

Автоматизированная система
**Внутриведомственная интеграционная шина министерства здравоохранения
Самарской области» (этап 2012 года)**
Протокол обмена ЭМК между МИС

Самара 2012

Содержание

1.Правила подготовки пакета для информационного взаимодействия медицинских информационных систем лечебно-профилактических учреждений с региональным уровнем единой государственной информационной системы здравоохранения	4
1.1.Общие положения	4
1.2.Инструкция вставки.....	6
1.3.Инструкция обновления	6
2.Правила обработки и трансформации данных пакета при информационном взаимодействии медицинских информационных систем лечебно-профилактических учреждений с региональным уровнем единой государственной информационной системы здравоохранения	7
2.1.Общие положения	7
2.2. Создание утилит	8
2.3. Механизм работы СУОД в режиме «Сервер ЦОД»	13
2.4. Алгоритм импорт пакетов на стороне клиента.....	15
2.5. Алгоритм экспорт пакетов на стороне клиента	17
2.6. Алгоритм импорта пакета у сервера (для чайников)	19
2.7. Алгоритм экспорта пакета от сервера (для чайников).....	20
2.8.Критические допущения и запреты	22

Перечень сокращений

АРМ	–	автоматизированное рабочее место специалиста медицинской организации: персональный компьютер и прикладное программное обеспечение, автоматизирующее отдельные функции в деятельности специалиста
АС	–	автоматизированная система
БД МИС	–	база данных медицинского учреждения, которая отражает ресурсы, результаты деятельности учреждения, в том числе результаты оказанных населению Самарской области медицинских услуг
ЕГИСЗ	–	Единая Государственная информационная система в сфере здравоохранения
ИИР	–	интегрированный информационный ресурс регионального фрагмента единой государственной информационной системы в сфере здравоохранения Самарской области
ЛВС	–	локальная вычислительная сеть
ЛПУ	–	лечебно-профилактическое учреждение Самарской области
МИАЦ	–	государственное учреждение здравоохранения «Самарский областной информационно-аналитический центр»
МИС	–	медицинская информационная система
МО	–	медицинские организации - учреждения здравоохранения, медицинская организация, орган исполнительной власти и органы местного самоуправления, осуществляющие деятельность по оказанию государственных и муниципальных услуг в сфере здравоохранения, аптечная и фармацевтическая организации, а также лечебно-профилактические учреждения
ОС	–	операционная система
РФ ЕГИСЗ, Система	–	Региональный фрагмент Единой государственной информационной системы в сфере здравоохранения Самарской области
СУОД	–	система управления обработкой данных
ЦОД	–	центр обработки данных
ЭМК	–	электронная медицинская карта

1. Правила подготовки пакета для информационного взаимодействия медицинских информационных систем лечебно-профилактических учреждений с региональным уровнем единой государственной информационной системы здравоохранения

1.1. Общие положения

Пакет данных это заархивированный файл «data.sql», содержащий набор инструкций для вставки (insert) и обновления (update) данных.

Состав и форматы передаваемых данных для работы сервиса СУОД определяется участниками информационного взаимодействия в соответствии с правовым основанием для информационного обмена.

Описание состава пакета составляется и изменяется только администраторами ЦОД.

Каждая инструкция в пакете вставляет или обновляет только одну запись БД МИС или ИИР.

Имя пакета формируется в соответствии с правилами, приведенными в Регламенте информационного взаимодействия МИС ЛПУ с РФ ЕГИСЗ.

Пример имени пакета: «101002001SLIST000000807.C7R90» – пакет от клиента «101002001», с маской «SLIST», имеет номер «000000807». Время создания пакета - «C7R90»

Пример «data.sql»:

```
SET NOCOUNT ON;  
BEGIN TRY  
BEGIN TRANSACTION;
```

```
INSERT POLD_FIO ( STATE, COUNTER0  
    , LPUIN, D_START, D_MODIF, D_FIN, OPERATOR, SURNAME,  
    NAME, SECNAME, SEX, BIRTHDAY, XKEY, FPERSON, MKEY, LPUOUT)  
    values ( 1 , 38154, 101002001, '2012-07-27 08:46:00', '2012-07-27  
08:46:00', NULL, 'Петров П.П', 'Иванов', 'Иван', 'Иванович', 2, '2012-06-23  
00:00:00', NULL, NULL, '38154', 101002001);
```

```
update POLD_FIO set LPUOUT = 101002001, D_MODIF = '2012-07-27
08:22:00', OPERATOR = 'Петров П.П.', SURNAME = 'Сидоров', STATE = 0
WHERE COUNTER0 = 23430 and LPUIN = 101002001
```

```
update POLD_FIO set LPUOUT = 101002001, D_MODIF = '2012-07-27
08:05:00', OPERATOR = 'Петров П.П.', SURNAME = 'Канарейкин', STATE =
0 WHERE COUNTER0 = 25791 and LPUIN = 101002001
```

```
COMMIT TRANSACTION;
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION
declare @S varchar(4000) set @S = ERROR_MESSAGE() + char(13) +
Строка ошибки: ' + cast(ERROR_LINE() as varchar(99))
RAISERROR (@S, 18, 10);
END CATCH
SET NOCOUNT OFF;
```

