



Ред База Данных

Версия 2.5

**Руководство по
репликации**

© Корпорация Ред Софт 2011

Данный документ содержит описание использования программы репликации «Ред Репликатор», входящей в состав «Ред База Данных». Документ рассчитан на пользователей, знакомых с принципами организации баз данных СУБД «Ред База Данных» и языком SQL.

www.red-soft.biz

Содержание

Введение.....	4
1 Установка и настройка.....	5
1.1 Файл конфигурации.....	5
2 Основные операции.....	7
2.1 Работа с сервисом репликатора.....	7
2.2 Параметры командной строки.....	7
3 Работа с источниками данных.....	9
3.1 Типы поддерживаемых источников.....	10
3.1.1 СУБД-источники.....	10
3.1.2 Источники-файлы.....	12
4 Работа с подписчиками данных.....	15
5 Экспорт данных.....	18
5.1 Режимы экспорта.....	18
5.2 Форматы экспорта.....	18
5.2.1 Формат двоичного файла	18
5.2.2 Формат XML файла	19
5.2.3 Формат CSV файла	19
6 Работа с репликатором через FTP.....	21
6.1 Структура каталогов FTP-сервера.....	21
6.2 Публикация дельты через FTP.....	22
6.3 Добавление дельты через FTP.....	23
7 Программные интерфейсы «Ред Репликатор».....	24
Приложение А - Служебные таблицы «Ред Репликатор».....	27

Введение

Программа «Ред Репликатор» (далее - репликатор) — одна из составных частей системы контроля и восстановления целостности СУБД «Ред База Данных», наряду со средствами контроля целостности файлов сервера, метаданных баз данных и средствами резервного копирования и восстановления баз данных, входящих в поставку СУБД «Ред База Данных»¹

Программа «Ред Репликатор» предназначена для:

- синхронизации информации в базах данных — источниках (или файлах), и Мастер базе;
- создания полных резервных копий Мастер-базы баз данных (mirrors), или выборок нужных данных по шаблонам (reflections);
- поддержание этих копий в актуальном состоянии;
- публикации на FTP полных копий БД или изменений в базе за определенный период.

Основные понятия и определения, используемые в данном документе:

Репликация данных – это процесс создания их точной копии, без прерывания работы приложений. Такая копия (или сразу несколько копий) может располагаться как на другом локальном диске сервера, так и храниться на удаленном компьютере, на случай сбоя в работе всего главного вычислительного центра организации.

Источник данных(Source) – база данных или файл, предоставляющая хранящуюся в ней информацию для тиражирования на другие сервера.

Мастер-база(Master) – база данных, хранящая точную копию информации одного или более источников.

Подписчик данных(Subscriber) – база данных, принимающая тиражируемую источником данных информацию.

Дельта(delta) – файл, содержащий изменения с момента последней публикации

Лист(list) – файл, содержащий все данные подписчика.

¹ Подробнее см. Руководство администратора

1 Установка и настройка

«Ред Репликатор» входит в поставку всех версий «Ред База Данных», кроме Открытой редакции. Он располагается в каталоге \tools\Replicator каталога установки «Ред Базы Данных».

1.1 Файл конфигурации

Настройка параметров работы репликатора производится с помощью конфигурационного файла - redreplicator.xml. Этот файл состоит из двух разделов – общие настройки (CommonSettings) и настройки хранилища данных (MasterDatabases).

Секция CommonSettings – содержит общие настройки работы реплткатора:

- UpdateInterval – частота обновления созданного архива для поддержания его в актуальном состоянии (мин.);
- AutoCommit – количество изменений, полученных от источника, после которых делается подтверждение транзакции (по умолчанию - 10000);
- StopOnError – остановка сервиса при ошибке (по умолчанию - false);
- DefaultDatabase – псевдоним мастер-базы по умолчанию;
- FTP – настройки FTP-сервера
- Enabled – доступность FTP-сервиса;
- FtpCommandPort – порт команд FTP (по умолчанию - 21);
- FtpDataPort – порт данных FTP (по умолчанию - 20);

Каждая мастер-база должна быть описана внутри тэга <Masterdatabases>...</Masterdatabases> файле конфигурации следующим образом:

- AliasName – Псевдоним мастер-базы;
- ConnectionString – путь до базы банных в сети или на локальном компьютере. Имеет вид хост:путь, например localhost:C:\DataBase\Master.fdb;
- UserName – имя пользователя базы данных;
- Password – пароль пользователя;
- DefaultSubscriber – подписчик по умолчанию;

