/целевого CSV-файла. Например, если элемент recordSet определяет target\_type как "CPU1" и target\_category как "CODA1", то имя выходного CSV-файла будет иметь шаблон \*CODA1\_0\_CPU1\_0\_\*.csv

Дочерние элементы	Множество	Описание
<a href="#">record</a>	0..n	Список сопоставлений столбцов между входным и

Дочерние элементы	Множество	Описание
		выходным CSV-файлами

Элемент XML: **record**

Элемент record определяет сопоставление столбцов между входным и выходным CSV-файлами с условиями фильтрации.

Атрибут	Обязательный	Описание
name	Да	Имя столбца выходного CSV-файла. Имя должно быть уникальным в пределах recordset.
source	Да	Исходный столбец, соответствующий выходному столбцу. Элемент source может быть просто именем столбца исходного/входного CSV-файла или включать одну или несколько функций преобразования. В этом случае функции будут выполнены, а окончательный результат — помещен в выходной CSV-файл. См. полный перечень поддерживаемых фильтров и их синтаксис в <a href="#">приложении A</a>
condition	Нет	Условие фильтрации строк. См. полный перечень поддерживаемых фильтров и их синтаксис в <a href="#">приложении A</a>  Примечание: Если условие не выполняется, столбец выходного CSV-файла примет значение “NotFound”
description	Нет	Описание выходного столбца

## Примеры политики преобразования

### Поворотное преобразование:

Поворотное преобразование подразумевает преобразование нескольких строк исходного CSV-файла в одну строку выходного CSV-файла в соответствии со столбцами идентификаторов. Модуль преобразования отслеживает строки с одинаковыми значениями столбца идентификаторов и объединяет их в одну строку путем слияния других столбцов. Поворотное преобразование настраивается для элемента [recordSet](#) в XML-файле преобразования путем присвоения атрибуту [doPivot](#) значения “true”. Поворотное преобразование обычно выполняется для данных, собранных из Performance Agent, когда CODA записывает метрики в виде пар ключа и значения при использовании источника данных HP SPI (DBSPI\_MSS\_REPORT(MSSQL), DBSPI\_ORA\_REPORT(Oracle) и др.).

Рассмотрим данные об использовании пространства экземпляром Oracle из Oracle SPI. Собранные CSV будут выглядеть следующим образом

DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_723914438884967.csv							
HOSTNAME	DATASOURCE	CLASSNAME	AGENTTIMESTAMP	INSTANCE	METRICID	VALUEID	VALUE
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:30	BAT92	212	1	1890
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:35	BAT92	212	1	1890
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:40	BAT92	212	1	1890
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:45	BAT92	212	1	1890
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:30	BAT92	212	2	225.48
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:35	BAT92	212	2	225.56
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:40	BAT92	212	2	225.56
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:45	BAT92	212	2	225.56
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:30	USERS	213	1	1.22
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:35	USERS	213	1	1.22
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:40	USERS	213	1	1.22
shr1.ind.hp.com	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/5/2012 9:45	USERS	213	1	1.22

Данные CSV явно показывают, что метрики записаны в виде пар ключей и значений в отдельных столбцах. В CSV-файле столбец "METRICID" со значением "212" соответствует использованию пространства экземпляром, а столбец "VALUEID" со значением "1" и "2" соответствует общему и свободному пространству. Эти два столбца в разных строках должны быть объединены для каждого уникального сочетания столбцов идентификаторов "HOSTNAME", "INSTANCE" и "AGENTTIMESTAMP" перед загрузкой в хранилище данных SHR.

Политика преобразования для поворотного преобразования данных CSV, описанных выше, приводится далее.

```
<?xml version="1.0" encoding="UTF-8"?>
<etldefinition type="TRANSFORM" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="SHRTransformNamespace schema_transform.xsd"
xmlns="SHRTransformNamespace">
<!-- specify a condition on METRICID column at recordSet level to pick only
instance space utilization metrics -->
<recordSet name="InstanceSpaceutilization"
condition="METRICID IN (212.0)"
source_type="InstanceSpaceutilization"
source_category="DBSPI_ORA_REPORT"
target_type="InstanceSpaceUtilization"
target_category="DBSPI_ORA_REPORT" doPivot="true">
<record name="HostName" source="HOSTNAME" id="true" />
<record name="InstanceName" source="INSTANCE" id="true" />
<record name="AGENTTIMESTAMP" source="AGENTTIMESTAMP" id="true" />
<!-- specify a condition on VALUEID column to pick the value of
allocated instance space -->
```

```

<record name="InstanceSpaceAllocatedSize" source="VALUE"
 condition="VALUEID=1.0" />
<!-- specify a condition on VALUEID column to pick the value of
 free instance space -->
<record name="InstanceSpaceFreeSize" source="VALUE"
 condition="VALUEID=2.0" />
</recordSet>
</etldefinition>

```

Выходной CSV-файл будет выглядеть следующим образом

DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_723914438884975.csv				
HOSTNAME	INSTANCENAME	AGENTTIMESTAMP	InstanceSpaceAllocatedSize	InstanceSpaceFreeSize
shr1.ind.hp.com	BAT92	4/5/2012 9:30	1890	225.48
shr1.ind.hp.com	BAT92	4/5/2012 9:35	1890	225.56
shr1.ind.hp.com	BAT92	4/5/2012 9:40	1890	225.56
shr1.ind.hp.com	BAT92	4/5/2012 9:45	1890	225.56

## Использование условий для фильтрации данных

XML политики преобразования поддерживает различные условия фильтрации, которые могут использоваться для фильтрации строк во входном/исходном CSV-файле. Условия можно указать в элементе `recordSet` (для фильтрации строк) и элементе `record` (для фильтрации столбцов). Ниже представлено несколько примеров указания условий

```

<!-- The condition below is defined to select the rows whose CLASSNAME column
value is "MEMORY" and HOSTNAME column value is not "NULL" -->

```

```

<recordSet name="Memory"
 condition="CLASSNAME IN (MEMORY) AND HOSTNAME NOT IN (NULL)"
 source_type="MEMORY"
 source_category="SiS"
 target_type="MEMORY"
 target_category="SiS" >

```

```

<record name="AGENTTIMESTAMP" source="AGENTTIMESTAMP" id="true"/>

```

```

<record name="HOSTNAME" source="HOSTNAME" id="true"/>

```

```

<record name="MEMORY_MB_FREE" source="MEMORY_MB_FREE"

```

```

 <!-- Filter out negative values-->

```



```

 condition="MEMORY_MB_FREE != -1" />
<record name="MEMORY_MB_TOTAL" source="MEMORY_MB_TOTAL"
 condition="MEMORY_MB_TOTAL != -1" />
</recordSet>

```

Возьмем следующий исходный CSV

<b>SiS_0_MEMORY_0_723914438884967.csv</b>				
<b>HOSTNAME</b>	<b>CLASSNAME</b>	<b>AGENTTIMESTAMP</b>	<b>MEMORY_MB_FREE</b>	<b>MEMORY_MB_TOTAL</b>
shr1.ind.hp.com	MEMORY	4/5/2012 9:30	4096	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:35	4096	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:40	4096	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:45	3584	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:50	3584	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:55	3584	8192
Null	Null	4/5/2012 10:00	3584	8192
shr1.ind.hp.com	MEMORY	4/5/2012 10:00	-1	-1
shr1.ind.hp.com	MEMORY	4/5/2012 10:00	3584	8192

Применение политики преобразования к этому исходному CSV-файлу приведет к созданию следующего выходного CSV-файла

<b>SiS_0_MEMORY_0_723914438884975.csv</b>				
<b>HOSTNAME</b>	<b>CLASSNAME</b>	<b>AGENTTIMESTAMP</b>	<b>MEMORY_MB_FREE</b>	<b>MEMORY_MB_TOTAL</b>
shr1.ind.hp.com	MEMORY	4/5/2012 9:30	4096	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:35	4096	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:40	4096	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:45	3584	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:50	3584	8192
shr1.ind.hp.com	MEMORY	4/5/2012 9:55	3584	8192
shr1.ind.hp.com	MEMORY	4/5/2012 10:05	3584	8192

## Использование функций

Политика преобразования поддерживает набор стандартных функций, которые можно использовать в исходном столбце CSV. Функции также можно использовать в условиях. В этом разделе описываются наиболее распространенные функции. См. полный список поддерживаемых функций в [приложении A](#).

### OPR\_TOLOWER, OPR\_TOUPPER

Функции OPR\_TOLOWER и OPR\_TOUPPER преобразовывают строковое значение в указанном столбце исходного CSV-файла в верхний или нижний регистр соответственно

Использование:

OPR\_TOLOWER(source\_column)

OPR\_TOUPPER(source\_column)

```
<record name="InstanceName" source="OPR_TOLOWER(INSTANCE)"/>
```

```
<record name="SegmentName" source=" OPR_TOUPPER(SEGMENT) "
 condition="OPR_TOLOWER(HOSTNAME)=shr1.ind.hp.com" />
```

### OPR\_APPEND

Эта функция дополняет значения двух указанным столбцов с разделителем.

