МЕДИАЛОГ
Медицинская информационная система

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

МОДУЛЬ "ПРОТОКОЛ HL7"

«Пост Модерн Технолоджи»
2017

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ МОДУЛЯ "ПРОТОКОЛ HL7" ... 3

2. ЛИЦЕНЗИОННАЯ ПОЛИТИКА ... 4

3. ФАЙЛЫ ПОСТАВКИ ... 5

4. ОБЩАЯ СХЕМА НАСТРОЙКИ СИНХРОНИЗАЦИИ ПО ПРОТОКОЛУ HL7 ... 6

5. УСТАНОВКА И НАСТРОЙКА HL7 СЕРВЕРА ... 7

5.1. Установка HL7 сервера ... 7

5.2. Настройка HL7 сервера ... 7

5.2.1. Основные настройки HL7 сервера (HL7.ini) ... 7

5.2.2. Настройки WEB службы и службы Windows (HL7Server.exe.config) ... 8

5.3. Запуск HL7 сервера ... 9

6. НАСТРОЙКА СХЕМ ВХОДА И ВЫХОДА ... 11

6.1. Настройка конфигурации исходящих сообщений (HL7_OUT.xml) ... 11

6.2. Настройка конфигурации входящих сообщений (HL7_IN.xml) ... 14

7. НАСТРОЙКА ТРИГГЕРОВ ТАБЛИЦ ... 18

8. HL7 МОНИТОР ... 20

8.1. Установка и настройка HL7 монитора ... 20

8.2. Интерфейс HL7 монитора ... 20

9. ПАРАМЕТРЫ МОДУЛЯ HL7 ... 23

10. ТАБЛИЦЫ БАЗЫ ДАННЫХ МОДУЛЯ "ПРОТОКОЛ HL7" ... 24

1. НАЗНАЧЕНИЕ МОДУЛЯ "ПРОТОКОЛ HL7"

Медицинская информационная система МЕДИАЛОГ поддерживает стандарт Health Level 7 (HL7). Поддержка стандарта HL7 позволяет интегрировать МЕДИАЛОГ с другими медицинскими системами, например, с PACS системами известных западных производителей, и обеспечить обмен клинической, финансовой и административной информацией.

Подробнее о протоколе HL7 см. официальный сайт http://www.hl7.org.

2. ЛИЦЕНЗИОННАЯ ПОЛИТИКА

Для возможности использования функционала модуля "Протокол HL7" в МИС МЕДИАЛОГ необходимо наличие одной серверной лицензии на модуль *"Протокол HL7"*.

Лицензия *"Интеграция с PACS"* приобретается на каждый подключаемый к PACS прибор.

Подробнее о лицензиях и получении ключа активации см. в файле **Pmtlics.pdf**.

Из таблицы ниже видно, что лицензию и опцию к ней возможно приобрести дополнительно в рамках тарифных планов Standart и Enterprise.

Модуль/Опция	**Тип лицензии**	**Тарифный план**			
		Compact	**Standart**	**Enterprise**	**Модульный**
ПРОТОКОЛ HL7	сервер	✖			
Интеграция с PACS	прибор	✖			

✓ - *включено в стандартный набор тарифного плана*
✖ - *недоступно в рамках тарифного плана*
_ - *возможно приобрести дополнительно к набору*

См. также разделы "Типы лицензий" и "Тарифные планы" в файле **Pmtlics.pdf**.

3. ФАЙЛЫ ПОСТАВКИ

1. **HL7 веб-сервер**
   - HL7.ini – настройки HL7 сервера
   - HL7_IN.xml – конфигурация входящих сообщений
   - HL7_OUT.xml – конфигурация исходящих сообщений
   - HL7Server.exe – сервер
   - HL7Server.exe.config – настройки web службы и службы windows
   - HL7Server.exe.manifest – манифест сервера
   - UniLibrary.dll – внутренняя библиотека web службы
   - console.bat – файл запуска веб-сервера в консольном режиме
   - reg.bat – файл регистрации веб-сервера в качестве службы windows
   - unreg.bat – файл удаления веб-сервера в качестве службы windows
2. **HL7 COM объект (используется HL7 веб-сервером)**
   - HL7Lib.dll – HL7 COM объект
   - reg.bat – файл регистрации com-объект
   - unreg.bat – файл удаления com-объекта
3. **HL7 Монитор**
   - HL7Mon.exe – HL7 Монитор
   - HL7Mon.ini – настройки HL7 Монитора

Принцип работы

HL7 веб-сервер (1) при старте начинает сканировать таблицу HL7_EVENTS и в случае появления в ней новых событий создает соответствующее каждому событию HL7 сообщение и отправляет его удаленному веб-серверу. Одновременно HL7 веб-сервер принимает входящие HL7 сообщения от удаленного веб-сервера и обрабатывает их согласно настройкам.

HL7 COM объект (2) необходим для работы HL7 веб-сервера.

HL7 Монитор (3) используется для контроля за состоянием и текущими событиями HL7 веб-сервера. Его можно запускать удаленно.