Пример файла конфигурации:

```
<?xml version="1.0" ?>
<ReplicatorConfig>
  <CommonSettings UpdateInterval="1" AutoCommit="10000"
  StopOnError="False" DefaultDatabase="Master">
    <FTP Enabled="true" FtpCommandPort="21"
    FtpDataPort="20"/>
  </CommonSettings>
  <MasterDatabases>
```

```
<Database AliasName="Master" ConnectionString="C:\
Example\MASTER.FDB" Physical="" UserName="SYSDBA"
Password="masterkey" DefaultSubscriber="" />

<!--

<Database AliasName="Test2"
ConnectionString="127.0.0.1:D:\example.fdb"
Physical="" UserName="SYSDBA"
Password="masterkey" DefaultSubscriber="">

</Database> !

-->

</MasterDatabases>
</ReplicatorConfig>
```

2 Основные операции

Репликатор может использоваться из командной строки и как служба (сервис) операционной системы. Командная строка необходима для настройки репликации данных – добавление, удаление, активация и деактивация подписчиков/источников, экспорт данных, и др.

Сервис же работает независимо от пользователя, и предназначен для выполнения автоматических операций, например обновления подписчиков, а так же для работы FTP-сервера.

2.1 Работа с сервисом репликатора

Управление сервисом репликатора:

```
redreplicator [-install | -uninstall]
```

В версии для операционной системы Linux сервис репликатора запускается командой

```
./RedReplicator -service -daemon
```

Если не указать параметр `-daemon`, сервис репликатора не уйдет в фоновый режим.

Сервис работает согласно настройкам файла конфигурации (см. п. 1.1).

2.2 Параметры командной строки

Репликатор имеет следующие опции:

- `-target <db-name>` – использовать мастер-базу, указанную в файле конфигурации в секции `<MasterDatabases>`, но отличную от указанной в параметре `DefaultDatabase`;
- `-config <configfile>` – Указать файл конфигурации отличный от файла по умолчанию;
- `-local` – Использование репликатора в качестве DCOM-клиента для удаленного репликатора;
- `-verbose <mode>` – Режим записи лога. Параметр `<mode>` может принимать два значения: 1 – краткий режим лога, 2 – более подробный режим. По умолчанию используется краткий режим лога.
- `-nolog` – Не записывать результат выполнения команды в лог;

Данные ключи могут быть указаны при выполнении следующих операций с помощью репликатора:

- `activate` – активация репликации;
- `deactivate` – остановка репликации;
- `add <subscriber> <connection-string> [descriptor <desc.xml> physical <filename>] [disabled]` – создание подписчика. Здесь `subscriber` – псевдоним подписчика, `connection string` – строка соединения с базой данных подписчика в формате `<host>:<db_name|alias>` `descriptor` – путь к файлу дескрип-

- тора подписчика, `physical` – путь к базе данных подписчика. Опция Опция `disabled` – указывает на создание не активированного подписчика;
- `disabled` – создание неактивного подписчика (если параметр не указан, то по умолчанию будет создан активный подписчик);
 - `addsrc <source> <descriptor> [disabled]` – создание источника. Здесь `source` – псевдоним источника, `descriptor` – путь к файлу дескриптору источника. Опция `disabled` – указывает на создание не активированного источника.
 - `delete <sub | src> <subscriber | source>` – удаление подписчика или источника. Здесь `sub|src` – объект операции: `sub` – источник, `src` – подписчик, `subscriber | source` – псевдоним источника/подписчика.
 - `enable <sub | src> <subscriber | source>` – активация подписчика или источника;
 - `disable <sub | src> <subscriber | source>` – деактивация подписчика или источника;
 - `reset <subscriber> [descriptor <desc.xml>] [physical <filename>]` – пересоздание подписчика базы;
 - `update <sub | src> <subscriber | source> | ALL` – обновление подписчика из мастер-базы либо мастер-базы из источника. Параметр `ALL` означает обновление всех подписчиков, либо из всех источников;
 - `validate <subscriber>` – проверка соответствия структуры базы подписчика и мастер-базы;
 - `ftp [<control-port> [<data-port>]]` – запуск FTP-сервера в консольном режиме. Опция доступна только для отладки с базой данных с названием «Test»;
 - `export <filename> <generation | list> <subscriber> [<TableName>] [bin | xml | csv]` – экспорт данных:
 - `filename` – имя файла;
 - `generation | list` – режим экспорта²;
 - `subscriber` – псевдоним подписчика;
 - `TableName` – таблица подписчика;
 - `bin|xml|csv` – формат файла – бинарный, xml-, или csv-файл;
 - `executescript <filename> [<subscriber>]` – выполнить файл с sql-командами.