Использование:

OPR\_APPEND(source\_column1, source\_column2, delimiter)

```
<record name="Name" source="OPR_APPEND(HOSTNAME, INSTANCE, :)" />
```

HOSTNAME,	INSTANCE	→	Name
shr1.ind.hp.com,	BAT92		shr1.ind.hp.com:BAT92

### OPR\_STRINGSPLIT

Эта функция использует строковое выражение, указанное вторым, для разделения значения имени указанного столбца и возвращает значение из массива строк после разделения с указанным индексом. Обратите внимание, что индекс начинается с 0.

Использование:

OPR\_STRINGSPLIT(<имя столбца>, <строковое выражение, которое будет использоваться для разделения>, <индекс разделенных данных для возврата>)

```
<record name="DisplayName"
 source="OPR_STRINGSPLIT(HOSTNAME, ., 0)"/>
```

HOSTNAME,		→	DisplayName
shr1.ind.hp.com			shr1

# Глава 12: Определение XML правила сверки

XML-файл политики сверки определяет набор правил сверки (процесса добавления CIUID к собранным данным фактов) собранных данных фактов из таких источников данных, как РА, с реестром типов ЭК, созданным во время сбора данных топологии RTSM. Каждое правило определяет набор условий, включающих столбцы из CSV-файла топологии/измерений (sourcecolumn — используется для построения реестра) и CSV-файла фактов (targetcolumn — используется для построения ключей для поиска в реестре).

## Структура XML

```
< etldefinition >
< rule >
 < condition >
 < sourcecolumn />
 < operator ></ operator >
 < targetcolumn />
 </ condition >
 < relation ></ relation >
</ rule >
</ etldefinition >
```

## Элемент XML: etldefinition

etldefinition — это корневой элемент определения политики преобразования.

Атрибут	Обязательный	Описание
type	Да	Определяет шаг ETL. Должен иметь значение "RECONCILE" для политики сверки
contentpack_name	Нет	Имя пакета содержимого
description	Нет	Краткое описание политики преобразования

Дочерние элементы	Множество	Описание
rule	1..n	

## Элемент XML: rule

Атрибут	Обязательный	Описание
name	Да	Значимое для пользователя имя правила
paClass	Да	Класс PA, к которому следует применить правило сверки. Значение также может быть псевдонимом класса PA. Это значение используется для выбора подходящего csv-файла для применения правила.  Например, если установлено значение “CPU”, правило будет применено к CSV-файлу со следующим шаблоном именованя  *_0_CPU_0_*.csv
ciType	Да	Тип ЭК, по которому выполняется поиск в реестре
idColumnName	Нет	Столбец идентификаторов, который будет использоваться для построения реестра и набора данных измерений. По умолчанию это столбец CIID.
targetColumnName	Нет	Имя столбца идентификаторов в выходном CSV-файле для данных фактов.  Значение по умолчанию: CIID
defaultValue	Нет	Значение, которое должно использоваться в качестве значения столбца идентификаторов по умолчанию, если столбец идентификаторов имеет значение null. Этот атрибут также является заместительным входным параметром, замещается значение указанного столбца.  Пример:  defaultValue="\${<SOURCE_COLUMN>}"
deploymentScenario	Нет	Сценарий, в котором следует применить правила сверки. Допустимые значения <ul style="list-style-type: none"> <li>• RTSM</li> <li>• OM</li> </ul> Если этот атрибут не указан, правило будет применяться ко всем сценариям.
category	Нет	Категория, для которой применяется сверка
ignorecase	Нет	Булево значение, которое указывает, следует ли игнорировать регистр во время сверки. По умолчанию установлено значение true.

Атрибут	Обязательный	Описание
description	Нет	Краткое описание правила сверки

Дочерние элементы	Множество	Описание
<a href="#">condition</a>	1..n	Список условий
<a href="#">relation</a>	0..n	Логическая связь между условиями

Элемент XML: **condition**

Атрибут	Обязательный	Описание
description	Нет	Описание условия

Дочерние элементы	Множество	Описание
<a href="#">sourcecolumn</a>	1..1	Исходный столбец используется для построения реестра
<a href="#">operator</a>	1..1	Логический оператор
<a href="#">targetcolumn</a>	1..1	Столбец CSV-файла фактов, который будет использоваться для сверки по реестру

Элемент XML: **sourcecolumn**

Элемент sourceColumn определяет имя столбца CSV-файла измерений, который будет использоваться для построения реестра.

Атрибут	Обязательный	Описание
name	Да	Имя столбца CSV в CSV-файле измерений
prefix	Нет	<p>Строка префикса, которую необходимо добавить к значению исходного столбца для создания ключа реестра/бизнес-ключа. В качестве префикса может использоваться строка или замещающий параметр для исходного столбца, который заменяет значение указанного столбца как префикс при построении реестра. Замещающий параметр будет иметь форму <code>prefix="{&lt;SOURCE_COLUMN&gt;}"</code></p> <p>Пример:</p> <pre>Instance    &lt;sourcecolumn name="Instance" suffix="ORA_"/&gt;            → Instance1   ORA_Instance1</pre>
suffix	Нет	Строка суффикса, которую необходимо добавить к

Атрибут	Обязательный	Описание
		<p>значению исходного столбца для создания ключа реестра/бизнес-ключа. В качестве суффикса может использоваться строка или замещающий параметр для исходного столбца, который заменяет значение указанного столбца как суффикс при построении реестра. Замещающий параметр будет иметь форму <code>suffix="{&lt;SOURCE_COLUMN&gt;}"</code></p> <p>Host        &lt;sourcecolumn name="Host" suffix=".ind.hp.com"/&gt;                        →          shr1        shr1.ind.hp.com</p>
function	Нет	<p>Функция, которая будет применена к исходному столбцу.</p> <p>Примечание: В настоящее время поддерживается только функция <code>substring</code>. Эта функция возвращает подстроку столбца указанным индексом при разделении по указанному шаблону.</p> <p>Пример  <code>"Oracle_Instance1" → function="SUBSTRING("_",1)" → "Instance1"</code></p>
description	Нет	Описание исходного столбца

**Элемент XML: operator**

Элемент `operator` определяет оператор сравнения между исходным столбцом и целевым столбцом. Ниже представлены поддерживаемые операторы

- EQUALS — равенство исходного и целевого столбцов
- LIKE — соответствие значения целевого столбца шаблону в исходном столбце

**Элемент XML: targetcolumn**

Элемент `targetColumn` определяет имя столбца из столбца CSV-файла фактов, который будет использоваться для сверки данных фактов с реестром.

Атрибут	Обязательный	Описание
name		Имя столбца CSV в CSV-файле фактов
prefix		<p>Строка префикса, которую необходимо добавить к значению целевого столбца для создания ключа поиска. В качестве префикса может использоваться строка или замещающий параметр для целевого столбца, который заменяет значение указанного столбца как префикс при построении реестра. Замещающий параметр будет иметь форму <code>prefix="{&lt;TARGET_COLUMN&gt;}"</code></p>

Атрибут	Обязательный	Описание
		<p>Пример:</p> <pre>Instance      &lt;targetcolumn name="Instance" suffix="ORA_" /&gt;</pre> <p style="text-align: center;">→</p> <pre>Instance1      ORA_Instance1</pre>
suffix		<p>Строка суффикса, которую необходимо добавить к значению целевого столбца для создания ключа поиска. В качестве суффикса может использоваться строка или замещающий параметр для целевого столбца, который заменяет значение указанного столбца как суффикс при построении реестра. Замещающий параметр будет иметь форму <code>suffix="{&lt;TARGETE_COLUMN&gt;}"</code></p> <pre>Host      &lt;targetcolumn name="host" suffix=".ind.hp.com" /&gt;</pre> <p style="text-align: center;">→</p> <pre>shr1      shr1.ind.hp.com</pre>
function		<p>Функция, которая будет применена к целевому столбцу.</p> <p>Примечание: В настоящее время поддерживается только функция <code>substring</code>. Эта функция возвращает подстроку столбца указанным индексом при разделении по указанному шаблону.</p> <p>Пример</p> <pre>"Oracle_Instance1" → function="SUBSTRING("_",1)" → "Instance1"</pre>
description		Описание целевого столбца

#### Элемент XML: **relation**

Элемент `relation` определяет логическую связь между двумя условиями. В настоящее время поддерживаются только операторы `"AND"` и `"OR"`.

## Примеры правила сверки

Рассмотрим развертывание СУБД Oracle в сценарии RTSM. Информация о топологии и типах ЭК для СУБД Oracles собрана в RTSM, и метрики производительности СУБД Oracle собраны в интеллектуальных подключаемых модулях HP для БД Oracle. Эти две группы

данных связаны посредством процесса сверки. Сверка данных — это двухэтапный процесс

1. Построение реестра сверки ЭК с помощью данных, собранных в RTSM
2. Добавление UID ЭК в данный фактов, собранных агентом HP Operation Agent, путем поиска в реестре сверки.