Где шапка скрипта:

```
SET NOCOUNT ON;
BEGIN TRY
BEGIN TRANSACTION;
```

Окончание скрипта:

```
COMMIT TRANSACTION;
END TRY
BEGIN CATCH
ROLLBACK TRANSACTION
declare @S varchar(4000) set @S = ERROR_MESSAGE() + char(13) +
Строка ошибки: ' + cast(ERROR_LINE() as varchar(99))
RAISERROR (@S, 18, 10);
END CATCH
SET NOCOUNT OFF;
```

Шапка и окончание скрипта одинаковые для каждого пакета.

1.2. Инструкция вставки

Инструкция вставки должна содержать обязательные поля
STATE, COUNTER0, LPUIN, D_START, D_MODIF, D_FIN, LPUOUT

Где STATE – всегда принимает значение 1; COUNTER0 – код записи (значение автоинкрементного поля); LPUIN – код клиента; LPUOUT – код клиента, приславшего пакет; D_START – дата создания записи (технологическая дата); D_MODIF – дата последнего изменения записи (технологическая дата); D_FIN – дата удаления записи (технологическая дата).

1.3. Инструкция обновления

Инструкция обновления должна содержать обязательные поля
STATE, D_MODIF, LPUOUT

Где STATE – всегда принимает значение 0; LPUOUT – код клиента приславшего пакет; D_MODIF – дата последнего изменения записи (технологическая дата).

COUNTER0, LPUIN – первичный ключ записи (уникальный в рамках Самарской области), обновление записей происходит по первичному ключу

2. Правила обработки и трансформации данных пакета при информационном взаимодействии медицинских информационных систем лечебно-профилактических учреждений с региональным уровнем единой государственной информационной системы здравоохранения

2.1. Общие положения

Для того чтобы включить в информационный обмен таблицу БД необходимо создать для неё триггеры и утилиты.

Триггеры отвечают за фиксацию изменений данных в таблице БД.

Утилиты необходимы для работы ПО сервиса СУОД. Для работы с каждой таблицей ПО СУОД необходимо создать две утилиты:

Первая утилита отвечает за формирование инструкций вставки записей,

Вторая отвечает за формирование инструкций обновления записей.

Каждая утилита состоит из 2-х скриптов: шапки скрипта и запроса.

Шапка скрипта в утилите на вставку записи содержит:

**«INSERT INTO <название таблицы> ([<Имя столбца>, ...])
VALUES ()»**

Запрос при выполнении возвращает результат в виде строк для блока VALUES в шапке. Если присоединить шапку скрипта к каждой строке результата, получим набор готовых инструкций вставки данных.

Шапка скрипта в утилите на обновление данных в записи содержит:

«UPDATE < название таблицы > SET»

Запрос возвращает набор строк, при присоединении которых к шапке скрипта получают готовые инструкции на обновление.

Все эти утилиты хранятся в БД СУОД в таблице «SUOD_JRULE_Z_D» в 5-х полях:

№ п.п.	Поле	Комментарии
1		имя скрипта
2		имя базы данных, в которой находится нужная таблица
3	TABLE_NAME	имя таблицы, которую включили в обмен и к которой применимы данные утилиты
4	SCRIPT_INS_PREF	содержит шапку скрипта утилиты на вставку
5	SCRIPT_INS_CMD	содержит запрос утилиты на вставку
6	SCRIPT_UPD_PREF	содержит шапку скрипта утилиты на обновление данных
7	SCRIPT_UPD_CMD	содержит запрос утилиты на обновление данных

Создание создания утилит и триггеров

Для создания утилит и триггеров необходимо заполнить следующие поля:

В поле «Имя базы данных» указываем имя БД, в которой находится нужная таблица;

В поле «Название таблицы» указываем имя таблицы.