4. ОБЩАЯ СХЕМА НАСТРОЙКИ СИНХРОНИЗАЦИИ ПО ПРОТОКОЛУ HL7

- *Функциональность доступна при наличии лицензии на модуль "Протокол HL7"*

Настройки синхронизации по протоколу HL7 выполняются по следующей схеме:

1. Установить и настроить HL7 сервер.
2. Настроить схемы входа и выхода (определить формат входящих и исходящих сообщений, которыми будут обмениваться две системы).
3. Настроить триггеры таблиц, по которым будут выгружаться изменения.
4. В параметрах модулей МЕДИАЛОГ разрешить синхронизацию по протоколу HL7.
5. Установить и настроить HL7 монитор.

5. УСТАНОВКА И НАСТРОЙКА HL7 СЕРВЕРА

- *Функциональность доступна при наличии лицензии на модуль "Протокол HL7"*

Для работы HL7 сервера требуется установленный Microsoft .NET Framework 3.5.

- Установка HL7 сервера
- Настройка HL7 сервера
- Запуск HL7 сервера

5.1. Установка HL7 сервера

Для установки HL7 сервера необходимо выполнить следующие действия:

1. Установить Microsoft .NET Framework 3.5, если это не было сделано ранее.
2. Скопировать файлы поставки в каталог, откуда будет запускаться HL7 сервер (например, C:\PMT\HL7*).
3. Зарегистрировать COM объект, который будет использовать HL7 сервер: для этого нужно выполнить файл HL7Lib\reg.bat.
4. Настроить HL7 сервис: файл HL7Server\HL7Server.exe.config.
5. Зарегистрировать HL7 сервер в качестве службы Windows: для этого нужно выполнить файл HL7Server\reg.bat.
6. Настроить соединение с базой данных: файл HL7Server\hl7.ini

*Не рекомендуется копировать файлы в каталог Program Files, в этом случае возможны проблемы при регистрации службы.

5.2. Настройка HL7 сервера

Настройки HL7 сервера выполняются в следующих файлах:

- Основные настройки HL7 сервера (HL7.ini)
- Настройки WEB службы и службы Windows (HL7Server.exe.config)

Файлы должны находиться в каталоге, куда установлен HL7 сервер и в котором находится основной исполняемый файл HL7Server.exe (например, C:\Program Files\PMT\HL7\HL7Server).

5.2.1. Основные настройки HL7 сервера (HL7.ini)

Файл HL7.ini. Основные настройки HL7 сервера.

Редактирование файла производится в текстовом редакторе или с помощью специальной утилиты ToolSetup (Administration.pdf). Настройка строки соединения возможна только с помощью данной утилиты.

```
[common]
```

```
LogServer=1
```

Логирование основных событий HL7 сервера. Автоматически создается файл \log\server.log.

Ошибки независимо от этого параметра также записываются в \log\errors.log

```
LogMessages=0
```

Логирование входящих и исходящих сообщений в файлы \log\input.log \log\output.log.

Данный параметр по умолчанию выключен. Так как длина сообщений может быть большой, запись всех сообщений в файл может тормозить работу сервера и занимать много места на диске, поэтому рекомендуется включать этот параметр только на время отладки.

`OwnEvents=0`

Данный параметр заставляет HL7 сервер обрабатывать только те события из таблицы HL7_EVENTS, на которые он настроен. Это дает возможность одновременно использовать несколько HL7 серверов с одной базой данных. Параметр по умолчанию выключен.

`LogSize=200`

Количество хранящихся на сервере последних событий. Значение по умолчанию — 200.

`[database]`

`ConnectionString=xxx`

Строка соединения с базой данных.

`Interval=10`

Периодичность сканирования новых событий в таблице HL7_EVENTS в секундах. Минимум — 2 секунды.

`MaxSendCount=20`

Количество повторных отправок HL7 сообщения. Сообщение посылается повторно, если на него не приходит подтверждающее сообщение.

5.2.2. Настройки WEB службы и службы Windows (HL7Server.exe.config)