#### Построение реестра сверки ЭК с помощью данных, собранных в RTSM

Данные, собранные из RTSM			
ORA_Deployment_0_oracle_0_722990140479794.csv			
CiType	Cild	database_dbsid	host_dnsname
oracle	5935697711c6c268ca53888e5d30ecc7	BAT92	SHR1.IND.HP.COM
oracle	211408823a8cc17eca52174549f82f79	ORCL11G	SHR2.IND.HP.COM
oracle	1472234cfc2ee5bc2b3f2f79249ebc0f	VM8	SHR3.IND.HP.COM



```
<rule name="Reconciliation rule for InstanceSpaceutilization" ciType="oracle" paClass="InstanceSpaceutilization">
 <condition>
 <sourcecolumn name="database_dbsid"/>
 <operator>EQUALS</operator>
 <targetcolumn name="InstanceName"/>
 </condition>
 <relation>AND</relation>
 <condition>
 <sourcecolumn name="host_dnsname"/>
 <operator>EQUALS</operator>
 <targetcolumn name="HostName"/>
 </condition>
</rule>
```



Построение реестра сверки на основе правила сверки, заданного для исходных столбцов, собранных в RTSM



Реестр сверки для типа ЭК "oracle"	
business_key ciid	business_key ciid
_BAT92_SHR1.IND.HP.COM	5935697711c6c268ca53888e5d30ecc7
_ORCL11G_SHR2.IND.HP.COM	211408823a8cc17eca52174549f82f79
_VM8_SHR3.IND.HP.COM	1472234cfc2ee5bc2b3f2f79249ebc0f

Добавление UID ЭК в данным фактов, собранных агентом HP Operation Agent, путем поиска в реестре сверки.

Данные фактов (использование места в экземпляре Oracle), собранные из HP Operations Agent						
DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_723914434447063.csv						
HOSTNAME	DATASOURCE	CLASSNAME	AGENTTIMESTAMP	METRICID	INSTANCENAME	VALUEID
SHR1.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:25	215	BAT92	1
SHR1.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:30	215	BAT92	1
SHR1.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:35	215	BAT92	1
SHR2.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:25	173,1	vm8	1
SHR2.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:30	215	vm8	1
SHR2.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:35	209	vm8	1



Build reconcile registry lookup key based on the reconciliation rule defined for target columns collected from HP Operations Agent

```
<rule name="Reconciliation rule for InstanceSpaceutilization" ciType="oracle" paClass="InstanceSpaceutilization"
 <condition>
 <sourcecolumn name="database_dbsid"/>
 <operator>EQUALS</operator>
 <targetcolumn name="InstanceName"/>
 </condition>
 <relation>AND</relation>
 <condition>
 <sourcecolumn name="host_dnsname"/>
 <operator>EQUALS</operator>
 <targetcolumn name="HostName"/>
 </condition>
</rule>
```



Lookup reconcile registry with lookup key to get the corresponding CIID  
 e.g. InstanceName = BAT92, Hostname=SHR1.ind.hp.com  
 Lookup key = \_BAT92\_SHR1.IND.HP.COM

Реестр сверки для типа ЭК "oracle"	
business_key	ciid
_BAT92_SHR1.IND.HP.COM	5935697711c6c268ca53888e5d30ecc7
_ORCL11G_SHR2.IND.HP.COM	211408823a8cc17eca52174549f82f79
_VM8_SHR3.IND.HP.COM	1472234cfc2ee5bc2b3f2f79249ebc0f



Add the CIID column to the collected fact data

Сверенные данные фактов (использование места в экземпляре Oracle)							
DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_723914434447065.csv							
HOSTNAME	DATASOURCE	CLASS NAME	AGENTTIME STAMP	METRICID	INSTANCE NAME	VALUEID	CIID
SHR1.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:25	215	BAT92	1	5935697711c6c268ca53888e5d30ecc7
SHR1.IND.HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:30	215	BAT92	1	5935697711c6c268ca53888e5d30ecc8

Сверенные данные фактов (использование места в экземпляре Oracle)							
DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_723914434447065.csv							
HOSTNAME	DATASOURCE	CLASS NAME	AGENTTIME STAMP	METRICID	INSTANCE NAME	VALUEID	CIID
SHR1.IND. HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:35	215	BAT92	1	5935697711c6c268ca53888e5d30ecc9
SHR2.IND. HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:25	173,1	vm8	1	211408823a8cc17eca52174549f82f79
SHR2.IND. HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:30	215	vm8	1	211408823a8cc17eca52174549f82f80
SHR2.IND. HP.COM	DBSPI_ORA_REPORT	DBSPI_ORA_REPORT	4/18/2012 5:35	209	vm8	1	211408823a8cc17eca52174549f82f81

# Глава 13: Определение XML правила промежуточного хранения

XML-файл правила промежуточного хранения определяет правила и сопоставления столбцов для загрузки данных из CSV-файла в соответствующую таблицу промежуточного хранения. XML-файл правила промежуточного хранения обычно разрабатывается с использованием документа интерфейса промежуточного хранения для базовой области промежуточного хранения (физических таблиц промежуточного хранения).

## Структура XML

```
< stageRule >
< targets >
< target >
< sources >
 < source />
</ sources >
< columnMap >
 < mapping />
</ columnMap >
< relational >
 < relation />
</ relational >
< keyColumns >
< key />
</ keyColumns >
</ target >
</ target >
</ stageRule >
```

## Элемент XML: stagerule

stagerule — это корневой элемент определения правила промежуточного хранения.

Атрибут	Обязательный	Описание
name	Да	Имя правила промежуточного хранения

Дочерние элементы	Множество	Описание
<a href="#">targets</a>	1..1	Группирование целевых объектов области промежуточного хранения

## Элемент XML: targets

Элемент targets группирует элемент target

Дочерние элементы	Множество	Описание
<a href="#">target</a>	1..n	Список целевых областей промежуточного хранения

Элемент XML: **target**

Элемент target связывает шаблон CSV-файла с таблицей промежуточного хранения и задает сопоставление между CSV-файлом и столбцами таблицы промежуточного хранения.

Атрибут	Обязательный	Описание
name	Да	Имя таблицы промежуточного хранения в документе интерфейса промежуточного хранения

Дочерние элементы	Множество	Описание
<a href="#">sources</a>	1..1	Группа шаблонов CSV-файлов источника
<a href="#">columnMap</a>	1..1	Группа сопоставлений CSV-файлов и столбцов таблицы промежуточного хранения
<a href="#">relational</a>	0..1	Условие Join для объединения столбцов из двух исходных CSV-файлов
<a href="#">keyColumns</a>	0..1	Ключевые столбцы CSV-файла (для удаления дублированных строк)

Элемент XML: **sources**

Дочерние элементы	Множество	Описание
<a href="#">source</a>	1..n	Список шаблонов CSV-файлов источника

Элемент XML: **source**

Определяет шаблоны CSV-файлов источника, данные из которых перемещаются в таблицу промежуточного хранения.

Атрибут	Обязательный	Описание
категория	Да	Исходная категория. Если имя исходного CSV-файла — Mapper_DISK_0_Reconcile_DISK_0_SCOPE_0_DISK_0_2506038836739754.csv, то категория будет SCOPE.  <b>Примечание.</b> Обычно категория в имени CSV-файла будет именем источника данных PA или представления CMDB
type	Да	Тип источника. Если имя исходного CSV-файла — Mapper_DISK_0_Reconcile_DISK_0_SCOPE_0_DISK_0_2506038836739754.csv, то тип будет DISK.  <b>Примечание.</b> Обычно тип в имени CSV-файла будет именем класса PA или типа ЭК CMDB
alias	Да	Пользовательский псевдоним сочетания типа и категории
schemaName	Да	Имя схемы документа интерфейса промежуточного хранения, включающего определение промежуточного хранения целевого объекта
retentionDays	Нет	Период хранения данных в таблице промежуточного хранения в днях. Значение по умолчанию: 3

Элемент XML: **columnMap**

Дочерние элементы	Множество	Описание
<a href="#">mapping</a>	1..n	Список сопоставлений CSV-файлов и столбцов таблицы промежуточного хранения

Элемент XML: **mapping**

Элемент mapping определяет сопоставление между столбцом CSV-файла и столбцом таблицы промежуточного хранения

Атрибут	Обязательный	Описание
srcColumn	Да	Имя исходного столбца CSV-файла, которому предшествует псевдоним источника  Пример: source_alias.csvColumnName
tgtColumn	Да	Имя столбца таблицы, заданное в документе интерфейса промежуточного хранения

Элемент XML: **relational**

Дочерние элементы	Множество	Описание
<a href="#">relation</a>	1..n	Список сопоставлений CSV-файлов и столбцов таблицы промежуточного хранения

Элемент XML: **relation**

Сопоставление столбца CSV-файла и столбца таблицы промежуточного хранения

Атрибут	Обязательный	Описание
key	Да	Условие Join для объединения двух CSV-файлов

Элемент XML: **keyColumns**

Дочерние элементы	Множество	Описание
<a href="#">key</a>	1..n	Список ключевых столбцов в CSV-файле

Элемент XML: **key**

Определяет ключевые столбцы в CSV-файле

Атрибут	Обязательный	Описание
alias	Да	Псевдоним исходного CSV-файла
srcColumn	Да	Имя столбца исходного CSV-файла
sortKey	Нет	Столбец исходного CSV-файла, который следует учитывать при удалении дублированного значения. Значение по умолчанию: столбец "INSERTTIME", и дубликаты будут удалены путем сохранения строки с самым поздним временем вставки и удаления остальных строк.