Для создания триггера на заданную таблицу необходимо «Сгенерировать скрипт триггера», если поле «Имя скрипта» пустое, то откроется меню сохранения скрипта, если путь уже указан то следующие скрипты будут сохраняться по указанному пути, его всегда можно изменить в поле «Имя скрипта» (путь используется только при создании триггеров). При создании триггера галочка «Для клиента» не анализируется – т.к. триггеры и на клиенте и на сервере одинаковые.

Сохранённый скрипт будет иметь следующее имя: Add_ + «Имя БД» + _ + «Имя таблицы» + _ToSUOD.sql.

2.2. Создание утилит

При создании утилит необходимо анализировать состояние галочки «Для клиента», если она стоит то утилиты будут созданы для клиента, если не стоит, то утилиты будут созданы для сервера.

Для создания утилит необходимо нажать на кнопку «Сгенерировать utils»

Утилиты серверной и клиентской СУОД различаются, т.к. различный механизм выгрузки данных с сервера и с клиента (на клиенте более простой, требующий меньше нагрузки на сервер).

Формируются 5 файлов:

1. «**_inspref.sql**» и «**_ins.sql**» утилиты формирующие скрипт вставки записи;

2. «**_updpref.sql**» и «**_upd.sql**» - утилиты формирующие скрипт обновления записи;

3. «**_sql.sql**» - содержит скрипт который вставляет в СУОД сгенерированные утилиты.

«Для клиента», утилиты формируются и сохраняются в каталоге «d:\MIAC\PROGRAMS\SUOD\add-SUOD\имя БД\CLIENT\имя таблицы»;

Утилиты для сервера формируются и сохраняются в каталоге «d:\MIAC\PROGRAMS\SUOD\add-SUOD\имя БД\SERVER\имя таблицы».

Системы ведения данных (СВД) вставляют или обновляют записи в нужных таблицах БД. Для вставленных записей устанавливается значение поля STATE = 1, а для обновленных – STATE устанавливается в соответствии с обновляемыми полями.

Система трансформации данных (СТД) (в составе сервиса СУОД) обеспечивает синхронизацию (вставку или обновление записей) между «родной» БД в формате MS-SQL и «внешними» БД различных форматов и структур.

СТД определяет необходимость трансформации и наличие записей для синхронизации в соответствии с Журналом трансформации (ЖТ). Параметры ЖТ указаны в таблице 1.

Для всех записей, вставленных с помощью СВД или СТД необходимо указывать LPUIN – уникальный код клиента СУОД из настроечного файла ПО.CFG, переданного из ЦОД.

Настроечный файл ПО.CFG размещается в директориях каждой из СВД под именем СВД.CFG и директории Серверной СУОД SUOD.CFG. Таким образом, ПО.CFG будут отличаться для различных СВД.

Под одним идентификационным кодом LPUIN может работать несколько клиентов только в том случае, если с данным кодом не будет происходить информационный обмен через сервис СУОД.

Конкатенация полей *COUNTER0* и *LPUIN* обеспечивает уникальность записей во всем информационном пространстве обмена данными средствами сервиса СУОД.

Поле *COUNTER0* принимает значение автоинкрементного поля клиентской БД *COUNTER* при вставке записи путем скопировав его из. Необходимость в таком копировании объясняется тем, что при передаче записи от клиента серверу ЦОД, *COUNTER* получит новое значение, а *COUNTER0* – так и останется клиентским. *При переходе на режим работы онлайн всех участников* информационного взаимодействия потребуются удалить поля *COUNTER* и *STATE*, поле *COUNTER0* переопределить как автоинкрементное, а также удалить триггеры, устанавливающие *STATE* и отключить СУОД.

Настроечный файл, кроме «своего» уникального LPUIN, должен содержать MAC-адрес клиентского сервера, на котором установлен СУОД, признак *ACCESS_MODE* - *режим работы* (1-через СУОД; 2-онлайн), а также список «дружественных» LPUIN, чьи записи данный клиент может редактировать. Редактирование записей с «дружественным» LPUIN

осуществляется без изменения значения STATE, но с автоматическим формированием *письма-извещения* о необходимости исправления записи от данного учреждения учреждению, создавшему запись, а также сохранением данного письма в Журнале оповещений (ЖО), описанном в таблице 2. Данное письмо может отображаться пользователю, исправившему запись или сообщаться о том, что такое письмо будет отправлено учреждению-владельцу (опционально)

Таким образом, СУОД перед запуском процедуры экспорта должна, при ACCESS_MODE = 1, проверить соответствие MAC-адреса сервера MAC-адресу из настроенного файла, проидентифицировав клиента информационного пространства. После этого, должен быть выполнен экспорт новых и измененных записей со «своим» LPUIN, а также экспорт писем-извещений о необходимости изменения записей «дружественным» LPUIN. Редактирование записей с «дружественным» LPUIN при ACCESS_MODE = 2 запрещено.