2 Подробнее см. пункт 5.1

3 Работа с источниками данных

Синхронизация баз данных-источников, с базой-приемником (хранилищем) производится на основании дескриптора источника данных. Дескриптор представляет собой XML-файл, формат которого зависит от типа источника. Дескриптор определяет, какие именно данные из источника подлежат извлечению в мастер-базу.

С источниками данных предусмотрены следующие операции: добавление, удаление, активация, и обновление.

Для добавления источника необходимо в консоли выполнить команду:

```
Redreplicator addsrc <source> <path>
```

Здесь:

- `source` – псевдоним источника. Каждый источник должен иметь уникальный псевдоним.
- `path` – абсолютный или относительный путь к файлу дескриптора

Для удаления источника необходимо выполнить команду:

```
Redreplicator delete src <source>
```

После того, как источник добавлен, необходимо провести его активацию. Активация источника определяет возможность синхронизации с ним. Новый источник добавляется как активный. Для активации/деактивации предусмотрены команды:

```
Redreplicator enable src <имя_источника>
```

```
Redreplicator disable src <имя_источника>
```

Обновление источника подразумевает собой процесс копирования изменений в источнике в мастер-базу. Обновление может быть выполнено в ручную — при работе с репликатором в режиме командной строки. Для этого необходимо выполнить команду:

```
Redreplicator update src <имя_источника> /target  
<псевдоним_master-базы>
```

```
Redreplicator update src ALL
```

В первом случае будет выполнено обновление из указанного источника, во втором – из всех доступных (`enable`) источников.

Если не указан параметр `/target`, обновление будет производиться в мастер-базу заданную в конфигурационном файле репликатора по умолчанию.

Также синхронизация может производиться автоматически — если репликатор запущен как системная служба (или демон). Для указания интервалов обновления используется параметр `UpdateInterval` из файла конфигурации репликатора.

3.1 Типы поддерживаемых источников

- СУБД – доступ к данным будет осуществляться через клиентский API интерфейс этой СУБД;
- ODBC – подключение к базе данных производится через драйвер ODBC, с указанием строки подключения (менее предпочтительный вариант по быстродействию);
- FTP – внешний источник самостоятельно загружает изменения на FTP - сервер репликатора.
- FILE – источник самостоятельно экспортирует файл в папку Inbox репликатора.

Репликатор может работать с базами данных следующих СУБД:

- MySQL
- MSSQL Server 2005 (MSSQL Server 2000 доступен только через ODBC).
- Oracle
- Access
- Excel
- dBase (*.dbf файлы)

В дескрипторе тип источника (Type) указывается аналогичным образом, за исключением MSSQL Server – MSSQL.

3.1.1 СУБД-источники

Дескриптор для СУБД-источника представляет собой XML-файл, который имеет следующую структуру:

```
<Source Type="" Server="" Database="" DataSource=""
UserName="" Password="">
  <Object TableName="">
    <SourceSQL> </SourceSQL>
  <Filter/>
  <PrimaryKey>
    <Field Name="" />
  </PrimaryKey>
  <Fields>
    [<Field Name="" SourceName="" />]
  </Fields>
  [<Filter></Filter>]
</Object>
</Source>
```

Тэг Source – общее описание базы данных источника:

- Type – тип базы данных (описаны в главе 3.1);
- Server – имя или адрес сервера базы данных;
- Database – имя базы данных;
- DataSource – псевдоним источника в системном DNS, используется только для ODBC-источников как замена параметру Database;
- UserName – имя пользователя для подключения к БД;
- Password – пароль пользователя;

Тэг Object – задает описание таблиц для синхронизации

Тэг SourceSQL – SQL-запрос для выборки из источника

Тэг Filter – фильтр данных (необязателен) Фильтр записывается в дескриптор следующим образом:

```
<Filter>FieldName=Value</Filter>
```

- FieldName – имя поля;
- Value – значение.

Так же возможно использование некоторых функций, например:

```
<Filter>datefield=cast(${now} as date) - 30</Filter>
```

Тэг PrimaryKey – описание первичного ключа таблицы, здесь Field – имя поля первичного ключа.

Поля, указанные в *Fields* – особые названия полей в мастер-базе. Если поля не будут указаны – название полей будет аналогично названию этих полей в источнике данных.

