

Общее описание WideCoup Mediation

Система WideCoup Mediation является посредником между телекоммуникационными устройствами и системами обработки данных, реализующих процедуры тарификации, формирования счетов и т.д. Основное его назначение - преобразование данных в различных форматах из разных источников в единый формат хранения данных - DataRecord и передача их для дальнейшей обработки в системах расчета, процедурах тарификации, системе принятия решений и т.п.

Основные требования, стоящие перед любой системой сбора и предобработки данных, которым удовлетворяет WideCoup Mediation:

- Надежность;
- Возможность работы с форматами данных большинства современных телекоммуникационных устройств – будь-то АТС, маршрутизатор, точка доступа или Softswitch;
- Скорость обработки поступающей информации;
- Возможности удаленного управления и диагностики системы;
- Возможность подключения файлов сценариев или plug-in для приема данных специфичных устройств, которые выдают исходные данные в нестандартном виде;
- Возможность подключения файлов сценариев или plug-in для дополнительной обработки DataRecord перед передачей в специфические системы дальнейшей обработки, например PMS-системы, или в другое оборудование.

В связи с вышеперечисленными требованиями и учитывая большинство эксплуатируемых корпоративных серверных систем, система WideCoup Mediation была реализована в виде службы Windows 2000 и базируется на технологии .NET

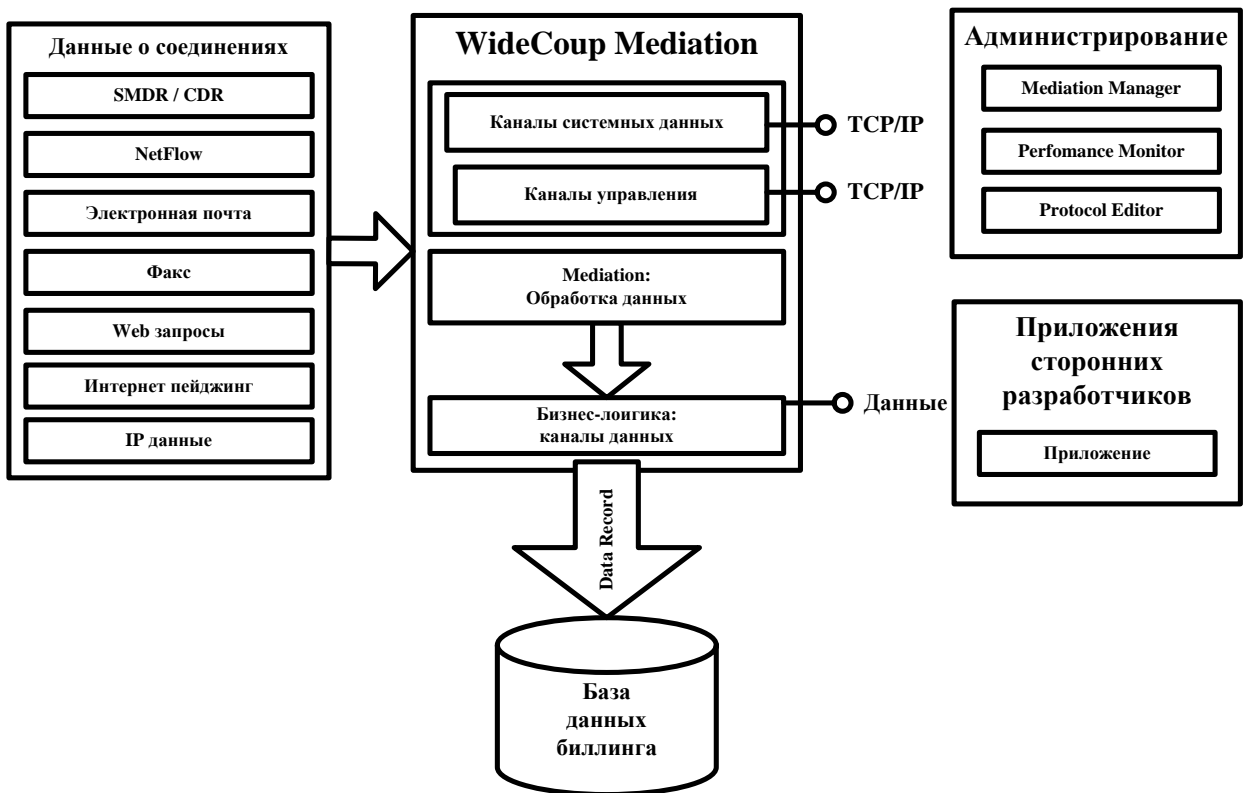


Рис. 1. Архитектура WideCoup Mediation

Основные преимущества:

- Использование Regular Expression (синтаксис совместим с PCRE - Perl Compatible Regular Expression), что позволяет разбирать практически все протоколы любых коммуникационных устройств.
- наращивание функциональности системы предобработки достигается за счет ее модульности и не требует дополнительных затрат на обновление ядра системы
- Высокая скорость разбора строк за счет прекомпилирования готовых регулярных выражений (более 1000 в секунду).
- возможность сбора данных по протоколам, описываемым внешними файлами (скриптами), дает возможность построения сложных алгоритмов подключения, запроса и проверки корректности исходных данных о звонках или услугах
- отправка предобработанных данных по TCP/IP, в RS-232 или файл в реальном режиме времени позволяет использовать систему совместно с системами управления персоналом для гостиничного бизнеса (PMS-системами), такими как Fidelio
- Отсутствие GUI позволяет полностью управлять службами WideCoup Mediation удаленно при помощи стандартного набора средств операционной системы Windows 2000 и выше.
- Использование XML стандарта для хранения всех конфигурационных файлов системы, протоколов разбора строк и подключения к внешним БД облегчает администрирование системы
- возможность обработки log-файлов на одной рабочей станции, минуя подключение к SQL серверу из регионов, существенно снижает требования системы для решения задач территориально-распределенных предприятий
- Наличие редактора протоколов (WideCoup Parser Editor) со встроенным модулем тестирования настроек при создании протоколов разбора строк.
- Поддержка статических типов данных на уровне предобработки исходных записей позволяет вести учет различных наложенных типов услуг, например видео-конференций и т.п., по отдельным алгоритмам и принципам
- Протоколирование работы как самой системы, так и процесса разбора строк.
- Использование стандартных счетчиков Perfomance Counter для диагностики работы системы.
- система оповещения по e-mail позволяет оперативно следить за состоянием системы и конфигурируется для каждого процесса (филиала) в отдельности, что дает возможность разделять полномочия по мониторингу различных частей системы

Основные функции системы

Основными функциями системы являются разбор исходных данных о соединениях или услугах и обработка полученных данных. Для этого используются принципы работы, основанные на протоколах разбора записей, логики обработки и структура результата обработки - универсальная запись Data Record.

Протокол разбора записей

Для создания нового протокола лучше использовать WideCoup Protocol Editor, хотя может быть использован любой XML-редактор. Готовый файл описания представляет собой XML файл с расширением .protocol, который непосредственно под требуемую задачу можно исправить в обычном Notepad.

Примеры готовых протокола простого формата станции Panasonic KX-TD500:

```

<Name>Входящий </Name>
- <DurationCall>
  <ReplacePattern>${hh}:${mm}:${ss} </ReplacePattern>
  <Offset>61 </Offset>
  <MatchPattern>(.)+(?<hh>[0-9]{2}):(?<mm>[0-9]{2})'(?<ss>
    [0-9]{2})(.)+ </MatchPattern>
  <durpartype>fromTimeString </durpartype>
  <Offset2>0 </Offset2>
</DurationCall>
- <InitiatorNumber>
  <ReplacePattern />
  <Offset>46 </Offset>
  <MatchPattern>\<INCOMING\> </MatchPattern>