Каждая СВД должна иметь свой список «дружественных» LPUIN. СВД должна ограничивать перечень записей, доступных для изменения, «своими» и «дружественными» записями.

Для расширения или изменения списка «дружественных» LPUIN клиентское учреждение должно оформить официальную заявку на изменение списка для конкретной СВД, после чего для перечня соответствующих учреждений в ЦОД будут сформированы новые настроечные файлы, которые могут быть автоматически переданы средствами Системы обновления ПО (СОП).

СУОД (сервис) при запуске создает массив задач экспорта с интервалами времен обмена для каждой из задач экспорта на основе Журнала задач экспорта (ЖЗ), а также параметрами экспорта (для того, чтобы не обращаться к БД каждый интервал обмена). Например, задача А должна экспортировать данные из таблиц X и Y с заданными условиями (также из

ЖЗ) в каждый второй интервал обмена, а задачи Б и В – из таблиц Q и Z, соответственно, каждый интервал. Параметры ЖЗ указаны в таблице 3. Также при запуске сервиса СУОД должен быть создан аналогичный массив трансформаций на основе ЖТ.

СУОД каждый *кратный* (минимальный, общий для всех задач СУОД) интервал времени обмена читает массив трансформаций проверяет, какие задачи трансформации должны быть выполнены в этот интервал времени и выполняет их. Затем СУОД читает массив задач экспорта, проверяет, какие задачи *экспорта* должны быть выполнены в этот интервал времени и выполняет экспорт. Экспорт записей должен быть основан на принципе STATE \diamond 0 и LPUIN = «свой» или «дружественный» – запись экспортируется. Если существуют записи с взведённым STATE и LPUIN \diamond «свой» и \diamond «дружественный» - то на эти записи формируется пакет оповещений (информационный пакет с суффиксом I), он содержит изменения записей, которые определяются по значению поля STATE. В пакете данных для ЦОД STATE должен иметь значение 1 для вставляемых записей и 0 для обновляемых записей. После выполнения задачи трансформации или экспорта, СУОД устанавливает следующий интервал трансформации/экспорта – признак, сколько интервалов СУОД должен пропустить перед тем, как выполнить трансформацию/экспорт данных заданной задачи. Кроме того, во время экспорта, СУОД должен увеличить на единицу номер последнего успешно экспортированного пакета. На сервере ЦОД импорт пакетов обеспечивается с учетом последовательного получения клиентских пакетов. Для каждой задачи нумерация пакетов должна быть уникальна.

Импорт данных клиентский СУОД выполняет с единственной проверкой последовательности номеров пакетов, полученных от ЦОД, с обязательной регистрацией в Журнале пакетов (ЖП). Параметры ЖП указаны в таблице 4. В пакете данных от ЦОД STATE должен иметь

значение 0 для вставляемых записей и -1 для обновляемых записей, т.е. изменять STATE при импорте не требуется.

Выполнение служебных пакетов, позволяющих устранить проблемы работы СУОД, а также обновить ПО СУОД и СВД производится с регистрацией в ЖП. Ошибки, полученные при обработке пакетов, фиксируются в ЖП, а также в логах и выполняется отправка пакета ошибки (ERR) в ЦОД. При корректной обработке пакета выполняется отправка пакета-ответа (RSP) в ЦОД.

Таким образом, *операции, выполняемые клиентским СУОД:*

- обработка служебных пакетов, позволяющих установить новое ПО, устранить проблемы СУОД, обновить ПО СУОД и СВД;
- трансформация данных в режимах импорта из внешних БД / экспорт во внешние БД;
- экспорт данных («своих» новых и обновленных записей);
- импорт данных (безусловный импорт пакетов по порядку);
- формирование пакетов-ответов и пакетов ошибок;
- формирование пакетов, содержащих письма-оповещения;
- получение пакетов от ЦОД с FTP (служебные пакеты, пакеты данных), отправка пакетов (пакеты данных, пакеты ответа, пакеты ошибок, пакеты оповещений).