Файл HL7Server.exe.config. Настройки WEB службы и службы Windows.

```
<configuration>
      <appSettings>
            <!-- ProgID управляемого com-сервера -->
            <add key="ComProgID" value="HL7Lib.Proc" />
            <!-- Идентификатор windows сервиса -->
            <add key="ServiceName" value="hl7_server" />
            <!-- Имя windows сервиса -->
            <add key="ServiceDisplay" value="PMT HL7 Server" />
            <!-- Описание windows сервиса -->
            <add key="ServiceDescription" value="HL7 сервер" />
            <!-- Настройки SOAP сервиса (1 - использовать opHL7Message; 2 -
использовать opHL7Message2; 3- файловый обмен; 4 - функция прямого tcp-
соединения; иначе - ошибка)-->
            <add key="Func" value="1"/>

            <!-- Параметры для 3-го метода -->
            <!-- Интервал сканирования входящих сообщений в секундах (по-
умолчанию 60 секунд, минимум 2 секунды) -->
            <add key="ScanInterval" value="10" />
            <!-- Способ сортировки входящих файлов (Name, Date) -->
            <add key="ScanSort" value="Date" />
            <!-- Папка для входящих сообщений -->
            <add key="InFolder" value="e:\HL7\local\in" />
            <!-- Папка для исходящих сообщений -->
            <add key="OutFolder" value="e:\HL7\local\out" />

            <!-- Параметры для 4-го метода -->
            <!-- Локальный порт для входящих сообщений -->
            <add key="LocalPort" value="7777" />
            <!-- Удаленный хост для исходящих сообщений -->
            <add key="RemoteHost" value="127.0.0.1" />
```

```
            <!-- Удаленный порт для исходящих сообщений -->
            <add key="RemotePort" value="8888" />

      </appSettings>

      ...

      <system.serviceModel>
            <services>
                  <service name="HL7Server.HL7WebService"
behaviorConfiguration="ServiceBehavior">
                        <host>
                              <baseAddresses>
                                    <!-- Адрес локального HL7 веб-сервиса для
входящих сообщений. Можно указать любой незанятый порт.-->
                                    <add
baseAddress="http://192.168.1.23:12346/HL7Service" />
                        </host>
                  </service>
            </services>
      ...

            <client>
                  <!-- Адрес удаленного веб-сервиса, для отправки исходящих
сообщений -->
                  <endpoint address="http://80.253.2.26/ig_mu/wshl7_ig_mu.svc"
                binding="basicHttpBinding"
bindingConfiguration="bindingHl7Message"
                contract="HL7Server.portHl7Message" name="Hl7Message" />
            </client>
      </system.serviceModel>
      ...
</configuration>
```

5.3. Запуск HL7 сервера

HL7 сервер может быть запущен в виде службы или в консольном режиме.

Сервер, установленный в виде службы, запускается автоматически при старте операционной системы.

Чтобы изменить какие-либо параметры службы или перезапустить ее, нужно открыть Панель управления, выбрать Администрирование / Службы, и открыть службу **PMT HL7 Server** (рис. 5.1).

Примечание. Служба регистрируется с именем, заданным в настройках (HL7Server.exe.config); таким образом, имя службы может отличаться от "PMT HL7 Server".

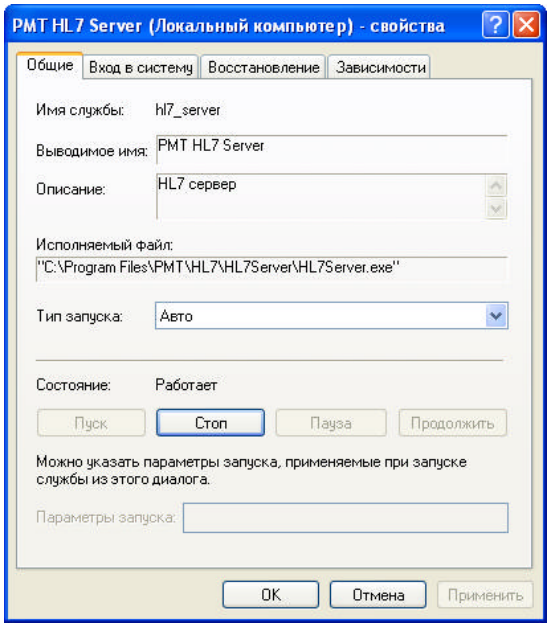

Рисунок 5.1. Служба PMT HL7 Server

Чтобы запустить сервер в консольном режиме, нужно выполнить файл HL7Server\console.bat (рис. 5.2). В консольном режиме выводится отладочная информация по HL7 Server.

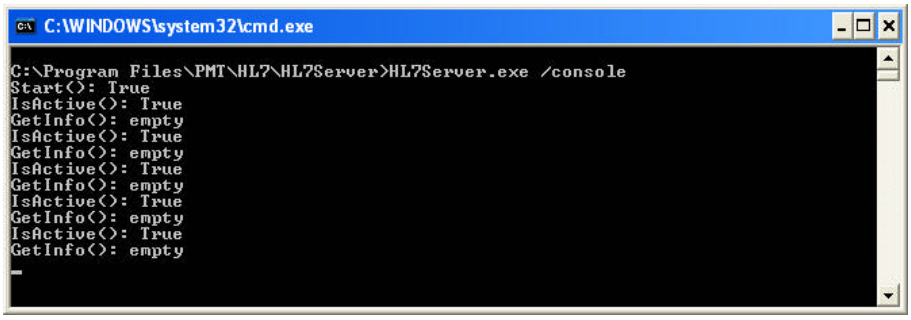

Рисунок 5.2. HL7 сервер, запущенный в консольном режиме

6. НАСТРОЙКА СХЕМ ВХОДА И ВЫХОДА

- *Функциональность доступна при наличии лицензии на модуль "Протокол HL7"*

- Настройка конфигурации исходящих сообщений (HL7_OUT.xml)
- Настройка конфигурации входящих сообщений (HL7_IN.xml)