## Примеры правила промежуточного хранения

### Пример XML-файла правила промежуточного хранения

См. пример правила промежуточного хранения `Stagerule_CPU_SCOPE.xml` в разделе вложений. Он обеспечивает загрузку данных CSV в таблицу промежуточного хранения

## Пример объединения столбцов CSV

В некоторых случаях исходный CSV-файл (данные/метрики из класса PA) может не иметь столбцов, необходимых для таблицы промежуточного хранения. Однако отсутствующие столбцы могут быть доступны в других исходных CSV-файлах (данных/метриках из класса PA). В таких случаях столбцы из CSV-файлов должны быть объединены в соответствии с условием, а затем загружены в таблицу промежуточного хранения. См. пример правила промежуточного хранения `Stagerule_column_merge_example.xml` в разделе вложений. Оно определяет объединение столбцов между CSV-файлами.



## Глава 14: Определение XML потока ABC

ABC (Audit, Balance, Control) — это платформа, обеспечивающая моделирование и запуск рабочих процессов. Она позволяет устанавливать отношения родительский-дочерний между выполняемыми задачами и группировать наборы связанных задач в потоки. Каждая задача в потоке называется шагом. В настоящее время SHR использует только функцию контроля ABC. XML-файл определения потока обычно содержит набор связанных определений шагов и их отношения.

### Структура XML

```
< JobStream >
 < JobStreamMetaInfo >
 < JobStreamMetaData />
 </ JobStreamMetaInfo >
 < JobStreamSteps >
 < JobStreamStep >
 < JobStreamStepMetaInfo >
 < JobStreamStepMetaData />
 </ JobStreamStepMetaInfo >
 </ JobStreamStep >
 </ JobStreamSteps >
 < JobStreamLinks >
 < JobStreamLink />
 </ JobStreamLinks >
</ JobStream >
```

### Элемент XML: JobStream

Элемент JobStream — это корневой элемент файла потока.

Атрибут	Обязательный	Описание
dwid	Да	Уникальный идентификатор потока. Значение должно начинаться с имени пакета содержимого, за которым следуют символ '@' и уникальное имя потока
businessname	Да	Метка потока

Атрибут	Обязательный	Описание
scheduleloadstarttime	Нет	<p>Определяет время начала выполнения потока ABC. Это время начала следует использовать только при первом выполнении потока. После выполнения потока данные о выполнении будут доступны в таблицах времени выполнения, и этот атрибут не будет учитываться.</p> <p>Примечание: Значение этого атрибута имеет шаг 5 и должно задаваться только в 24 часовом формате, часы и минуты разделяются двоеточием (:). Например, 22:05 означает, что поток будет выполнен в 22:05. Если этот атрибут не указан, поток будет загружен при первом выполнении операции abcLoadNRun, которая будет запущена после импорта потока.</p>
scheduleloadfrequency	Нет	<p>Указывает частоту (в минутах) повторной загрузки потока. Например, если установлена частота 30 минут, поток будет загружаться каждые 30 минут. Значение по умолчанию: 5 минут</p>

Дочерние элементы	Множество	Описание
<a href="#">JobStreamMetaInfo</a>	0..1	Метаданные потока
<a href="#">JobStreamSteps</a>	1..1	Группа шагов потока
<a href="#">JobStreamLinks</a>	0..1	Группа отношений потока

Элемент XML: [JobStreamMetaInfo](#)

Дочерние элементы	Множество	Описание
<a href="#">JobStreamMetaData</a>	1..n	Метаданные потока в виде пар имя-значение

Элемент XML: **JobStreamMetaData**

Атрибут	Обязательный	Описание
name	Да	Имя метаданных
value	Да	Значение метаданных

## Элемент XML: JobStreamSteps

Дочерние элементы	Множество	Описание
<a href="#">JobStreamStep</a>	1..n	Список шагов

Элемент XML: **JobStreamStep**

Элемент JobStreamStep представляет задачу в потоку.

Атрибут	Обязательный	Описание
dwid	Да	Уникальный идентификатор шага
businessname	Да	Метка шага
catalog	Да	Каталог ABC, из которого направляются исполняемые файлы. В качестве значения всегда устанавливается каталог "platform", который включает набор мнемоник.
executableidentifier	Да	<p>Значение представляет мнемонику, которая сопоставляется с базовым двоичным файлом платформы. Возможные значение</p> <p><b>COLLECT</b> — консолидирует и перемещает собранные данные из папки %PMDB_HOME\collect в %PMDB_HOME\collect. В качестве аргумента используются тип и категория. Тип и категория, указанные как атрибут, должны быть связаны с политикой/правилом в XML политики сбора данных, чтобы CSV-файл сбора данных мог быть обработан этим шагом.</p> <p>Синтаксис: type:category</p> <p><b>TRANSFORM</b> — выполняет преобразование собранных данных на основе заданной политики преобразования. В качестве аргумента используются тип и категория. Сочетание элементов type и category, указанное как аргумент, должно соответствовать аналогичному сочетанию в имени выходного CSV-файла, созданного родительским шагом (COLLECT/ RECONCILE), чтобы выходной CSV-файл выбирался этим шагом TRANSFORM для преобразования.</p> <p>Синтаксис: type:category</p> <p><b>RECONCILE</b> — выполняет сверку данных в соответствии с заданным правилом сверки. Использует тип dim и категорию dim в качестве аргумента. Сочетание элементов type и category, указанное как аргумент, должно соответствовать аналогичному сочетанию в имени выходного CSV-файла, созданного родительским шагом (COLLECT/ TRANSFORM), чтобы выходной CSV-файл выбирался этим шагом RECONCILE для сверки.</p> <p>Синтаксис: type:category</p>

Атрибут	Обязательный	Описание
		<p>STAGE — загрузка данных из csv-файла в таблицы промежуточного хранения в соответствии с правилом промежуточного хранения. В качестве аргумента используется имя файла правила промежуточного хранения</p> <p>LOAD — загружает данные из таблицы промежуточного хранения в таблицу хранилища данных. Использует имя таблицы хранилища данных в качестве аргумента</p> <p>AGGREGATE — выполняет сведение данных в соответствии с конфигурацией, заданной в XML-модели, и вставляет сводные данные в сводную таблицу. Использует имя сводной таблицы с именем схемы в качестве префикса как аргумент (например, schema_name:aggregate_table_name)</p> <p>EXEC_PROC — выполняет SQL-процедуру. Использует имя процедуры в качестве аргумента</p>
arguments	Нет	Аргумент для шага с указанным executableidentifier.
maxexectime	Да	Максимальное время исполнения в минутах
maxretries	Да	Максимальное количество повторов при отказе.

Дочерние элементы	Множество	Описание
<a href="#">JobStreamStepMetaInfo</a>	0..1	Метаданные для шага

Элемент XML: **JobStreamStepMetaInfo**

Дочерние элементы	Множество	Описание
<a href="#">JobStreamStepMetaData</a>	1..n	Метаданные потока в виде пар имя-значение

Элемент XML: **JobStreamStepMetaData**

**Примечание.** JobStreamStepMetaData — будет автоматически добавлен CPC для всех шагов, кроме шагов с “EXEC\_PROC” в качестве [executableidentifier](#)

Атрибут	Обязательный	Описание
name	Да	Имя метаданных
value	Да	Значение метаданных

## Элемент XML: JobStreamLinks

Дочерние элементы	Множество	Описание
<a href="#">JobStreamLink</a>	1..n	Определение отношения между шагами

Элемент XML: **JobStreamLink**

Элемент JobStreamLink определяет отношение родительский-дочерний между шагами.

**Примечание.** Шаги без атрибута parentstepidentifier будут выполняться параллельно

Атрибут	Обязательный	Описание
stepidentifier	Да	dwid шага
parentstepidentifier	Нет	dwid родительского шага

## Примеры определения потока ABC

Потоки ABC в SHR можно разделить на два типа

### Поток ETL («Извлечение, преобразование и загрузка»)

Поток ETL содержит шаги, связанные с процессом ETL (сбор, преобразование, сверка и промежуточное хранение). См. пример файла потока ETL\_CPU\_Stream.xml в разделе вложений, который определяет шаги ETL для данных производительности ЦП из Performance Agent

### Поток хранилища данных

Поток хранилища данных содержит шаги по перемещению данных из области промежуточного хранения в таблицы хранилища данных и их дальнейшему сведению. См. пример файла потока Datawarehouse\_CPU\_Stream.xml разделе вложений. Он определяет шаги по перемещению данных о производительности ЦП из области промежуточного хранения в таблицы хранилища данных и их последующему сведению.

# Глава 15: Атрибуты type и category в политиках ETL

Все политики SHR ETL определяют два атрибута: type и category для привязки правила/политики к CSV-файлу, содержащему эти два атрибута в имени.