- Name – название поля в мастер-базе;
- SourceName – оригинальное название поля;

Элементы Field могут содержать дополнительные атрибуты:

- FieldType – integer/string/blob/clob/date/timestamp и т.п.;
- FieldSize – для типов, требующих длину;
- FieldPrecision – для типов, требующих указание точности;
- XMLPlacement – element/attribute/cdata;

Пример дескриптора для СУБД-источника:

```
<Source Type="MySQL" Server="localhost"
Database="MySQL_Source" DataSource="" UserName="root"
Password="">
<Object TableName="MySQL_FirstTable">
<SourceSQL>select * from First_Table</SourceSQL>
<Filter/>
<PrimaryKey>
  <Field Name="FT_ID" />
</PrimaryKey>
</Object>
</Source>
```

```
</PrimaryKey>
<Fields>
  <Field Name="ID" SourceName="FT_ID" />
  <Field Name="F_INT" SourceName="FT_INT" />
  <Field Name="F_STR" SourceName="FT_STR" />
  <Field Name="F_TEXT" SourceName="FT_TEXT" />
  <Field Name="F_DATE" SourceName="FT_DATE" />
</Fields>
</Object>
</Source>
```

3.1.2 Источники-файлы

В качестве источников-файлов могут выступать файлы XML- и CSV-формата.

```
<Source Type="FILE">
  <DataFormat DateFormat="" TimeFormat=""
    DateTimeFormat="" NullString="" FolderFormat="" />
<Object TableName="" XMLRowsPath="">
<PrimaryKey>
  <Field Name="" />
</PrimaryKey>
<Fields>
  [<Field Name="" SourceName="" FieldType="" />...]
</Fields>
</Object>
</Source>
```

Тэг DataFormat – описание формата данных

- DateFormat – формат даты;
- TimeFormat – формат времени;
- DateTimeFormat – формат дата+время;
- NullString – формат пустой строки;
- FloatDelimiter – разделитель дробных чисел;
- CSVComma – разделитель полей в CSV-файле;
- CSVQuote – цитата в CSV-файле;
- FolderFormat – формат каталога, в котором хранится источник-файл;

Тэг Object:

- TableName – имя таблицы в мастер-базе;
- XMLRowsPath – строка, определяющая имя тега в xml-источнике, тэги которого содержат описание данных, т.е. саму запись.

Тэги Filter, PrimaryKey и Fields аналогичны тэгам в дескрипторе базы данных, тэг Fields в данном случае является обязательным. Также поддерживается параметр `{now}` – текущая дата (рекомендуется использовать его только для баз RDB/Firebird).

После добавления источника репликатор сформирует структуру каталогов для работы с данным источником

Для обновления мастер-базы из источника-файла необходимо:

1. Войти в директорию <каталог_репликатора/inbox/псевдоним_мастер-базы/псевдоним_источника>;
2. войти в подкаталог, соответствующий типу обновления (delta, list);
3. войти в подкаталог, соответствующий типу файла (xml, csv, bin);
4. войти в подкаталог, соответствующий названию таблицы;
5. создать каталог в соответствии с форматом, указанным в дескрипторе источника (FolderFormat).
6. скопировать файл в созданный каталог и выполнить команду обновления.

Пример дескриптора для источника-файла:

```
<Source Type="FILE">
<DataFormat DateFormat="dd.mm.yyyy"
TimeFormat="hh:mm:ss" DateTimeFormat="dd.mm.yyyy
hh:nn:ss" NullString="null" FolderFormat="yyyy-mm-dd
hh-nn-ss-zzz" />
<Object TableName="XML_TABLE" XMLRowsPath="Rows">
<PrimaryKey>
  <Field Name="ID" />
</PrimaryKey>
<Fields>
  <Field Name="ID" SourceName="ID"
  FieldType="Integer" />
  <Field Name="DATETIME_F" SourceName="DATETIME_F"
  FieldType="DateTime" />
  <Field Name="DATE_F" SourceName="DATE_F"
  FieldType="Date" />
  <Field Name="TIME_F" SourceName="TIME_F"
  FieldType="Time" />
</Fields>
</Object>
```

```
</Source>
```

Пример xml-источника:

```
<Publication Signature="RedReplicator" Type="list"
TableName="XML_TABLE">
<Rows>
  <Row P_ACTION="merge" ID="1"
  DATETIME_F="12.12.2112 21:21:12"
  DATE_F="14.12.2014" TIME_F="14:14:14" />
  <Row P_ACTION="merge" ID="2"
  DATETIME_F="02.02.2002 20:02:20" DATE_F="null"
  TIME_F="null" />
  <Row P_ACTION="merge" ID="3" DATETIME_F="null"
  DATE_F="14.12.2014" TIME_F="null" />
  <Row P_ACTION="merge" ID="4"
  DATETIME_F="30.03.3003 03:30:30" DATE_F="null"
  TIME_F="14:14:14" />
  <Row P_ACTION="merge" ID="5" DATETIME_F="null"
  DATE_F="14.12.2014" TIME_F="null" />
</Rows>
</Publication>
```

4 Работа с подписчиками данных

Для работы с подписчиками данных предусмотрен следующий набор операций: добавление, удаление, активация/деактивация, пересоздание, обновление, и проверка соответствия.