6.1. Настройка конфигурации исходящих сообщений (HL7_OUT.xml)

Файл HL7_OUT.xml. Настройка конфигурации исходящих сообщений.

```
<?xml version="1.0" encoding="windows-1251"?>
<!— Кодировка самого файла HL7_OUT.xml должна соответствовать указанной здесь
кодировке-->
<hl7>
      <events>
            <!—
            В секции перечисляются обрабатываемые события из таблицы
HL7_EVENTS.
<code> - содержимое поля HL7_EVENTS.REC_TYPE + HL7_EVENTS.ACTION_TYPE. Если
поле HL7_EVENTS.ACTION_TYPE не пусто, его значение прибавляется со знаком
подчеркивания.
<name> - используется только в этом файле настроек (для удобства названия
секций) и его можно не задавать. Тогда <name> будет совпадать с <code>.
            -->
            <event desc="Изменение пациента">
                  <name>PATIENT</name>
                  <code>P_U</code>
            </event>
            <event desc="Запись на прием">
                  <name>PLANNING</name>
                  <code>PLANNING_UPDATE</code>
            </event>
      </events>

      <sections>
            <!—
            Секции описывают формат сообщения. Если имя секции совпадает с
описанными выше событиями (events), то на основе этой секции и будет построено
HL7 сообщение. Также есть промежуточные секции (например, common): они
используются в других секциях (через свойство <parent>), чтобы избежать
настройки для каждого сообщения одного и того же.
            -->
            <COMMON>
                  <xml>
                        <!—
                        Кодировка исходящего сообщения. Перед отправкой
сообщение будет перекодировано  в соответствии с указанной кодировкой, она же
будет указана в xml-заголовке отправляемого сообщения.
                        -->
                        <encoding>UTF-8</encoding>
                  </xml>
                  <segments>
                        <!—
                        Из этих сегментов будет состоять исходящее xml-
сообщение
```

```
                        -->
                        <MESSAGE_INFO>
                              <!–
                              В фигурных скобках указываются имена полей sql-
запросов. При формировании сообщения описанные ниже (в секции queries) запросы
будут выполнены и из результирующей записи значения полей будут соответственно
подставлены в исходящее сообщение (уже без фигурных скобок).
Есть несколько фиксированных системных полей, значение которых заполняет сам
HL7 сервер:
{MESSAGE_ID} - автоматически генерируется для нового сообщения
{EVENT_ID} - HL7_EVENTS. HL7_EVENTS_ID
{REC_ID} - HL7_EVENTS.REC_ID
{HL7_COMMENT} - HL7_EVENTS.COMMENT
{EXT_CODE} - HL7_EVENTS.EXT_CODE
{EXT_CODE1} - HL7_EVENTS.EXT_CODE1
{EXT_CODE2} - HL7_EVENTS.EXT_CODE2
{CURRENT_DATE} - HL7_EVENTS.CURRENT_DATE
                              -->
                              <MESSAGE_ID>{MESSAGE_ID}</MESSAGE_ID>
                              <DATETIME>{CURRENT_DATE}</DATETIME>
                        </MESSAGE_INFO>
                  </segments>
            </COMMON>

            <PATIENT>
                  <parent>COMMON</parent>
                  <xml>
                        <!–
                        Имя корневого элемента исходящего HL7 сообщения. Его
можно не указывать, тогда будет браться имя секции. В данном случае - PATIENT.
                        -->
                        <name>PAT</name>
                  </xml>
                  <segments>
                        <PATIENT_INFO>
                              <PATIENTS_ID>{PATIENTS_ID}</PATIENTS_ID>
                              <NOM>{NOM}</NOM>
                              <PRENOM>{PRENOM}</PRENOM>
                              <BIRTHDAY>{BIRTHDAY}</BIRTHDAY >
                        </PATIENT_INFO>
                        <!–
                        В сегменте может быть указан конкретный SQL-запрос. Это
значит, что данный сегмент может быть повторен в исходящем сообщении несколько
раз, в зависимости от числа записей, которые вернет этот запрос при выполнении.