## Атрибуты type и category в политиках сбора данных

Политики сбора SHR определяют два уникальных атрибута для каждого полученного CSV-файла. Эти два атрибута используются в имени CSV-файла и используются для идентификации файла для дальнейшей обработки: преобразования, сверки и промежуточного хранения. Правила/политика преобразования, сверки и промежуточного хранения должны определять подходящие атрибуты type и category для применения правил к нужным CSV-файлам в соответствии с этими атрибутами. П полученному CSV-файлу будет применяться следующий шаблон именованя.

```
category_0_type_0_.csv
```

**Примечание.** Если атрибуты type и category не заданы в политике сбора данных, будут использоваться значения по умолчанию в соответствии с типом политики сбора данных.

## Атрибуты type и category в политике сбора данных RTSM

В политике сбора данных RTSM элементы type и category задаются для каждого определения типа ЭК, и для каждого определения типа ЭК создается CSV-файл. Если атрибуты type и category не заданы для определения типа ЭК, используются значения по умолчанию. Значение элемента type по умолчанию — имя типа ЭК, а значение элемента category по умолчанию — имя представления. Рассмотрим тип ЭК oracle в представлении ORA\_Deployment

```
<view name="ORA_Deployment">
 <citype name="oracle">

 </citype>
</view>
```

В примере выше атрибуты type и category не заданы, значит будет использоваться значение type по умолчанию "oracle" и значение category по умолчанию "ORA\_Deployment". Полученный CSV-файл будет иметь имя

```
ORA_Deployment_0_oracle_0_.csv
```

Определение типа ЭК может включать несколько пользовательских сочетаний `type` и `category` (псевдонимов), см. ниже

```
<view name="ORA_Deployment">
 <citype name="oracle">
 <aliassource>
 <aliastarget type="oracle" category="ORA_Deployment"/>
 <aliastarget type="database" category="ORA_Deployment"/>
 </aliassource>

 </citype>
</view>
```

В примере выше задано два сочетания элементов `type` и `category`, значит будет создано два CSV-файла с именами:

```
ORA_Deployment_0_oracle_0_.csv
```

```
ORA_Deployment_0_database_0_.csv
```

## Атрибуты `type` и `category` в политике сбора данных OM

В политике сбора данных OM элементы `type` и `category` задаются для каждого правила сбора данных OM, включающего класс PA и источник данных, и для каждого определения правила создается CSV-файл. Если атрибуты `type` и `category` не заданы для определения правила, используются значения по умолчанию. Значение элемента `type` по умолчанию — значение атрибута `citype`, а значение элемента `category` по умолчанию — имя коллекции OM. Рассмотрим правило сбора данных OM для базы данных Oracle

```
<sncollection name="ORA_Deployment" mappedby="DBSPI_ORA_GRAPH">
 <rule citype="oracle" class="DBSPI_ORA_GRAPH" datasource="DBSPI_ORA_GRAPH">

 </rule>
</sncollection>
```

В примере выше атрибуты `type` и `category` не заданы, значит будет использоваться значение `type` по умолчанию "oracle" и значение `category` по умолчанию "ORA\_Deployment". Полученный CSV-файл будет иметь имя

```
ORA_Deployment_0_oracle_0_.csv
```

Определение правила может включать несколько пользовательских сочетаний `type` и `category` (псевдонимов), см. ниже

```

<snccollection name="ORA_Deployment" mappedby="DBSPI_ORA_GRAPH">
 <rule citype="oracle" class="DBSPI_ORA_GRAPH" datasource="DBSPI_ORA_GRAPH">
 <aliassource>
 <aliastarget type="oracle" category="ORA_Deployment"/>
 <aliastarget type="database" category="ORA_Deployment"/>
 </aliassource>

 </rule>
</snccollection>

```

В примере выше задано два сочетания элементов `type` и `category`, значит будет создано два CSV-файла с именами:

```
ORA_Deployment_0_oracle_0_.csv
```

```
ORA_Deployment_0_database_0_.csv
```

## Атрибуты `type` и `category` в политике сбора данных OA

В политике сбора данных OA элементы `type` и `category` задаются для каждого определения класса OA, и для каждого определения класса создается CSV-файл. Если атрибуты `type` и `category` не заданы для определения класса, используются значения по умолчанию. Значение элемента `type` по умолчанию — имя класса, а значение элемента `category` по умолчанию — имя источника данных. Рассмотрим источник данных Oracle SPI

```

<datasource name="DBSPI_ORA_REPORT">
 <class name="DBSPI_ORA_REPORT" summarized="true">

 </class>
</datasource>

```

В примере выше атрибуты `type` и `category` не заданы, значит будет использоваться значение `type` по умолчанию "DBSPI\_ORA\_REPORT" и значение `category` по умолчанию "DBSPI\_ORA\_REPORT". Полученный CSV-файл будет иметь имя

```
DBSPI_ORA_REPORT_0_DBSPI_ORA_REPORT_0_.csv
```

Определение класса может включать несколько пользовательских сочетаний `type` и `category` (псевдонимов), см. ниже

```

<datasource name="DBSPI_ORA_REPORT">
 <class name="DBSPI_ORA_REPORT" summarized="true">
 <aliassource>
 <aliastarget type="InstanceAvailability" category="DBSPI_ORA_REPORT"/>
 <aliastarget type="InstanceSpaceutilization" category="DBSPI_ORA_REPORT"/>
 </aliassource>

 </class>
</datasource>

```



В примере выше задано два сочетания элементов `type` и `category`, значит будет создано два CSV-файла с именами:

`*DBSPI_ORA_REPORT_0_InstanceAvailability_0_*.csv`

`*DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_*.csv`

## Атрибуты `type` и `category` в политике сбора данных БД

В политике сбора данных БД элементы `type` и `category` задаются для каждого определения запроса `join`, и для каждого определения запроса `join` создается CSV-файл. Рассмотрим источник данных Oracle SPI

```
<joinquery type="Interface_Fact" category="Network">
 <content>SELECT * FROM nps_f_hour_InterfaceMetrics</content>
</joinquery>
```

Для примера выше полученный CSV-файл будет иметь имя

`*Network_0_Interface_Fact_0_*.csv`

## Атрибуты `type` и `category` в политике преобразования

В политике преобразования используется два набора атрибутов `type` и `category`: элементы `source type`, `source category` и `target type`, `target category` входят в каждый набор записей. Элементы `source type` и `source category` используются для идентификации исходного/входного CSV-файла для применения правил преобразования, в то время как `target type` и `target category` будут использоваться в имени выходного CSV-файла.

Элементы `source type` и `source category` должны быть заданы в соответствии с типом и категорией выходного CSV-файла, созданного процессом ETL (сбор данных/сверка), который выполняется перед процессом преобразования.

Возьмем следующее правило преобразования

```
<recordSet name="InstanceSpaceutilization" condition="METRICID IN (212.0)"
 source_type="InstanceSpaceutilization" source_category="DBSPI_ORA_REPORT"
 target_type="InstanceSpaceUtilization" target_category="DBSPI_ORA_REPORT"
 doPivot="true">

</recordSet>
```

В этом примере `source type` имеет значение "InstanceSpaceutilization", а `source category` — значение "DBSPI\_ORA\_REPORT". Таким образом правило применяется к исходному CSV-файлу (созданному процессом сбора данных/сверки), имя которого включает соответствующие элементы `type` и `category`

`(*DBSPI_ORA_REPORT_0_InstanceSpaceutilization_0_*.csv)`

## Атрибуты type и category в правиле сверки

Политика сверки не определяет атрибут type, вместо него определяется атрибут "citype", который идентифицирует CSV-файл для построения реестра сверки, и атрибуты "paclass" и "category", которые идентифицируют CSV-файл фактов для сверки.

Атрибуты "paclass" и "category" должны быть заданы в соответствии с типом и категорией выходного CSV-файла, созданного процессом ETL (сбор данных/преобразование), который выполняется перед процессом сверки. Атрибут "citype" должен быть задан в соответствии с атрибутом citype в политике сбора данных RTSM. Кроме того, выходной CSV-файл будет иметь те же имена, что исходный CSV-файл

Возьмем следующее правило сверки

```
<rule name="Reconciliation rule for InstanceSpaceutilization"
 ciType="oracle" paClass="InstanceSpaceutilization" category="DBSPI_ORA_REPORT">

</rule>
```

Это правило применяется к CSV-файлу с шаблоном имени файла \*\_0\_oracle\_0\_\*.csv (созданным процессом сбора данных RTSM) для построения реестра и CSV-файлу со следующим шаблоном имени файла

*\*DBSPI\_ORA\_REPORT\_0\_InstanceSpaceutilization\_0\_\*.csv* (создается процессом сбора данных/преобразования) для сверки данных.

## Атрибуты type и category в правиле промежуточного хранения

В файле правила промежуточного хранения несколько наборов атрибутов type и category задаются для каждого целевого объекта идентифицируют исходные CSV-файлы, данные из которых загружаются в соответствующую таблицу промежуточного хранения.