Структуры таблиц Журнал трансформации, Журнал оповещений, Журнал задач экспорта, Журнал пакетов приведена в документе «Регламент информационного взаимодействия медицинских информационных систем ЛПУ с региональным уровнем Единой государственной информационной системы здравоохранения».

2.3. Механизм работы СУОД в режиме «Сервер ЦОД»

СУОД активизируется каждый кратный интервал, проверяет наличие пакетов во входящей директории. При наличии пакетов СУОД выполняет:

Тип пакета	Действия СУОД ЦОД
Пакет данных	Импорт пакета, в соответствии с Журналом разрешений импорта (ЖР)
Пакет ответа	Устанавливает статус отправленного пакета в «Обработан»
Пакет ошибок	Устанавливает статус отправленного пакета в «Ошибка» и извещает администратора ЦОД
Пакет оповещений	Импорт оповещений в ЖО и перенаправление пакета адресату

Импорт пакета данных сопровождается установкой STATE на основе триггерных команд для обновляемых записей и на основе данных самого пакета – для вставляемых записей, с обязательной проверкой вставки/обновления записей с LPUIN, полученными от этого же LPUIN.

Трансформация данных в ЦОД осуществляется от имени клиента, не передающего данные в ЦОД. Таким образом, код клиента должен быть задан в ЖТ и указываться в соответствующем поле вставляемой/обновляемой СТД записи. STATE следует задавать равным 1 при вставке записи и в соответствии с изменяемыми полями – при обновлении.

Вставка/обновление данных онлайн осуществляется средствами удаленной СВД непосредственно в БД ЦОД. При вставке STATE=1, при корректировке >1. Корректировка «чужих» или «дружественных» записей запрещена, создание разрешено только со «своим» LPUIN. Данные запреты должны быть реализованы в СВД.

Экспорт данных из ЦОД должен осуществляться адресно. СУОД вначале экспорта должен перечитать массив Журнал задач экспорта (ЖЗЭ). Если для данной задачи интервал обмена 0, перечитываем массив Журнал клиентов-издателей (ЖКИ) пакета этого типа (на веру берем тот факт, что издателя не может быть без подписчика(ов)). По каждому издателю формируем (экспортируем) скрипты данных, архивируем (с паролем и скрытием содержимого архива), перечитываем Журнал клиентов-подписчиков (ЖКП) на пакеты данных и в соответствии с клиентскими номерами предыдущих экспортированных пакетов копируем в клиентские

каталоги полученный пакет с соответствующими именами файлов (учитывая индивидуальные номера пакетов). ЖКИ, ЖКП и ЖП загружаются при запуске сервиса СУОД и описаны в таблицах 5, 6 и 7, соответственно. Таким образом, получив в цикле код клиента и определив, что на его данные есть подписчики, следует сформировать пакет данных от имени клиента-владельца записей и разослать всем подписчикам. Может быть ситуация, при которой подписчиком будет сам клиент, от чьего имени создана/обновлена запись. Например, в электронную регистратуру записываются интернет-пользователь, в LPUIN указывается код клиента-учреждения, в которое осуществляется запись, поэтому именно этому клиенту и следует передавать данные. Завершаться экспорт должен установлением STATE=0. В избежание изменения STATE онлайн-пользователем во время экспорта данных следует предварять экспорт копированием полей STATE, COUNTER в Журнал экспорта (ЖЭ), после чего сбрасывать его в ноль и работать с соединенными записями с журналом экспорта, а в случае проблем экспорта – возвращать STATE у экспортируемых записей в предыдущее значение командой OR.

Структуры таблиц Журнал клиентов-издателей, Журнал клиентов-подписчиков, Журнал задач экспорта приведена в документе «Регламент информационного взаимодействия медицинских информационных систем ЛПУ с региональным уровнем Единой государственной информационной системы здравоохранения».

2.4. Алгоритм импорт пакетов на стороне клиента

1. Отсортировали файлы-пакеты во входной директории по дате и времени формирования пакетов (т.е. по расширению пакетов)
2. Берем последовательно каждый из пакетов (П), разбираем имя пакета на код_клиента (9), маску_пакета (5), номер_пакета (9)
3. Открываем массив «Журнала задач» (ЖЗ)

4. Находим в ЖЗ задачу по маске пакета (ЖЗ.PACK_MASK = П.маска_пакета)

5.Проверяем активность задачи (ЖЗ.ACTIVE=1 и ЖЗ.IMPORT_ERROR_CODE=0) и получаем код задачи (ЖЗ.PACK_CODE) и номер последнего импортированного пакета с данной маской (ЖЗ.PACK_IMPORT_LAST_NUM)