                        -->
                        <INSURANCE query="insurance">
                              <PROGRAM_ID>{PROGRAM_ID}</PROGRAM_ID>
                              <PROGRAM_CODE>{PROGRAM_CODE}</PROGRAM_CODE>
                        </INSURANCE>
                  </segments>
                  <queries>
                        <!–
                        SQL-скрипты, которые будут последовательно выполнены
перед созданием исходящего сообщения.
<order> - порядок выполнения скрипта (если не указан, то выполняются
последовательно)
<type> - open или exec (определяет способ выполнения скрипта)
```

```
<name> - можно указать имя запроса. Это необязательно и нужно только для
повторяющихся сегментов (таких, как INSURANCE — см. выше)
<text> - текст скрипта.
В скриптах могут использоваться параметры (:PARAM_NAME), которые по имени
должны совпадать с системными полями (см. выше) либо с именами полей предыдущих
запросов.
Например, первый запрос (типа open) возвращает несколько полей, значение
которых можно через параметры использовать во втором и последующих скриптах.
                        -->
                        <query>
                              <order>1</order>
                              <type>open</type>
                              <text>
select
  PATIENTS_ID,
  NOM,
  PRENOM,
  NE_LE as BIRTHDAY
from
  PATIENTS
where
  PATIENTS_ID = :REC_ID
                              </text>
                        </query>
                        <query>
                              <order>2</order>
                              <type>open</type>
                              <name>insurance</name>
                              <text>
select
  FM_CLINK.FM_CLINK_ID as PROGRAM_ID,
  FM_CLINK.CODE as PROGRAM_CODE
from
  FM_CLINK_PATIENTS
  inner join FM_CLINK ON FM_CLINK.FM_CLINK_ID = FM_CLINK_PATIENTS.FM_CLINK_ID
where
  FM_CLINK_PATIENTS.PATIENTS_ID = :REC_ID
                              </text>
                        </query>
                  </queries>
            </PATIENT>

            <!—
            Следующая секция (сообщение) наследуется от PATIENTS,
следовательно, будет использовать его структуру. Также наследуются и скрипты,
но они могут быть изменены на свои. Структура сообщения также будет дополнена.
            -->
            <PLANNING>
                  <parent>PATIENT</parent>
                  <segments>
                        <PLANNING_INFO>
                              <PLANNING_ID>{PLANNING_ID}</PLANNING_ID>
                              <COMMENT>{PLANNING_COMMENT}</COMMENT>
                              <EXAM_CODE>{EXAM_CODE}</EXAM_CODE>
                              <EXAM_NAME>{EXAM_NAME}</EXAM_NAME>
                              <SUBJ_ID>{SUBJ_ID}</SUBJ_ID>
                              <SUBJ_NAME>{SUBJ_NAME}</SUBJ_NAME>
```

```
                        </PLANNING_INFO>
                  </segments>
                  <queries>
                        <query>
                              <type>open</type>
                              <text>
select
  p.PATIENTS_ID,
  p.NOM,
  p.PRENOM,
  p.NE_LE as BIRTHDAY,
  pl.PLANNING_ID,
  pl.COMMENTAIRE as PLANNING_COMMENT,
  ex.CODE as EXAM_CODE,
  ex.NAME as EXAM_NAME,
  sj.pl_subj_id as SUBJ_ID,
  sj.NAME as SUBJ_NAME
from
  planning pl join patients p on pl.PATIENTS_ID = p.PATIENTS_ID
              join pl_exam ex on ex.pl_exam_id = pl.pl_exam_id
              join pl_subj sj on sj.pl_subj_id = pl.pl_subj_id
where
  pl.PLANNING_ID = :REC_ID
                              </text>
                        </query>
                        <query>
                              <order>3</order>
                              <type>exec</type>
                              <text>
delete from HL7_EVENTS where MESSAGE_ID = :MESSAGE_ID
                                    </text>
                        <query>
                  </queries>
            </PLANNING>
      </sections>
</hl7>
```