Для добавления подписчика необходимо в консоли выполнить команду:

```
redreplicator add <subscriber> <connection-string>  
[descriptor <desc.xml>|physical <filename>] [disabled]
```

- subscriber – псевдоним подписчика
- connection string – строка соединения с базой данных подписчика в формате host: pathalias
- descriptor – путь к файлу дескриптора подписчика
- physical – путь к сетевому каталогу с базой данных подписчика
- disabled – создание неактивного подписчика

Для удаления источника необходимо выполнить команду:

```
Redreplicator delete sub <subscriber>
```

Активация подписчика определяет возможность синхронизации с ним. Новый источник добавляется как активный. Для активации/деактивации предусмотрены команды:

```
Redreplicator enable sub <subscriber>  
Redreplicator disable sub <subscriber>
```

Пересоздание подписчика – процесс изменения информации и полной синхронизации с мастер-базой. Для этого необходимо выполнить команду:

```
Redreplicator reset <subscriber> [descriptor  
<descriptor_name>] [physical <filename>]
```

Пересоздание может использоваться, например, в случае изменения структуры источников, и, соответственно, мастер-базы.

Обновление источника подразумевает собой процесс копирования изменений из мастер-базы в базы-подписчики. Обновление может быть выполнено вручную — при работе с репликатором в режиме командной строки. Для этого необходимо выполнить команду:

```
Redreplicator update sub <subscriber>  
Redreplicator update sub ALL
```

Для проверки соответствия мастер-базы и базы подписчика предусмотрена команда

```
Redreplicator validate <subscriber>
```

Синхронизация подписчиков с мастер-базой выполняется на основе дескриптора, аналогично источникам. Дескриптор имеет следующую структуру:

```
<Publication>
<DataFormat DateFormat="" TimeFormat=""
DateTimeFormat="" NullString="" FloatDelimiter=""
CSVDelimiter="" CSVQuote="" />
<Object TableName="" FullSync="">
<PrimaryKey>
  <Field Name="" />
</PrimaryKey>
<Fields>
  [<Field Name="" FieldType="" />...]
</Fields>
<FromClause />
<FilterClause />
<ChangedRecordsSQL />
</Object>
</Publication>
```

Тэг `DataFormat` - аналогично форматам источника;

Тэг `Object`:

- `TableName` – имя результирующей таблицы на подписчике;
- `FullSync` – атрибут, указывающий на необходимость делать полное сравнение записей.

Тэг `Fields` – аналогичен тэгу в дескрипторе источников;

Тэг `PrimaryKey` – аналогичен тэгу в дескрипторе источников;

- `<FromClause>` - часть SQL запроса со списком выборки;
- `<FilterClause>` - условия фильтрации выборки;
- `<ChangedRecordsSQL>` - SQL запрос получения первичных ключей только изменившихся записей.

Если указана одна таблица во `FromClause` то первичный ключ может определяться автоматически. В этом случае также автоматически генерируется `ChangedRecordsSQL`, иначе, необходимо прописать запрос в `ChangedRecordsSQL` вручную, либо установить атрибут `FullSync="1"`.

Пример дескриптора подписчика:

```
<Publication>
<DataFormat DateFormat="dd.MM.yyyy" TimeFormat="hh:mm"
DateTimeFormat="dd.MM.yyyy hh:nn:ss" NullString="Null"
FloatDelimiter="." CSVDelimiter="Tab" CSVQuote="|" />
<Object TableName="MYSQL_FIRSTTABLE" FullSync="1">
<Fields>
```

```
<Field Name="ID" FieldType="Integer"/>
<Field Name="F_INT" FieldType="Integer"/>
<Field Name="F_STR" FieldType="String" size="20"/>
<Field Name="F_TEXT" FieldType="Blob"/>
<Field Name="F_DATE" FieldType="DATE"/>
</Fields>
<PrimaryKey>
  <Field Name="ID" />
</PrimaryKey>
<FromClause />
<FilterClause />
<ChangedRecordsSQL />
</Object>
</Publication>
```

5 Экспорт данных

Для экспорта данных необходимо в консоли вы полнить команду:

```
Redreplicator export <filename> <generation | list>  
<subscriber> [<TableName>] [bin | xml | csv]
```

- filename – имя файла
- generation|list – режим экспорта
- Subscriber – псевдоним подписчика
- TableName – таблица подписчика
- Bin|xml|csv – формат файла – банарный, xml-, или csv-файл

При экспорте файлов содержание любых blob-полей будет представлено в base-64 кодировке.