```

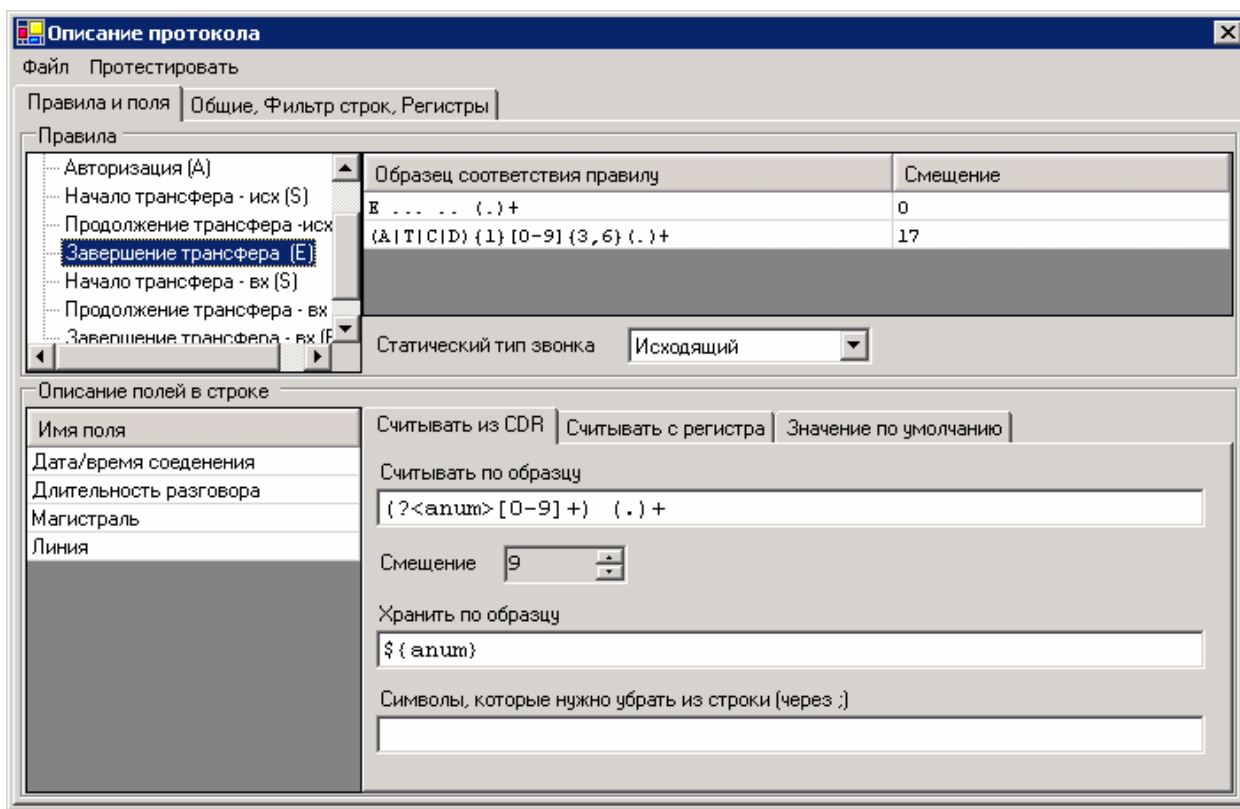


Рис. 2. Пример описания протокола разбора исходных данных

Для удобства тестирования написанного протокола, в Protocol Editor встроен модуль, который позволяет визуально увидеть процесс разбора строки взятой из локального файла-примера.