6.2. Настройка конфигурации входящих сообщений (HL7_IN.xml)

Файл HL7_IN.xml. Настройка конфигурации входящих сообщений.

```
<?xml version="1.0" encoding="windows-1251"?>
<!— Кодировка самого файла HL7_IN.xml должна соответствовать указанной здесь
кодировке-->
<hl7>
      <events>
            <!—
            В данной секции перечисляются обрабатываемые входящие HL7
сообщения.
<code> - имя корневого элемента HL7 сообщения
<name> - используется только в этом файле настроек (для удобства названия
секций) и его можно не задавать. Тогда <name> будет совпадать с <code>.
            -->
            <event desc = "изменение пациента">
                  <name>PATIENT</name>
                  <code>ADT_A08</code>
            </event>
            <event desc = "подтверждение сообщения">
```

```
                <name>ACK</name>
                <code>ACK</code>
            <!-- Маска для приема файлов. Указывается при файловом
обмене (Func=3)-->
                <mask>*</mask>
            </event>
        </events>
        <sections>
            <!–
            Секции описывают формат сообщения. Имя секции должно совпадать с
описанными выше сообщениями (events). Также есть промежуточные секции
(например, common), они используются в других секциях (через свойство <parent>)
чтобы избежать настраивания для каждого сообщения одного и того же.
            -->
            <COMMON>
                <!–
                Описываем сегменты входящего сообщения: где какие данные
                -->
                <segments>
                    <MESSAGE_INFO>
                        <!–
                        В фигурных скобках указываются имена полей, к
которым потом можно обращаться в SQL-запросах.
                        -->
                        <MESSAGE_ID>{MESSAGE_ID}</MESSAGE_ID>
                        <DATETIME>{CURRENT_DATE}</DATETIME>
                    </MESSAGE_INFO>
                </segments>
            </COMMON>
            <PATIENT>
                <parent>COMMON</parent>
                <!–
                Если заполнено поле <answer_event>, то после выполнения всех
sql-скриптов будет создана новая запись в таблице HL7_EVENTS. Это нужно для
формирования ответного сообщения. Для записи в HL7_EVENTS используются
следующие значения:
REC_TYPE = значение секции <answer_event>
ACTION_TYPE = значение секции <answer_action> - это поле в HL7_EVENTS
использовать не обязательно
EXT_CODE1 = {EXT_CODE1}
EXT_CODE2 = {EXT_CODE2}
Последние два поля можно использовать для передачи каких-то параметров для
ответного сообщения. Например, входящее сообщение – запрос по пациенту по
СНИЛС. Тогда через {EXT_CODE1} можно передать СНИЛС, чтобы при обработке этой
записи из HL7_EVENTS можно было найти пациента и отправить ответ.
                -->
                <answer_event>PATIENT_ACK</answer_event>
                <segments>
                    <PATIENT_INFO>
                        <GUID>{GUID}</GUID>
                        <NOM>{NOM}</NOM>
                        <PRENOM>{PRENOM}</PRENOM>
                        <BIRTHDAY type="datetime">{BIRTHDAY}</BIRTHDAY>
                        <SEX>{POL}</SEX>
                    </PATIENT_INFO>
                        <!–
                        Если к секции привязан конкретный SQL-скрипт, то
```

```
он будет выполнен столько раз, сколько раз повторяется данная секция.
                                -->
                        <INSURANCE query="insurance">
                                <ID>{EXT_CLINK_PATIENTS_ID}</ID>
                                <PROGRAM_CODE>{PROGRAM_CODE}</PROGRAM_CODE>
                                <CONTR_CODE>{CONTR_CODE}</CONTR_CODE>
                                <!—
                                Для корректного формирования sql-скрипта надо
указывать тип поля. Возможные зуачения: "boolean", "integer", "float",
"datetime", в противном случае строка (для строковых значений указывать тип не
обязательно).
                                -->
                                <START_DATE type="datetime">{START_DATE}</
START_DATE>
                                <END_DATE type="datetime">{END_DATE}</END_DATE>
                                <ISDELETED type="boolean">{ISDELETED}</ISDELETED>
                                <GUID>{CLINK_GUID}</GUID>
                        </INSURANCE>
                  </segments>
                  <queries>
                        <query>
                                <order>1</order>
                                <!—
                                Здесь тип именно EXEC, потому что в данном случае
нужно не вернуть набор данных, а выполнить SQL-скрипт. Если есть хотя бы один
SQL-скрипт типа EXEC, то все скрипты будут выполняться в рамках одной
транзакции.
                                -->
                                <type>exec</type>
                                <text>
declare @pid int
declare @ne_le datetime
declare @pol bit
declare @count_rec int
declare @ERRMSG varchar(1024)

set @pid = null
set @ne_le = :BIRTHDAY
set @pol = case when :POL = 'M' then 0
                when :POL = 'F' then 1
                else NULL
           end

set @count_rec = (select count(*) from patients where GUID = :GUID)

if @count_rec = 0
begin
      exec up_get_id 'PATIENTS', 1, @pid output
      insert patients (patients_id, GUID, NOM, PRENOM, NE_LE, POL)
                      values (@pid, :GUID, :NOM, :PRENOM, ne_le, @pol)
end
else
if @count_rec = 1
begin
      select @pid = patients_id from patients where GUID = :GUID
      update patients set
            NOM = :NOM, PRENOM = :PRENOM, NE_LE = @ne_le, POL = @pol
      where patients_id = @pid
```

```
end
else
begin
      set @ERRMSG = 'Некорректный входящий GUID пациента. GUID [' + :GUID + ']'
      raiserror 50000 @ERRMSG
      return
end
                              </text>
                        </query>
                  </queries>
            </PATIENT>

            <!-
            Типичный пример обработки простого ответного сообщения
(подтверждающего получение нашего сообщения). Соответствующая запись из
HL7_EVENTS будет перенесена в HL7_EVENTS_LOG с указанием примечаний, которые
пришли с этим ответом.
            -->
            <ACK>
                  <parent>COMMON</parent>
                  <segments>
                        <ACK_INFO>
                              <MESSAGE_ID>{ACK_MESSAGE_ID}</MESSAGE_ID>
                              <CODE>{ACK_CODE}</CODE>
                              <COMMENT>{ACK_TEXT}</COMMENT>
                        </ACK_INFO>
                  </segments>
                  <queries>
                        <query>
                              <order>1</order>
                              <type>exec</type>
                              <text>
insert HL7_EVENTS_LOG
      (HL7_EVENTS_ID, REC_ID, REC_TYPE, ACTION_TYPE, EXT_CODE, CREATION_DATE,
SEND_COUNT, MESSAGE_ID, SUCCESS, COMMENT)
      select
            HL7_EVENTS_ID, REC_ID, REC_TYPE, ACTION_TYPE, EXT_CODE,
CREATION_DATE, SEND_COUNT,
            :ACK_MESSAGE_ID, case when :ACK_CODE = 'AA' then 1 else 0
end, :ACK_TEXT
      from HL7_EVENTS where MESSAGE_ID = :ACK_MESSAGE_ID
                              </text>
                        </query>
                        <query>
                              <order>2</order>
                              <type>exec</type>
                              <text>
delete from HL7_EVENTS where MESSAGE_ID = :ACK_MESSAGE_ID
                              </text>
                        </query>
                  </queries>
            </ACK>
      </sections>
</hl7>
```

7. НАСТРОЙКА ТРИГГЕРОВ ТАБЛИЦ

Чтобы создать пользовательский триггер для таблицы, необходимо открыть Microsoft SQL Server Management Studio, подключиться к серверу, выбрать базу данных и открыть раздел **Triggers** (рис. 7.1). Здесь можно создать новый триггер или изменить существующий.