5.1. Если задача активна, то делаем запись в Журнал пакетов (ЖП: COUNTER, LPUIN, PACK_CODE, OPERDATE, PACKETNUM, PACKNAME, OPERTYPE, OPERSTAT, PACKET, SYS_DATE)

5.2. Проверяем последовательность получения пакетов от сервера, таким образом, чтобы номер в имени пакета был на единицу больше последнего нами импортированного пакета (П.номер_пакета = ЖЗ.PACK_IMPORT_LAST_NUM+1). При несоответствии переходим к п. 6

5.3. Разархивируем пакет с проверками имени вложенного скрипта и т.п., в случае ошибок переходим к п.6

5.4. Подключаемся к БД, описанной в ЖЗ (SERVER, BASE, LOGIN, PASSWORD, AuthorizationOS) и выполняем скрипт. В случае выявления ошибок переходим к п. 6

5.5. В случае успеха, формируем RSP-пакет (пустой файл с признаком успешной обработки первичного пакета от ЦОД), выкладываем его в выходную директорию (для сервера) и обновляем ЖП (OPERSTAT = успешно обработан)

6. Делаем запись в Журнал ошибок (ЖО) (COUNTER, JP_KEY [ссылка на ЖП.COUNTER], ERR_MESSAGE, JOURNAL_TYPE) и формируем RSP-пакет (архивный файл с признаком ошибочной обработки в имени и содержащим текстовый файл ошибки [из ЖО.ERR_MESSAGE]). Выкладываем в выходную директорию. Стопорим импорт пакетов с этой маской (ЖЗ.IMPORT_ERROR_CODE := ЖО.COUNTER) и обновляем запись в ЖП (OPERSTAT := ошибка)

Таким образом, клиентский импорт задействует таблицы ЖЗ, ЖП, ЖО.

2.5. Алгоритм экспорт пакетов на стороне клиента

1. Срабатывает таймер.
2. Открываем массив ЖЗ.
3. Проверяем каждую активную задачу (ЖЗ.ACTIVE=1 и ЖЗ.EXPORT_ERROR_CODE=0) на предмет необходимости немедленного выполнения экспорта (локальная программная переменная массива ЖЗ CURRENT_PACK_EXPORT_DELAY=1).
4. Если CURRENT_PACK_EXPORT_DELAY > 1 уменьшаем на единицу и переходим к следующей задачи массива.
5. Если CURRENT_PACK_EXPORT_DELAY = 1, то запускаем экспорт.
 - 5.1. Подключаемся к БД с параметрами ЖЗ.SERVER, ЖЗ.BASE, ЖЗ.LOGIN, ЖЗ.PASSWORD, ЖЗ.AuthorizationOS.
 - 5.2. Задаем CURRENT_PACK_EXPORT_DELAY := ЖЗ.PACK_EXPORT_DELAY/
 - 5.3. Проверяем каждую активную подзадачу массива Журнала подзадач (ЖпЗ) (ЖпЗ.ACTIVE=1) для текущей задачи на предмет необходимости немедленного выполнения экспорта этой подзадачи (локальная программная переменная массива ЖпЗ.CURRENT_EXPORT_DELAY=1).
 - 5.4. Если ЖпЗ.CURRENT_EXPORT_DELAY>1 уменьшаем на единицу. Проверяем, если ЖпЗ.CURRENT_EXPORT_DELAY < ЖЗ.CURRENT_PACK_EXPORT_DELAY, то задаем ЖпЗ.CURRENT_EXPORT_DELAY := ЖпЗ.CURRENT_EXPORT_DELAY и переходим к следующей подзадаче.

5.5. Если ЖпЗ.CURRENT_EXPORT_DELAY=1, то иницилируем экспорт данной подзадачи (каждая подзадача отвечает за экспорт отдельной таблицы (Т)). Экспорт должен начаться с копирования всех принадлежащих этому клиенту строк таблицы Т (Т.LPUIN=номеру_клиента, производящего экспорт) в Журнал откатов (ЖОт). Копироваться должны столбцы COUNTER, STATE (STATE2) и ЖпЗ.код_таблицы Т. Копирование осуществляется путем выполнения скрипта, который выбирает из Т подзадачи все записи у которых Т.LPUIN=номеру_клиента производящего экспорт, STATE > 0 или STATE2 > 0 и записывает их COUNTER, STATE (STATE2) с предварительным именем формируемого пакета в ЖОт. В случае ошибки осуществляется откат значений полей STATE и/или STATE2 на основе данных ЖОт.