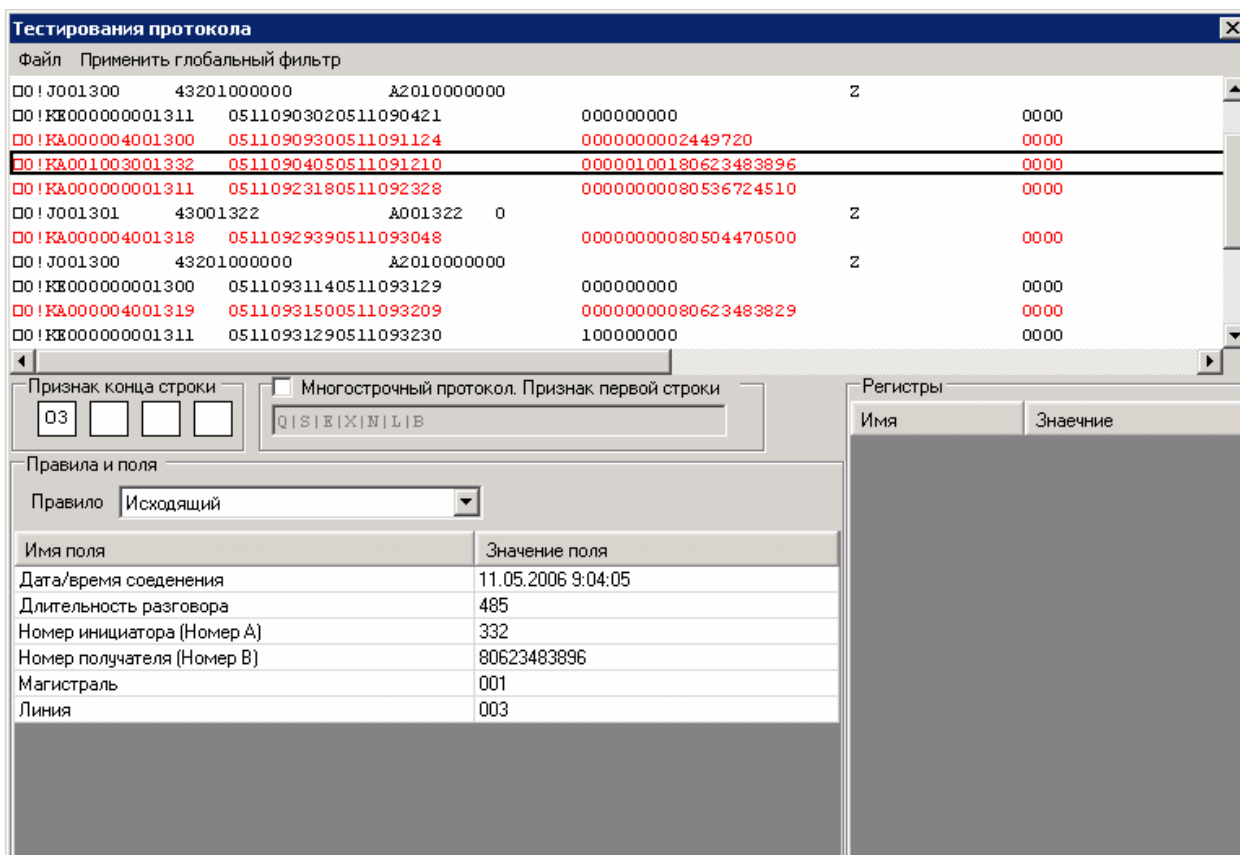


Рис. 3. Результат разбора строки исходного файла

Логика протокола Call Logic

Протокол представляет собой набор правил, которые в свою очередь содержат описания полей. На уровне протокола задается:

- название протокола и номер устройства в БД;
- признак первой строки и количество строк в записи (для многострочных протоколов, например со станции Meridian);
- глобальный фильтр попадания записей для разбора (все остальные складываются отдельно);
- регистры - специальные автоматы, существующие независимо и хранящие в себе информацию единую для нескольких строк.

Правила необходимы для распределения записей по статическим типам и для отсеивания мусора станций (дубликатов строк, трассировки и т.п.).

Отдельно фиксируется алгоритм агрегации записей, а также наполнения текущей записи результатами или историей обработки других записей. Это позволяет строить логические цепочки при обработке сложных и зависимых друг от друга структур и создавать на основе агрегации результирующие записи для минимизации кол-ва исходных данных.

Сама процедура описания полей, получаемых из строки, представляет собой написание соответствующих регулярных выражений для каждого из полей, а также ввода особых алгоритмов разбора, индивидуальных для каждого поля.

Структура DataRecord

С целью стандартизации исходных записей был разработан единый протокол представления обработанных данных - DataRecord. В нем хранятся результаты обработки исходных данных – либо в БД, либо во внешних системах. Относительно этой структуры происходит процесс сериализации или десериализации объектов WideCoup Mediation.

Данные, хранимые в DataRecord:

- **cid** - уникальный идентификатор услуги
- **NumberAts** - уникальный идентификатор устройства коммутации
- **pkt_ID** - идентификатор пакета услуги. Почти всегда создается автоматически. При загрузке из файлов поддерживается назначение фиксированных пакетов, чтобы потом облегчить их обработку.
- **TariffMode** - режим тарификации. К примеру: тарифицировать (0) или перетарифицировать (1)
- **DurationRing** - длительность инициализации услуги в секундах. К примеру, для звонков через АТС - время дозвона
- **DateTimeStartCall** - дата/время начала предоставления услуги. К примеру, для звонков через АТС - точка во времени, когда начался разговор
- **DurationCall** - длительность предоставления услуги или количество фиксированных единиц объема услуги, предоставленных по данной записи. К примеру, для звонков через АТС - время разговора. Для конвергентного биллинга - количество единиц объема услуги, например факт перевода 24 страниц текста на английский язык
- **InitiatorNumber** - уникальное значение иницилирующего вызов пользователя. К примеру, для звонков через АТС: для внутреннего и исходящего типов - внутренний номер, для входящего и транзита – АОН и т.д.
- **RecipientNumber** - уникальное значение пользователя-получателя услуги. К примеру, для звонков через АТС: для внутреннего типа - набранный внутренний номер, для исходящего типа - набранный номер из сети ТСОП, для входящего типа - внутренний номер, для транзита - набранный исходящий номер и т.д.
- **Route** - уникальное значение группы каналов устройства коммутации использованной. К примеру, для звонков через АТС: для услуги исходящего или транзитного типа – магистраль.
- **Trunk** - уникальное значение канала устройства коммутации. К примеру, для звонков через АТС: для услуги исходящего или транзитного типа - физическая линия в магистрали.