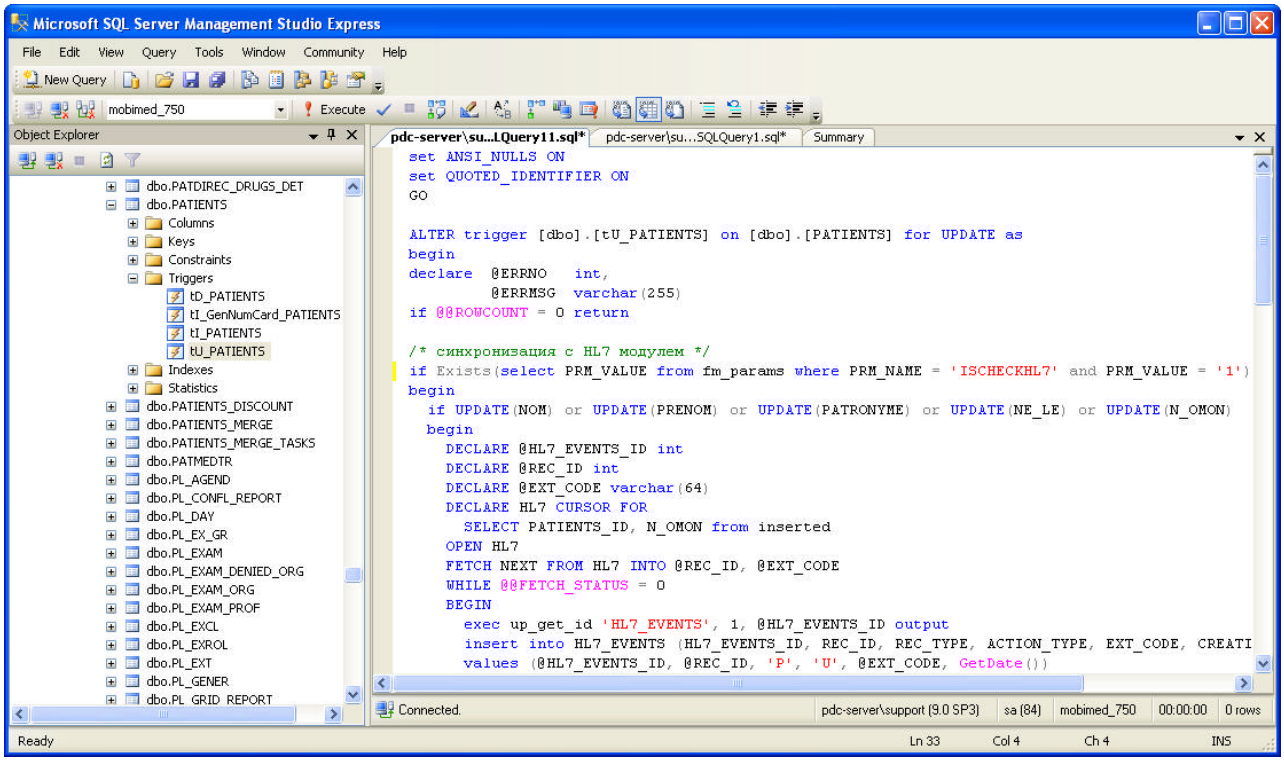

Рисунок 7.1. Редактирование триггера для таблицы PATIENTS

Триггер, отвечающий за синхронизацию по протоколу HL7, записывает определенные данные в таблицу HL7_EVENTS. Структура таблицы и характеристика содержащихся в ней данных приведены в разделе ТАБЛИЦЫ БАЗЫ ДАННЫХ МОДУЛЯ "ПРОТОКОЛ HL7".

Ниже представлен пример триггера, выполняющегося при изменении пациента, а именно — полей NOM (Фамилия), PRENOM (Имя), PATRONYME (Отчество), NE_LE (Дата рождения), N_OMON (Номер ЭМК) таблицы PATIENTS (Пациенты).

Триггер выполняется только в том случае, когда в параметрах модуля установлен флаг **Синхронизация медицинской информации через HL7 протокол**, т.е. FM_PARAMS.PRM_VALUE=1 для параметра с именем 'ISCHECKHL7'. Проверка данного параметра

```
if Exists(select PRM_VALUE from fm_params where PRM_NAME = 'ISCHECKHL7' and
PRM_VALUE = '1')
```

должна выполняться для любого триггера.