Возьмем следующее правило промежуточного хранения

```
<stageRule xmlns="http://www.hp.com/SHR/Stage/v1.0" name="DB Oracle Instance Space Utilization"
 <targets>
 <target name="DB Oracle Instance Space Utilization">
 <sources>
 <source alias="ISU" type="InstanceSpaceUtilization" category="DBSPI_ORA_REPORT"
 schemaName="CoreDatabaseOracle"/>
 </sources>

 </target>
 </targets>
</stageRule>
```

Это правило применяется к CSV-файлу с шаблоном имени файла *\*DBSPI\_ORA\_REPORT\_0\_InstanceSpaceutilization\_0\_\*.csv* (создается процессом сбора/преобразования/сверки) для загрузки данных из CSV-файла в таблицу промежуточного хранения

## Рабочий процесс CSV-файла

### Политика сбора данных OA

```
<datasource name="DBSPI_ORA_REPORT">
 <class name="DBSPI_ORA_REPORT" summarized="true">

 </class>
</datasource>
```



*\*DBSPI\_ORA\_REPORT\_0\_InstanceSpaceutilization\_0\_\*.csv*

### Правило преобразования

```
<recordSet name="InstanceSpaceutilization" condition="METRICID IN (212.0)"
 source_type="InstanceSpaceutilization" source_category="DBSPI_ORA_REPORT"
 target_type="InstanceSpaceUtilization" target_category="DBSPI_ORA_REPORT"
 doPivot="true">

</recordSet>
```



*\*DBSPI\_ORA\_REPORT\_0\_InstanceSpaceutilization\_0\_\*.csv*

### Правило выверки

```
<rule name="Reconciliation rule for InstanceSpaceutilization"
 ciType="oracle" paClass="InstanceSpaceutilization" category="DBSPI_ORA_REPORT">

</rule>
```



*\*DBSPI\_ORA\_REPORT\_0\_InstanceSpaceutilization\_0\_\*.csv*

### Правило промежуточного хранения

```
<stageRule xmlns="http://www.hp.com/SHR/Stage/v1.0" name="DB Oracle Instance Space Utilization">
 <targets>
 <target name="DB Oracle Instance Space Utilization">
 <sources>
 <source alias="ISU" type="InstanceSpaceUtilization" category="DBSPI_ORA_REPORT"
 schemaName="CoreDatabaseOracle"/>
 </sources>

 </target>
 </targets>
</stageRule>
```

# Глава 16: Определение XML стратегии

XML стратегии определяет конфигурацию для создания различных артефактов из XML-модели. Он определяет следующее

- Стратегию создания интерфейса и схемы области промежуточного хранения из XML-модели. Стратегия может быть задана на уровне схемы, таблицы фактов или таблицы измерений, а также для определенной таблицы фактов или измерений. Стратегия, заданная на определенном уровне, переопределяет стратегию на более высоких уровнях.
- Список, для которых не следует создавать таблицы промежуточного хранения
- Конфигурация простоев для таблиц фактов (включение/выключения добавления сведений о простоев, измерение, которое будет учитываться при добавлении данных о простое)
- Список таблиц измерений, для которых не следует создавать таблицы простоев

**Примечание.** XML стратегии должен содержать конфигурацию для XML-модели. Если существует несколько XML-моделей, XML стратегии должен быть задан по отдельности для каждой XML-модели.

## Структура XML

```
< schema >
 < factTables >
 < table />
 </ factTables >
 < dimensionTables >
 < table />
 </ dimensionTables >
 < stageExclusions >
 < excludeTable />
 </ stageExclusions >
 < downtimeExclusions >
 < dimensionTable />
 </ downtimeExclusions >
</ schema >
```

## Элемент XML: schema

Элемент `schema` — это корневой элемент XML определения стратегии.

Атрибут	Обязательный	Описание
<code>name</code>	Да	Имя схемы. Имя схемы должно совпадать с именем схемы в XML соответствующей модели
<code>strategy</code>	Нет	<p>Глобальная стратегия создания схемы таблицы промежуточного хранения для данной модели. Допустимые значения</p> <ul style="list-style-type: none"> <li><code>normalize</code></li> </ul> <p>Созданный DDL таблицы промежуточного хранения создается для каждой таблицы фактов/измерений и будет включать столбец для каждого столбца в таблице фактов/измерений и столбец для каждого столбца бизнес-ключа из всех родительских таблиц измерений.</p> <ul style="list-style-type: none"> <li><code>Denormalize</code></li> </ul> <p>Эта стратегия не поддерживается в настоящее время</p> <p>Значение по умолчанию: <code>normalize</code></p>
<code>referenceLevel</code>	Нет	<p>Уровень ссылки таблицы должен учитываться при создании схемы для таблицы промежуточного хранения.</p> <p>Например, рассмотрим следующую иерархию ссылок таблицы</p> <p>Table1 → DimTable1 → DimTable2</p> <p>В примере выше DimTable1 является ссылкой уровня 1, а DimTable2 — ссылкой уровня 2 из таблицы Table1. Если для параметра <code>referenceLevel</code> выбрано значение “1” для таблицы Table1, то созданный DDL таблицы промежуточного хранения для таблицы “Table1” будет включать столбец для каждого столбца бизнес-ключа из таблицы “Table1” и столбец для каждого столбца бизнес-ключа из таблицы “DimTable1”, но не будет включать столбец для каждого столбца бизнес-ключа из таблицы “DimTable2” (таблица “DimTable2” будет пропущена).</p>

Дочерние элементы	Множество	Описание
<a href="#">factTables</a>	0..1	Стратегия для таблиц фактов
<a href="#">dimensionTables</a>	0..1	Стратегия для таблиц измерений

Дочерние элементы	Множество	Описание
<a href="#">stageExclusions</a>	0..1	Список таблиц фактов и измерений, для которых не следует создавать таблицу промежуточного хранения
<a href="#">downtimeExclusions</a>	0..1	Список таблиц измерений, для которых не следует создавать таблицу времени простоя

## Элемент XML: factTables

Атрибут	Обязательный	Описание
strategy	Да	<p>Глобальная стратегия создания схемы таблицы промежуточного хранения для таблиц фактов. Допустимые значения</p> <ul style="list-style-type: none"> <li>normalize Созданный DDL таблицы промежуточного хранения создается для каждой таблицы фактов и будет включать столбец для каждого столбца в таблице фактов и столбец для каждого столбца бизнес-ключа из всех родительских таблиц измерений.</li> <li>Denormalize Эта стратегия не поддерживается в настоящее время Значение по умолчанию: normalize</li> </ul>
referenceLevel	Нет	<p>Уровень ссылки таблицы должен учитываться при создании схемы для таблицы промежуточного хранения для всех таблиц фактов.</p> <p>Например, рассмотрим следующую иерархию ссылок таблицы</p> <p>FactTable1 → DimTable1 → DimTable2</p> <p>В примере выше DimTable1 является ссылкой уровня 1, а DimTable2 — ссылкой уровня 2 из таблицы FactTable1. Если для параметра referenceLevel выбрано значение "1" для таблицы FactTable1, то созданный DDL таблицы промежуточного хранения для таблицы фактов "FactTable1" будет включать столбец для каждого столбца бизнес-ключа из таблицы "FactTable1" и столбец для каждого столбца бизнес-ключа из таблицы "DimTable1", но не будет включать столбец для каждого столбца бизнес-ключа из таблицы "DimTable2" (таблица "DimTable2" будет пропущена).</p>
enrichDowntime	Нет	Булево значение, указывающее, включено ли

Атрибут	Обязательный	Описание
		добавление сведений о простоях для всех таблиц фактов. По умолчанию добавление сведений о простоях включено для всех таблиц фактов.

Дочерние элементы	Множество	Описание
<a href="#">table</a>	0..n	Список таблиц фактов и конфигурация стратегии.

Элемент XML: **table**

Атрибут	Обязательный	Описание
name	Да	Имя таблицы фактов. Имя должно совпадать с именем таблицы фактов, заданным в XML модели Примечание: Имя чувствительно к регистру
strategy	Нет	Стратегия создания схемы таблицы промежуточного хранения для таблицы фактов. Допустимые значения <ul style="list-style-type: none"> <li>normalize Созданный DDL таблицы промежуточного хранения создается для таблицы фактов и будет включать столбец для каждого столбца в таблице фактов и столбец для каждого столбца бизнес-ключа из всех родительских таблиц измерений.</li> <li>Denormalize Эта стратегия не поддерживается в настоящее время Значение по умолчанию: normalize</li> </ul>
referenceLevel	Нет	Уровень ссылки таблицы должен учитываться при создании схемы для таблицы промежуточного хранения для таблицы фактов. Например, рассмотрим следующую иерархию ссылок таблицы FactTable1 → DimTable1 → DimTable2 В примере выше DimTable1 является ссылкой уровня 1, а DimTable2 — ссылкой уровня 2 из таблицы FactTable1. Если для параметра referenceLevel выбрано значение “1” для таблицы FactTable1, то созданный DDL таблицы промежуточного хранения для таблицы фактов “FactTable1” будет включать столбец для каждого