5.1 Режимы экспорта

Экспорт (публикация) данных может производиться в 2 режимах:

- generation — только те данные, которые изменились с момента последней публикации;
- list - экспорт всех данных.

Под дельтой понимается список, включающий в себя записи, которые подверглись изменению в Master-базе за некоторый период времени. Все изменения в реплицируемой базе индексируются, и им присваивается уникальный номер, называемый «поколением». Для генерации дельты необходимо указать базовое поколение – поколение предыдущей полученной публикации.

5.2 Форматы экспорта

Оба режима экспорта могут быть представлены в следующих форматах:

- двоичный файл;
- XML файл;
- CSV файл.

5.2.1 Формат двоичного файла

Формат двоичного файла имеет следующую структуру:

- Заголовок:
 - Описание (тип String)
 - Начальное поколение (тип Integer)
 - Конечное поколение (тип Integer)
- Описание таблиц³:

3 Пустое имя таблицы означает конец описания таблиц

- Имя таблицы (тип String)
- Первичный ключ (тип String)
- Описание полей:⁴
 - Имя поля (тип String)
 - Тип (SQL_SHORT, SQL_LONG, SQL_FLOAT, SQL_DOUBLE, SQL_VARYING, SQL_TEXT, SQL_BLOB, SQL_DATE, SQL_TYPE_TIME, SQL_TYPE_DATE)
 - Размер (тип Integer)
 - Scale (тип Integer)
- Данные.
 - P_DELETEFLAG (тип Byte) – маркер удаления записи
 - список значений полей - ISNULL (тип Byte) | значение

5.2.2 Формат XML файла

Формат XML файла для экспорта описывается в дескрипторе подписчика, но в целом имеет следующую структуру:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Publication Signature="RedReplicator" Type=""
  TableName="">
<Rows>
  <Row P_ACTION=""FIELD_1=""FIELD_2=""FIELD_3=""/>
  <Row P_ACTION=""FIELD_1=""FIELD_2=""FIELD_3=""/>
</Rows>
</Publication>
```

Тэг *Publication*:

- Type – тип экспорта – дельта или лист;
- TableName – имя экспортированной таблицы;

Тэг *Rows* – записи таблицы. Каждый тэг *Row* внутри данного тэга описывает запись одну запись таблицы базы данных в виде *Имя Поля = "Значение"*. Атрибут *P_ACTION* может принимать значения *merge/delete* и описывает действие - вставку либо удаление.

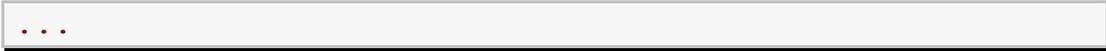
Метаданные таблицы, первичные ключи и форматы экспорта даты, времени, разделителя вещественных типов описываются в дескрипторе.

5.2.3 Формат CSV файла

Формат экспорта в csv имеет следующую структуру:

P_Action	Field_1	Field_2	Field_3 ... Field_N
----------	---------	---------	---------------------

⁴ Пустое имя поля означает конец описания полей



В первой строке указываются наименования экспортируемых полей (Field_1; ...Field_N). Затем записывается последовательность записей.

В режиме получения дельты добавляется колонка P_Action, значение элементов которой указывает на тип операции – merge или delete. В режиме получения полного списка данное поле не выгружается.

Разделитель элементов в CVS файле и обрамление полей записи, а также метаданные таблицы, первичные ключи, форматы экспорта даты и времени, указываются в дескрипторе подписчика.

6 Работа с репликатором через FTP

6.1 Структура каталогов FTP-сервера

Репликатор может принимать обновления от источников для вливания в мастер-базу и выкладывать дельта-файлы для баз-приемников с помощью встроенного FTP-сервера. Структура каталогов FTP-сервера имеет следующий вид:

```

.. \<псевдоним БД>
  \OUTBOX
    \<псевдоним подписчика>
      \delta
        \bin
        \xml
          \<TableName1>
          ...
        \csv
          \<TableName1>
          ...
      \list
        \bin-
        \xml
          \<TableName1>
          ...
        \csv
          \<TableName1>
          ...
      current-generation
  \INBOX
    \<псевдоним источника>
      \delta
        \bin
        \xml
          \<TableName1>
          ...

```