5.6. Создаем текстовый файл-скрипт (Ф), в который будут записаны SQL-инструкции по вставке/обновлению клиентских данных на сервере. Файл Ф состоит из шапки (блок Try и старт транзакции), тела – инструкций INSERT и UPDATE по каждой подзадаче поочередно и блока завершения (в т.ч. отката транзакции и обработки возможных ошибок).

5.7. Экспорт каждой подзадачи начинается с перебора всех записей таблицы подзадачи, у которых STATE=1(используются значения из ЖОт). Для каждой записи, выполняется скрипт, который формирует инструкцию INSERT на добавление этой записи. Затем для записей, у которых STATE>1 или STATE2>0 выполняется скрипт, который формирует инструкцию UPDATE на обновление записи. Получаемые этими скриптами данные составляют тело файл Ф.

5.8. После того как файл экспорта полностью сформирован (экспорт завершён для всех активных подзадач), архивируем его с паролем в архив и задаём этому архиву имя по следующей маске: LPUIN(9) + PACK_MASK(5) + PACK_EXPORT_LAST_NUM(9) + . + YMDhm(5) .

- в случае ошибки, делаем запись в ЖО (COUNTER, JP_KEY, ERR_MESSAGE, JOURNAL_TYPE) и стопорим экспорт и обновляем ЖП.OPERSTAT=ошибка.

- в случае успеха, выкладываем архив в выходную директорию (для сервера) и обновляем ЖП (OPERSTAT = успешно сформирован)

Таким образом, клиентский экспорт задействует таблицы ЖЗ, ЖпЗ, ЖП, ЖО, ЖОт.

2.6. Алгоритм импорта пакета у сервера (для чайников)

1.Получили пакет.

2.Открываем массив «Журнала задач» (ЖЗ).

3.Находим в ЖЗ задачу по маске пакета (PACK_MASK) из имени пришедшего пакета.

4.Проверяем активность задачи (ACTIVE = 1 и IMPORT_ERROR_CODE = 0) и получаем код задачи (PACK_CODE)

5.Если задача активна, то делаем запись в Журнал пакетов (ЖП) (COUNTER, LPUIN, PACK_CODE, OPERDATE, PACKETNUM, PACKNAME, OPERTYPE, OPERSTAT, PACKET, SYS_DATE).

6.Проверяем последовательность пакетов от клиента. Получаем номер последнего импортированного пакета от этого клиента из Журнала импорта (ЖИ) с данной маской (PACK_IMPORT_LAST_NUM) и проверяем, чтобы номер в имени пакета был на единицу больше последнего импортированного пакета. Если нет – формируем ошибку (записываем в Журнал ошибок (ЖО) COUNTER, JP_KEY (ссылка на ЖП.COUNTER), ERR_MESSAGE, JOURNAL_TYPE) и стопорим импорт пакетов с этой маской от этого клиента (в поле ЖЗ.IMPORT_ERROR_CODE пишем код ошибки из ЖО) и обновляем запись в ЖП (OPERSTAT = ошибка).

7.Если последовательность не нарушена, подключаемся к БД, описанной в ЖЗ (SERVER, BASE, LOGIN, PASSWORD, AuthorizationOS)

8. Разархивируем пакет.

9.Выполняем скрипт.

10. В случае успеха, формируем RSP-пакет (пустой файл с признаком успешной обработки первичного пакета от ЦОД), выкладываем его в директорию клиента (при этом используется Журнал клиентов (ЖК)) и обновляем ЖП (OPERSTAT = успешно обработан).

11.В случае ошибки, делаем запись в ЖО (COUNTER, JP_KEY, ERR_MESSAGE, JOURNAL_TYPE) и формируем RSP-пакет (архивный файл с признаком ошибочной обработки в имени и содержащим текстовый файл ошибки (ERR_MESSAGE)). Выкладываем в директорию клиента.

Таким образом, серверный импорт задействует таблицы ЖЗ, ЖП, ЖО, ЖК, ЖИ.