Атрибут	Обязательный	Описание
		столбца бизнес-ключа из таблицы “FactTable1” и столбец для каждого столбца бизнес-ключа из таблицы “DimTable1”, но не будет включать столбец для каждого столбца бизнес-ключа из таблицы “DimTable2” (таблица “DimTable2” будет пропущена).
enrichDowntime	Нет	Булево значение, указывающее, включено ли добавление сведений о простоях для таблицы фактов.  Значение по умолчанию: true.  Примечание: Это значение переопределяет значение enrichDowntime, заданное родительским элементом factTables
downtimeDimension	Нет	Таблица измерений, простои которой должны учитываться при добавлении сведений о простоях в таблицу фактов.  Значение по умолчанию — основная таблица измерений (таблица измерений, на которую ссылается столбец dsi_key_id_ в таблице фактов) для таблицы фактов.  Примечание: Таблица фактов должна ссылаться на таблицу измерений

## Элемент XML: dimensionTables

Атрибут	Обязательный	Описание
strategy	Да	Глобальная стратегия создания схемы таблицы промежуточного хранения для таблиц измерений. Допустимые значения <ul style="list-style-type: none"> <li>• normalize</li> </ul> Созданный DDL таблицы промежуточного хранения создается для каждой таблицы измерений и будет включать столбец для каждого столбца в таблице измерений и столбец для каждого столбца бизнес-ключа из всех родительских таблиц измерений. <ul style="list-style-type: none"> <li>• Denormalize</li> </ul> Эта стратегия не поддерживается в настоящее время

Атрибут	Обязательный	Описание
		Значение по умолчанию: <code>normalize</code>
<code>referenceLevel</code>	Нет	<p>Уровень ссылки таблицы должен учитываться при создании схемы для таблицы промежуточного хранения для всех таблиц измерений.</p> <p>Например, рассмотрим следующую иерархию ссылок таблицы</p> <p><code>DimTable1 → DimTable2 → DimTable3</code></p> <p>В примере выше <code>DimTable2</code> является ссылкой уровня 1, а <code>DimTable3</code> — ссылкой уровня 2 из таблицы <code>DimTable1</code>. Если для параметра <code>referenceLevel</code> выбрано значение “1” для таблицы измерений <code>DimTable1</code>, то созданный DDL таблицы промежуточного хранения для таблицы измерений “<code>DimTable1</code>” будет включать столбец для каждого столбца бизнес-ключа из таблицы “<code>DimTable1</code>” и столбец для каждого столбца бизнес-ключа из таблицы “<code>DimTable2</code>”, но не будет включать столбец для каждого столбца бизнес-ключа из таблицы “<code>DimTable3</code>” (таблица “<code>DimTable3</code>” будет пропущена).</p>

Дочерние элементы	Множество	Описание
<code>table</code>	0..n	Список таблиц измерений и конфигурация стратегии.

Элемент XML: **table**

Атрибут	Обязательный	Описание
<code>name</code>	Да	<p>Имя таблицы измерений. Имя должно совпадать с именем таблицы фактов, заданным в XML модели.</p> <p>Примечание: Имя чувствительно к регистру.</p>
<code>strategy</code>	Нет	<p>Глобальная стратегия создания схемы таблицы промежуточного хранения для таблиц измерений.</p> <p>Допустимые значения</p> <ul style="list-style-type: none"> <li><code>normalize</code></li> </ul> <p>Созданный DDL таблицы промежуточного хранения создается для таблицы измерений и будет включать</p>

Атрибут	Обязательный	Описание
		<p>столбец для каждого столбца в таблице измерений и столбец для каждого столбца бизнес-ключа из всех родительских таблиц измерений.</p> <ul style="list-style-type: none"> <li>Denormalize</li> </ul> <p>Эта стратегия не поддерживается в настоящее время</p> <p>Значение по умолчанию: normalize</p>
referenceLevel	Нет	<p>Уровень ссылки таблицы должен учитываться при создании схемы для таблицы промежуточного хранения для таблицы измерений.</p> <p>Например, рассмотрим следующую иерархию ссылок таблицы</p> <p>DimTable1 → DimTable2 → DimTable3</p> <p>В примере выше DimTable2 является ссылкой уровня 1, а DimTable3 — ссылкой уровня 2 из таблицы DimTable1. Если для параметра referenceLevel выбрано значение "1" для таблицы DimTable1, то созданный DDL таблицы промежуточного хранения для таблицы измерений "DimTable1" будет включать столбец для каждого столбца бизнес-ключа из таблицы "DimTable1" и столбец для каждого столбца бизнес-ключа из таблицы "DimTable2", но не будет включать столбец для каждого столбца бизнес-ключа из таблицы "DimTable3" (таблица "DimTable3" будет пропущена).</p>
groupBridge	Нет	<p>Булево значение, указывающее, что таблица измерений является мостовой таблицей группы.</p> <p>Значение по умолчанию: false.</p>
locationBridge	Нет	<p>Булево значение, указывающее, что таблица измерений является мостовой таблицей расположения. Значение по умолчанию: false.</p>

Элемент XML: **stageExclusions**

Дочерние элементы	Множество	Описание
<a href="#">excludeTable</a>	1..n	Список таблиц фактов/измерений для исключения из создания таблицы промежуточного хранения

Элемент XML: **excludeTable**

Атрибут	Обязательный	Описание
name	Да	Имя таблицы измерений/фактов. Имя должно совпадать с

Атрибут	Обязательный	Описание
		именем таблицы фактов/измерений, заданным в XML модели

Элемент XML: **downtimeExclusions**

Дочерние элементы	Множество	Описание
<a href="#">dimensionTable</a>	1..n	Список таблиц измерений, которые не следует при создании таблицы простоев.

Элемент XML: **dimensionTable**

Атрибут	Обязательный	Описание
name	Да	Имя таблицы измерений, которую не следует при создании таблицы простоев. Имя должно совпадать с именем таблицы измерений, заданным в XML модели

## Пример определения стратегии

### Пример XML-файла стратегии

См. пример XML-файла стратегии `strategy.xml` и соответствующую XML-модель `model.xml`, которые доступны в разделе вложений и определяют стратегию для таблиц файлов и измерений, исключений промежуточного хранения и простоев.

## Приложение A: Фильтры и функции в политиках SHR ETL

SHR предоставляет набор фильтров и функций, которые можно использовать в таких политиках ETL, как политика сбора данных RTSM, политика сбора данных OM, политики сверки и преобразования для очистки/преобразования собранных данных. Ниже представлен список фильтров и функций, поддерживаемых SHR

### Фильтры

Поддерживаемые условия фильтрации

**<SOURCE\_COLUMN> NOT IN (VALUE1, VALUE2 ...)** — это условие проверяет, содержит ли SOURCE\_COLUMN любое из значений в представленном списке. Если это так, условие возвращает значение false и строка отклоняется для выходного набора данных, в противном случае строка принимается.

**<SOURCE\_COLUMN> IN (VALUE1, VALUE2 ...)** — это условие проверяет, содержит ли SOURCE\_COLUMN любое из значений в представленном списке. Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> LIKE <VALUE\_PATTERN>** — это условие проверяет, совпадает ли SOURCE\_COLUMN с указанным значением VALUE\_PATTERN. Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> = <VALUE >** — это условие проверяет, равно ли значение SOURCE\_COLUMN указанному значению VALUE. Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> != <VALUE >** — это условие проверяет неравенство значения SOURCE\_COLUMN указанному значению VALUE. Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> GREATER THAN <VALUE >** — это условие проверяет, что значение SOURCE\_COLUMN больше указанного значения VALUE (поддерживается численное и строковое сравнение). Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> GREATER THAN EQUAL TO <VALUE >** — это условие проверяет, что значение SOURCE\_COLUMN больше или равно указанному значению VALUE (поддерживается численное и строковое сравнение). Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> LESSER THAN <VALUE >** — это условие проверяет, что значение SOURCE\_COLUMN меньше указанного значения VALUE (поддерживается численное и строковое сравнение). Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

**<SOURCE\_COLUMN> LESSER THAN EQUAL TO <VALUE >** — это условие проверяет, что значение SOURCE\_COLUMN меньше или равно указанному значению VALUE (поддерживается численное и строковое сравнение). Если это так, условие возвращает значение true и строка принимается для выходного набора данных, в противном случае строка отклоняется.

Условия фильтрации можно сочетать с помощью операторов **AND** и/или **OR**. Несколько условий фильтрации можно сгруппировать, заключив их в квадратные скобки **[ ]** :

Пример сложного условия фильтрации представлен ниже

```
[BYLS_LS_ROLE=RESPOOL AND BYLS_LS_HOST_HOSTNAME NOT IN (NA)] OR BYCPU_ID_UTIL GREATER THAN EQUAL TO 100]
```

## Функции

Следующие функции/операторы можно использовать с условиями.