- **Trunk2** - уникальное значение канала устройства коммутации, используется аналогично Trunk, но фиксируемого дважды – например, для транзитного типа звонка. К примеру, для звонков через АТС - физическая линия в магистрале, через которую выходит транзитный звонок.
- **InitiatorAbonentName** - уникальное дополнительное значение пользователя-инициатора услуги. Для АТС – некоторые АТС и маршрутизаторы могут сбрасывать ФИО абонента из корпоративных справочников
- **RecipientAbonentName** - уникальное дополнительное значение пользователя-получателя услуги. Для АТС - некоторые АТС и маршрутизаторы могут сбрасывать ФИО абонента
- **StaticType** - уникальный идентификатор статического типа услуги. К примеру, для звонков через АТС - входящий, исходящий, внутренний, транзитный
- **DynamicType** - уникальный идентификатор динамического подтипа услуги. К примеру, для звонков через АТС - город, межгород и т.п. Если нет задачи определять его на этапе разбора, то будет определен в процессе тарификации по набранному номеру
- **Cost** - стоимость услуги, если она уже известна. К примеру, для звонков через АТС - когда мы получаем уже протарифицированные данные (например из счета оператора моб. связи)
- **AuthCode** - уникальное значение персонального кода пользователя-инициатора, использованного для получения услуги. К примеру, для звонков через АТС - код авторизации
- **SourceString** - сериализация самого себя

Поскольку система является конвергентной, но назначение тех или иных полей структуры может быть изменено. Например, для услуг IP телефонии в отличие от обычной телефонии ряд полей представляют иной смысл:

- **DurationCall** - объем данных в байтах
- **InitiatorNumber и RecipientNumber** - IP адреса источника и назначения
- **AuthCode** - логин абонента, который инициировал IP сессию
- **Trunk и Trunk2** - направление передачи данных (IP шлюзы)
- **DynamicType** - тип трафика

Запуск WideCoup Mediation

Установочный пакет WideCoup Mediation представляет собой инсталляционный файл MSI. После установки в установочном каталоге будет создана следующая структура:

- **CDRGarbage** - хранилище исходных данных, не прошедших обработку
- **CDRSources** - набор Plug-in для наиболее популярных источников (COM порт, TCP/IP, папка на локальном диске) или специфических, например для Starex CS-1000
- **CDRStorage** - хранилище всех получаемых источником данных в исходном виде
- **CDRTemp** - хранилище исходных данных, которые по каким либо причинам не были разобраны
- **DataProcessorTask** - набор Plug-in для приемников DataRecord (например, SQL Sender Unibill для запуска процедур тарификации WideCoup Billing или PMS RS-323 для Fidelio)
- **Protocols** - набор протоколов разбора для различных устройств

Простота запуска и использования WideCoup Mediation позволяет применять данное решение в компаниях, не имеющих высококвалифицированных специалистов. Для запуска WideCoup Mediation необходимо всего лишь ввести команду

```
net start Mediation
```

Для мониторинга работы WideCoup Mediation достаточно после её запуска открыть Event Viewer сервера и в разделе Mediation увидеть похожую картину:

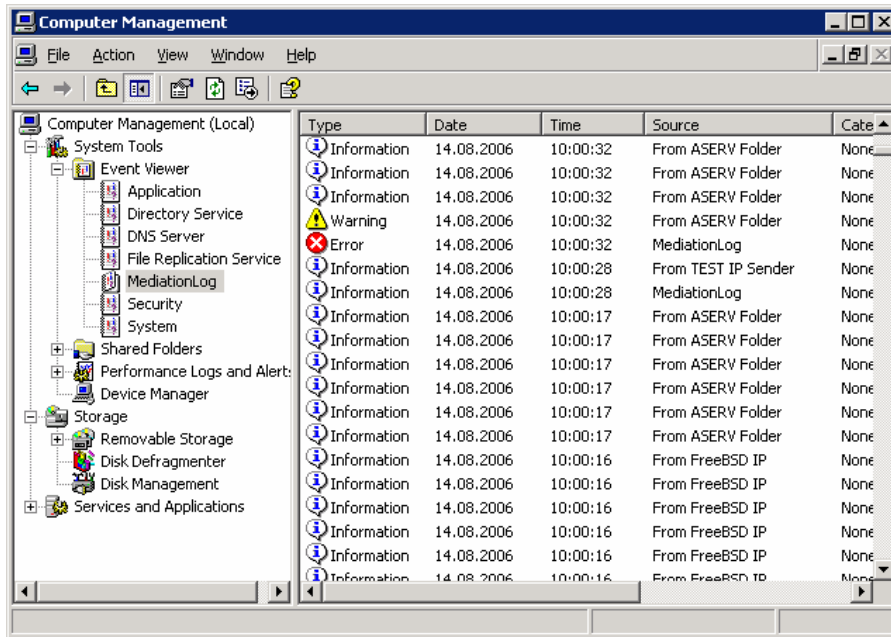


Рис. 4. Сообщения для диагностики запуска и работы Mediation

Если нет ошибок, помеченных красной иконкой, значит система запущена и работает нормально.

Особенное удовольствие доставляет мониторинг процессов обработки исходных данных – достаточно запустить Performance Monitor из панели управления Windows:

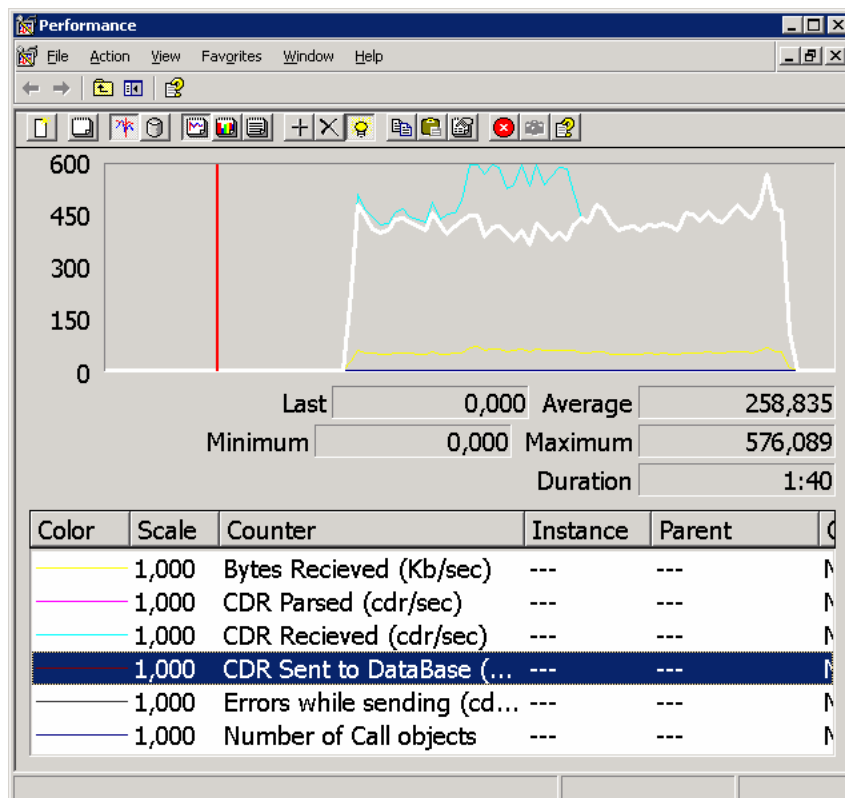


Рис. 5. Счетчики WideCoup Mediation в Performance Monitor

Перечень счетчиков доступных для мониторинга WideCoup Mediation:

- **CDR Sent to DataBase (cdr/sec)** - кол-во посланных записей DataRecord в БД или системы
- **CDR Parsed (cdr/sec)** - кол-во разобранных данных
- **CDR Recieved (cdr/sec)** - кол-во полученных записей для разбора
- **Bytes Recieved (Kb/sec)** - объем полученных данных из источника, КБ
- **Errors while sending (cdr/sec)** - кол-во данных, во время пересылки которых произошла ошибка