2.7. Алгоритм экспорта пакета от сервера (для чайников)

1. срабатывает таймер

2.Открываем массив ЖЗ

3.Проверяем каждую активную задачу (ACTIVE = 1 и EXPORT_ERROR_CODE = 0) на предмет необходимости немедленного выполнения экспорта (CURRENT_PACK_EXPORT_DELAY = 1 – локальная программная переменная массива ЖЗ)

4.Если CURRENT_PACK_EXPORT_DELAY > 1 уменьшаем на единицу и переходим к следующей задачи

5.Если CURRENT_PACK_EXPORT_DELAY = 1, то запускаем экспорт

6.Подключаемся к БД, описанной в ЖЗ (SERVER, BASE, LOGIN, PASSWORD, AuthorizationOS)

- задаем CURRENT_PACK_EXPORT_DELAY =
PACK_EXPORT_DELAY

- по каждому издателю выполняем следующие действия

- проверяем каждую активную подзадачу массива Журнала подзадач (ЖпЗ) (`ACTIVE = 1`) для этой задачи на предмет необходимости немедленного выполнения экспорта этой подзадачи (`ЖпЗ.CURRENT_EXPORT_DELAY = 1` – локальная программная переменная массива ЖпЗ)

- сравнения интервалов нужно производить только один раз для всех издателей интервалы будут такими же, а следовательно их изменять нужно только один в конце (когда запущен экспорт этого пакета от последнего издателя, (`Length(количество издателей) - 1 = номеру издателя в массиве(это последний)`)). Если `ЖпЗ.CURRENT_EXPORT_DELAY > 1` уменьшаем на единицу. Сравниваем с `ЖЗ.CURRENT_PACK_EXPORT_DELAY` – если `ЖпЗ.CURRENT_EXPORT_DELAY < ЖЗ.CURRENT_PACK_EXPORT_DELAY`, то задаем `ЖЗ.CURRENT_PACK_EXPORT_DELAY = ЖпЗ.CURRENT_EXPORT_DELAY` и переходим к следующей подзадаче – если `ЖпЗ.CURRENT_EXPORT_DELAY = 1`, то запускаем экспорт данной подзадачи (каждая подзадача отвечает за экспорт отдельной таблицы)

- экспорт должен начаться с переноса всех (принадлежащих этому издателю, т.е. `LPUIN = номеру издателя`, для которого сервер производит экспорт) `COUNTER` и `STATE (STATE2)` в Журнал откатов (ЖОт), выполняется скрипт, который выбирает из таблицы подзадачи все записи у которых: `LPUIN = номеру клиента издателя` и (`STATE <> 0` или `STATE2 <> 0` (если есть)), и записывает их `COUNTER, STATE, STATE2` (если есть) с именем формируемого пакета в ЖОт, это необходимо для отката значений полей `STATE` в случае ошибки

- экспортируемые записи записываются в файл экспорта имеющий расширение «SQL», это текстовый файл содержащий скрипты, в начале в него записывается шапка скрипта (содержит начало блока `Try` и старт транзакции), затем тело – состоящее из скриптов `INSERT` и `UPDATE` на

экспортируемые записи по каждой подзадаче поочередно, а в конце добавляется окончание с блоком завершения или отката транзакции (в случае ошибки) и обработки ошибки.

- экспорт каждой подзадачи начинается с перебора всех записей таблицы, у которых STATE = 1 (используются значения из ЖОт). Для каждой записи, выполняется скрипта который формирует инструкцию INSERT на добавление этой записи. Затем для записей у которых STATE \neq 1 (или STATE2 \neq 0) выполняется скрипт который формирует инструкцию UPDATE на обновлённые записи.

- после того как файл экспорта полностью сформирован (экспорт завершён для всех активных подзадач), архивируем его с паролем в архив и задаём этому архиву имя по следующей маске: LPUIN(9) + PACK_MASK(5) + PACK_EXPORT_LAST_NUM(9) + . + YMDhm(5)

- в случае ошибки, делаем запись в ЖО (COUNTER, JP_KEY, ERR_MESSAGE, JOURNAL_TYPE)

- в случае успеха, раскладываем этот пакет всем клиентам, подписавшимся на данные от этого издателя из Журнала подписчиков (ЖПод) изменяя в пакете номер пакета (для каждого клиента по каждой задаче он свой и берётся из Журнала номеров пакетов экспорта (ЖНП)).

Таким образом, серверный экспорт задействует таблицы ЖЗ, ЖпЗ, ЖП, ЖО, ЖОт, ЖК, ЖНП, ЖПод.

2.8. Критические допущения и запреты

- 1) Не возможна передача данных ЛПУ, не являющимся подписчиком на эти данные, т.к. невозможно определить была ли ему передана эта запись ранее или нет!

- 2) LPUIN – всего лишь уникальный код АРМа – не несёт информацию о ЛПУ.

- 3) В будущем все должны перейти на единую амбулаторную карту.

- 4) Подумать над добавлением поля XSTATE.