**OPR\_TOLOWER** – использование: **OPR\_TOLOWER(<имя столбца >)**

Эта функция преобразует значение указанного столбца CSV-файла в нижний регистр

***OPR\_TOUPPER – использование: OPR\_TOUPPER(<имя столбца>)***

Эта функция преобразует значение указанного столбца CSV-файла в верхний регистр

***OPR\_STRINGLENGTH – использование: OPR\_STRINGLENGTH(<имя столбца>)***

Эта функция возвращает длину строкового значения указанного столбца CSV-файла

***OPR\_SUBSTRING – использование: OPR\_SUBSTRING(<имя столбца>, <начальный индекс>) или OPR\_SUBSTRING(<имя столбца>, <start index>, <конечный индекс>)***

Эта функция возвращает подстроку значения указанного столбца CSV-файла. Эта функция может включать два или три аргумента. Первый аргумент — имя столбца, второй — начальный индекс подстроки, а третий — конечный индекс подстроки. Затем функция substring возвращает подстроку для указанного имени столбца с начального индекса (включительно) до конечного индекса, если он указан, в противном случае возвращается вся строка до конца.

***OPR\_APPEND – использование: OPR\_APPEND(<имя столбца 1>, <имя столбца 2>, <разделитель>)***

Эта функция дополняет значения указанных столбцов с указанным разделителем, возвращает <значение столбца 1><разделитель><значение столбца 2>. Кроме того, если разделитель строк должен быть “,” (COMMA) или “ ” (SPACE), то <разделитель> должен быть указан как OPR\_FIXEDVALUE(COMMA) или OPR\_FIXEDVALUE(SPACE).

***OPR\_SUM – использование: OPR\_SUM(<имя столбца 1>, <имя столбца 2>, <имя столбца 3>)***

Эта функция находит сумму значений из CSV-файла для указанных столбцов, возвращает <значение столбца 1> + <значение столбца 2> + <значение столбца 3>

***OPR\_DIFFERENCE – использование: OPR\_DIFFERENCE(<имя столбца 1>, <имя столбца 2>, <имя столбца 3>)***

Эта функция находит разность значений из CSV-файла для указанных столбцов, возвращает <значение столбца 1> - <значение столбца 2> - <значение столбца 3>

***OPR\_PRODUCT – использование: OPR\_PRODUCT(<имя столбца 1>, <имя столбца 2>, <имя столбца 3>)***

Эта функция рассчитывает произведение значений из CSV-файла для указанных столбцов, возвращает <значение столбца 1>\*<значение столбца 2>\*<значение столбца 3>

***OPR\_DIVIDE – использование: OPR\_DIVIDE(<имя столбца 1>, <имя столбца 2>, <имя столбца 3>)***

Эта функция делит значения из CSV-файла для указанных столбцов друг на друга, возвращает <значение столбца 1>/<значение столбца 2>/<значение столбца 3>

***OPR\_PERCENTILE – использование OPR\_PERCENTILE(<имя столбца 1>, <имя столбца 2>)***

Эта функция возвращает процентное значение для значений двух столбцов, возвращает <значение столбца 1>/<значение столбца 2>\*100

***OPR\_AVERAGE – использование OPR\_AVERAGE(<имя столбца 1>, <имя столбца 2>, <имя столбца 3>)***

Эта функция возвращает среднее для значений столбцов, указанных в функции, возвращает (<значение столбца 1>+<значение столбца 2>+<значение столбца 3>)/<количество столбцов>

***OPR\_MATHFLOOR – использование OPR\_MATHFLOOR(<имя столбца>)***

Эта функция находит минимальное значение для указанного столбца, возвращает математический минимум для <значение столбца>

***OPR\_MATHCEIL – использование OPR\_MATHCEIL(<имя столбца>)***

Эта функция находит максимальное значение для указанного столбца, возвращает математический максимум для <значение столбца>

***OPR\_FIXEDVALUE – использование OPR\_FIXEDVALUE(<строка>)***

Эта функция возвращает строку, указанную в функции, как есть и не обрабатывает ее как столбец csv-файла, возвращает <строка>

***OPR\_MAXVALUE – использование OPR\_MAXVALUE(<имя столбца 1>,<имя столбца 2>,<имя столбца 3>)***

Эта функция возвращает максимум среди значений столбцов, указанных в функции, возвращает максимум из <значение столбца 1>, <значение столбца 2>, <значение столбца 3> и т.д.

***OPR\_MINVALUE – использование OPR\_MINVALUE(<имя столбца 1>,<имя столбца 2>,<имя столбца 3>)***

Эта функция возвращает минимум среди значений столбцов, указанных в функции, возвращает минимум из <значение столбца 1>, <значение столбца 2>, <значение столбца 3> и т.д.

***OPR\_STRINGSPLIT – использование OPR\_STRINGSPLIT(<имя столбца>,<строковое выражение, которое будет использоваться для разделения>,<индекс разделенных данных для возврата>)***

Эта функция использует строковое выражение, указанное вторым, для разделения значения имени указанного столбца и возвращает значение из массива строк после разделения с указанным индексом. Обратите внимание, что этот индекс начинается с 0. Кроме того, при использовании разделителя “,” (COMMA) или “ ” (SPACE) выражение должно быть введено как OPR\_FIXEDVALUE(COMMA) или OPR\_FIXEDVALUE(SPACE) соответственно.

Функции, указанные выше, могут быть вложенными. Например, вложенная функция ниже действительна

***OPR\_SUBSTRING(OPR\_TOLOWER(BYLS\_LS\_HOSTNAME),0,OPR\_DIFFERENCE(OPR\_STRINGLENGTH(HOST\_HOSTNAME),1***

Кроме того, в строках условий можно использовать выражения if-then-else для отдельных записей, но не на уровне набора записей. Формат: <condition string>?<operator if true>:<operator if false>. Также следует учесть, что только один условный оператор действует для записи. Сама строка условия условного оператора может быть сложной и включать операторы. Пример строки выражения —

***[OPR\_TOUPPER(BYLS\_LS\_HOSTNAME) LIKE \*IND.HP.COM]?OPR\_FIXEDVALUE(INDIA):OPR\_TOUPPER(HOST\_DNSNAME)***

Выражение выше сведется к INDIA, если значение BYLS\_LS\_HOSTNAME соответствует шаблону \*.IND.HP.COM, в противном случае оно будет равно HOST\_DNSNAME в верхнем регистре

## Поддерживаемые функции в элементах Aggregate и Forecast

В таблице ниже перечислены поддерживаемые функции и их описания:

Имя функции	Описание	Поддерживается элементом Aggregate	Поддерживается элементом Forecast
avg	Среднее	Да	Да
min	Минимум	Да	Да
max	Максимум	Да	Да
cnt	Число	Да	Да
tot	Всего	Да	Да
med	Медиана	Да	Да
std	Стандартное отклонение	Да	Да
slope	Наклон	Да	Да
wav	Взвешенное среднее	Да	Да
perXX	Процентиль (где XX — любое значение менее 100)	Да	Да
lst	Последнее значение	Да	Нет
nlst	Последнее значение, отличное от null	Да	Нет
fXX	Прогноз (где XX — любое целое число)	Нет	Да
DTT[XX]	Дней до порога (где XX — любое целое число)	Нет	Да

## Ограничения

Ниже представлены ограничения при указании фильтров

1. Функции, указанные в этом документе, не могут применяться для значений компаратора при использовании условий IN , NOT IN и LIKE
2. Допускается только одна строка выражения на строку условия



# Глоссарий

## C

---

### **CDE**

CDE is a set of tools provided by SHR for development of content packs.

### **Collection policy**

A Collection policy is written in XML to define the metrics to be collected by a collector program from the specified data source.

### **Content pack**

Content packs are data marts deployed on the SHR performance management database platform. Content packs enable the platform to collect, store, process, and report the data. A content pack has three components – Domain, ETL, and Reports.

## D

---

### **Data Model**

Data model is a schema diagram that illustrates the relationship between dimension tables (that have attributes) and fact tables (that have measures).

### **Domain Component**

Domain component of a content pack defines the data model of the Domain you are reporting on along with the logic to perform processing on the data. It is independent of the data source you collect data from.

## E

---

### **ETL Component**

ETL component of a content pack is data source dependant; it defines the collection of

data from the specified data source

## L

---

### **Loading**

The process of loading data from the stage tables to the data warehouse tables.

## R

---

### **Reconciliation**

Data reconciliation is the technique of associating fact data to corresponding dimension data.

### **Reports Component**

The Reports component contains the SAP BusinessObjects Web Intelligence reports and universes.

## S

---

### **Staging**

Data staging is the process of moving the collected, transformed, and reconciled data into the staging tables.

## T

---

### **Transformation**

Data transformation is the optional step of cleaning the collected data according to business requirements.

## W

---

### **Workflow Streams**

Workflow streams in content packs are used to define and control the movement of data from one step to another.

# Отправить отзыв о документации

Если вы хотите прокомментировать этот документ, [отправьте электронное письмо](#) в группу разработки документации. Если в этой системе сконфигурирован почтовый клиент, щелкните ссылку вверху, и окно электронной почты откроется со следующей информацией в строке темы.

## **Отзыв о документе Руководство по разработке содержимого (Service Health Reporter 9.40)**

Просто впишите свой отзыв и нажмите кнопку "Отправить".

Если почтовый клиент отсутствует, скопируйте приведенный ниже текст в новое сообщение в почтовом веб-клиенте и отправьте его по адресу [docfeedback@hp.com](mailto:docfeedback@hp.com).

Нам важно ваше мнение!