```

\csv
    \<TableName1>
    ...
\list
    \bin
    \xml
    \<TableName1>
    ...
\csv
    \<TableName1>
    ...

```

К дочерним каталогам директории OUTBOX применимы команды GET и LIST. К дочерним каталогам директории INBOX применимы команды PUT и LIST.

Каталог DELTA предназначен для публикации дельта-файлов, а LIST - для экспорта всех записей для текущего подписчика. При публикации в list режиме базовое поколение приравнивается к нулю.

В каталогах DELTA и LIST содержатся подкаталоги, соответствующие возможным форматам дельта-файлов.

Публикации режимов delta и list из каталога BIN экспортируются в виде одного бинарного файла для всех таблиц подписчика. В каталогах XML и CSV создаются файлы, соответствующие именам таблиц подписчика.

В каталогах с именами псевдонимов подписчиков находится файл current-generation, в котором указывается номер текущего поколения для соответствующей БД.

6.2 Публикация дельты через FTP

Для получения файла публикации через ftp-интерфейс необходимо следующее:

- Подключиться к FTP при помощи FTP-клиента;
- Войти в директорию, соответствующую псевдониму базы;
- Войти в директорию OUTBOX;
- Войти в директорию, соответствующую псевдониму подписчика;
- Войти в директорию, соответствующую режиму публикации (delta/list);
- Войти в директорию, соответствующую формату дельты (bin/xml/csv);
- Войти в директорию, соответствующую имени нужной таблицы;
- Выполнить команду get;
- для публикации delta:

```
get <source> <dest>  
где <source> - имя файла;  
    <dest> - имя файла для сохранения дельты
```

- Для публикации list:

```
get <dest>  
где <dest> - имя файла для сохранения дельты
```

6.3 Добавление дельты через FTP

При публикации файла через ftp-интерфейс необходимо следующее:

- Подключиться к FTP при помощи FTP-клиента
- Войти в директорию, соответствующую псевдониму базы
- Войти в директорию INBOX
- Войти в директорию, соответствующую псевдониму источника
- Войти в директорию, соответствующую режиму публикации (delta/list)
- Войти в директорию, соответствующую типу дельты (bin/xml/csv)
- Если тип xml/csv, то войти в директорию с именем нужной таблицы
- Выполнить команду

```
put <source>  
где <source> - имя дельта-файла
```

- Отключиться от FTP сервера
Публикация дельта-файла через FTP-интерфейс доступна только для источников-файлов в xml, csv или бинарном формате. Формат файла для добавления аналогичен формату файла публикации.

7 Программные интерфейсы «Ред Реплика-тор»

«Ред Репликатор» может быть использован как COM/DCOM/CORBA сервер.

Работа с программным интерфейсом репликатора реализуется через следующие методы и свойства⁵:

Свойство/метод	Описание	Параметры	
		Входные	Выходные
procedure AddSubscriber(const Code: WideString; const ConnString: WideString; Active: WordBool; const Descriptor: WideString; const PhysicalFile: WideString); override;	Добавление подписчика	<i>Псевдоним, Строка соединения:</i> WideString <i>Состояние:</i> WordBool <i>Дескриптор, Путь к файлу:</i> WideString .	-
procedure DeleteSubscriber(const Code: WideString); override;	Удаление подписчика	<i>Псевдоним:</i> WideString	-
procedure ResetSubscriber(const Code: WideString; const Descriptor: WideString; const PhysicalFile: WideString); override;	Пересоздание подписчика	<i>Псевдоним, Дескриптор, Путь к файлу:</i> WideString	-
procedure UpdateSubscriber(const Code: WideString); override;	Обновление подписчика	<i>Псевдоним:</i> WideString	-
procedure UpdateAllSubscribers; override;	Обновление всех подписчиков	-	-
procedure ValidateSubscriber(const Code: WideString; out ErrorCount: Integer; out WarningCount: Integer); override;	Проверка соответствия структуры базы подписчика и мастер-базы	<i>Псевдоним:</i> WideString <i>Количество ошибок, количество предупреждений:</i> Integer .	-
procedure MakeExport(const Filename: WideString; Generation: Integer; const Subscriber: WideString; const DeltaMode: WideString; const TableName: WideString); override;	Экспорт данных подписчика	<i>Имя файла:</i> WideString; <i>Поколение:</i> Integer; <i>Подписчик, Режим, Таблица:</i> WideString	-
procedure UpdateFromSource(const Code: WideString); override;	Обновление с указанного источника	<i>Псевдоним:</i> WideString	-
procedure	Обновление со	-	-

5 Все описанные свойства доступны как для чтения, так и для установки значений. При отсутствии указанных свойств возможно использование методов Get и Set с аналогичными параметрами.

UpdateFromAllSources; override;	ВСЕХ ИСТОЧНИКОВ		
procedure AddSource(const Code: WideString; Active: WordBool; const Descriptor: WideString); override;	Добавление источника	<i>Псевдоним:</i> WideString; <i>Состояние:</i> WordBool; <i>Дескриптор:</i> WideString	-
procedure DeleteSource(const Code: WideString); override;	Удаление источника	<i>Псевдоним:</i> WideString	-
procedure GetSubscribers(out List: WideString); override;	Получение списка подписчиков	-	<i>Список подписчиков:</i> WideString
procedure GetSources(out List: WideString); override;	Получение списка источников	-	<i>Список источников:</i> WideString
procedure GetSubscriberByCode(const Code: WideString; out Active: WordBool; out ConnString: WideString; out Descriptor: WideString; out IsFound: WordBool); override;	Получить регистрационную информацию подписчика	<i>Псевдоним:</i> WideString	<i>Состояние:</i> WordBool; <i>Строка соединения,</i> <i>Дескриптор:</i> WideString; <i>Обнаружен:</i> WordBool
procedure GetSourceByCode(const Code: WideString; out Active: WordBool; out Descriptor: WideString; out IsFound: WordBool); override;	Получить регистрационную информацию источника	<i>Псевдоним:</i> WideString	<i>Состояние:</i> WordBool; <i>Дескриптор:</i> WideString; <i>Обнаружен:</i> WordBool
procedure SetSourceDesc(const Code: WideString; const Descriptor: WideString; out Result: WordBool); override;	Установить новый дескриптор для подписчика	<i>Псевдоним,</i> <i>Дескриптор:</i> WideString	<i>Результат:</i> WordBool
procedure SetSubscriberDesc(const Code: WideString; const Descriptor: WideString; out Result: WordBool); override;	Установить новый дескриптор для источника	<i>Псевдоним,</i> <i>Дескриптор:</i> WideString	<i>Результат:</i> WordBool
procedure ExecuteScript(const Script: WideString; const Code: WideString); override;	Выполнить файл с SQL-скриптом	<i>Путь к файлу,</i> <i>Псевдоним:</i> WideString	-
property ReplicationEnabled: WordBool;	Состояние репликации	-	-
property SubscriberActive[const Code: WideString]: WordBool;	Состояние подписчика	<i>Псевдоним:</i> WideString	-
property ActiveDatabaseName: WideString;	Имя активной мастер-базы	-	-
property ActiveDatabaseRplAlias: WideString;	Псевдоним активной мастер-базы	-	-
property SourceActive[const Code: WideString]: WordBool;	Состояние источника	<i>Псевдоним:</i> WideString	-

Для использования CORBA -интерфейса для управления службой Ред Репликатор, необходимо включить поддержку CORBA в конфигурационном файле репликатора, например:

```
<ReplicatorConfig>
<CommonSettings RplServerName="" UpdateInterval="1"
  DefaultDatabase="Test">
  <CORBA Enabled="true"
    IIOPAddress="inet:127.0.0.1:10000" />
  <FTP Enabled="false" FtpCommandPort="21"
    FtpDataPort="20" />
</CommonSettings>
</ReplicatorConfig>
```

- **Enabled** — используется для активации CORBA интерфейса для управления службой репликатора (по умолчанию false).
- Свойство **IIOPAddress** — указывает адрес и порт для соединения по интерфейсу CORBA (по умолчанию inet:127.0.0.1:10000).

Для работы CORBA интерфейса необходимы библиотеки MTDORB_UCUtils.dll (или MTDORB_UCUtils.so)



Приложение А - Служебные таблицы «Ред Репликатор»

При активации репликации репликатор добавляет в базу данных ряд служебных таблиц:

- RPL\$TABLES – Список таблиц, генерируемых в базе данных для каждого источника;
- RPL\$SUBSCRIBER – Регистрационная информация подписчиков;
- RPL\$SOURCE – Регистрационная информация источников;
- RPL\$GENERATION – Поколения мастер-базы репликатора;
- R\$RPL\$SUBSCRIBER – Таблица соответствий между поколением репликатора, и подписчикам, обновленным в этом поколении.

При добавлении источника репликатор добавляет информацию о нем в таблицу RPL\$SOURCE. При обновлении всех источников, либо определенного источника, Репликатор ищет по таблице RPL\$TABLES название таблицы, соответствующей каждому источнику. Каждому источнику может соответствовать только одна таблица, но каждой таблице могут соответствовать один или более источников. В случае если соответствия не было найдено, репликатор создает в мастер-базе 2 таблицы, соответствующие описанию в дескрипторе. Первая таблица содержит поле P_SOURCE, хранящая информацию о псевдониме источника, от которого получены данные, а так же поля, указанные в дескрипторе источника. Вторая таблица является служебной, и хранит информацию о поколениях данного источника.