При вставке значений в таблицу HL7_EVENTS

```
insert into HL7_EVENTS (HL7_EVENTS_ID, REC_ID, REC_TYPE, ACTION_TYPE, EXT_CODE,
CREATION_DATE)
values (@HL7_EVENTS_ID, @REC_ID, 'P', 'U', @EXT_CODE, GetDate())
```

значения, прописываемые в поля REC_TYPE и ACTION_TYPE, должны соответствовать коду события, заданному в файле настройки исходящих сообщений (HL7_OUT.xml). Если поле HL7_EVENTS.ACTION_TYPE не пусто, его значение прибавляется к HL7_EVENTS.REC_TYPE со

знаком подчеркивания, т.е. код события:

```
<code>P_U</code>
```

Поле REC_TYPE можно не использовать.

Пример триггера:

```
set ANSI_NULLS ON
set QUOTED_IDENTIFIER ON
GO

ALTER trigger [dbo].[tU_PATIENTS] on [dbo].[PATIENTS] for UPDATE as
begin
  /* синхронизация с HL7 модулем */
  /* для синхронизации нужно включить флаг Синхронизация медицинской информации
через HL7 протокол в параметрах системы */
if Exists(select PRM_VALUE from fm_params where PRM_NAME = 'ISCHECKHL7' and
PRM_VALUE = '1')
begin
  if UPDATE(NOM) or UPDATE(PRENOM) or UPDATE(PATRONYME) or UPDATE(NE_LE) or
UPDATE(N_OMON)
  begin
    DECLARE @HL7_EVENTS_ID int
    DECLARE @REC_ID int
    DECLARE @EXT_CODE varchar(64)
    DECLARE HL7 CURSOR FOR
      SELECT PATIENTS_ID, N_OMON from inserted
    OPEN HL7
    FETCH NEXT FROM HL7 INTO @REC_ID, @EXT_CODE
    WHILE @@FETCH_STATUS = 0
    BEGIN
      exec up_get_id 'HL7_EVENTS', 1, @HL7_EVENTS_ID output
      insert into HL7_EVENTS (HL7_EVENTS_ID, REC_ID, REC_TYPE, ACTION_TYPE,
EXT_CODE, CREATION_DATE)
      values (@HL7_EVENTS_ID, @REC_ID, 'P', 'U', @EXT_CODE, GetDate())
      FETCH NEXT FROM HL7 INTO @REC_ID, @EXT_CODE
    END
    CLOSE HL7
    DEALLOCATE HL7
  end
end
```

Подобные триггеры на создание, изменение и удаление пациента имеются в каждой базе. По умолчанию они неактивны. Чтобы активировать триггеры, достаточно удалить знаки комментария.

8. HL7 МОНИТОР

- *Функциональность доступна при наличии лицензии на модуль "Протокол HL7"*

- Установка и настройка HL7 монитора
- Интерфейс HL7 монитора

8.1. Установка и настройка HL7 монитора

HL7 монитор может быть установлен как на том же компьютере, что HL7 сервер, так и на одном или нескольких удаленных компьютерах.

Для установки HL7 монитора достаточно скопировать файлы поставки в каталог, откуда будет запускаться монитор (например, C:\Program Files\PMT\HL7\HL7Monitor), и настроить параметры запуска HL7 монитора в файле HL7Mon.ini:

```
[SERVICE]

; Адрес HL7 веб-сервера. Адрес и порт должны совпадать с указанными в настройках HL7 сервера (HL7Server.exe.config).
URL=http://192.168.1.23:12346/HL7Service

; Интервал запроса новой информации (обновление лога) от HL7 веб-сервера в секундах.
Interval=10

; Максимальное количество строк в логе событий.
LogSize=1000
```

8.2. Интерфейс HL7 монитора

Для запуска HL7 монитора нужно выполнить файл HL7Mon.exe. Откроется окно, изображенное на рис. 8.1.

В статусной строке внизу окна показано текущее состояние HL7 сервера:

- *HL7 сервер активен* — если сервер работает (принимает / отправляет сообщения),
- *HL7 сервер неактивен* — если сервер не работает (не принимает / не отправляет сообщения),
- *Отсутствует лицензия на протокол HL7* — если отсутсвует лицензия "Протокол HL7". Необходимо получить лицензию и перезапустить сервер.
- *Ошибка соединения с HL7 сервером, Ошибка соединения HL7 сервера с БД, Критическая ошибка HL7 сервера.*
  Возможные причины ошибок — остановка работы HL7 сервера (не запущена соответствующая служба), некорректная настройка монитора (неверно указаны адрес или порт HL7 сервера), какой-либо сбой в сети.

В основной таблице окна представлены события HL7 сервера. Как правило, событие связано с исходящим или входящим сообщением, которое можно просмотреть. В некоторых случаях событие не связано с сообщением, например, если в HL7_EVENTS окажется запись с таким событием МЕДИАЛОГ, на которое никаких собщений не настроено (строка 3 на рис. 8.1). Такие события имеют состояние *"Ошибка"*.

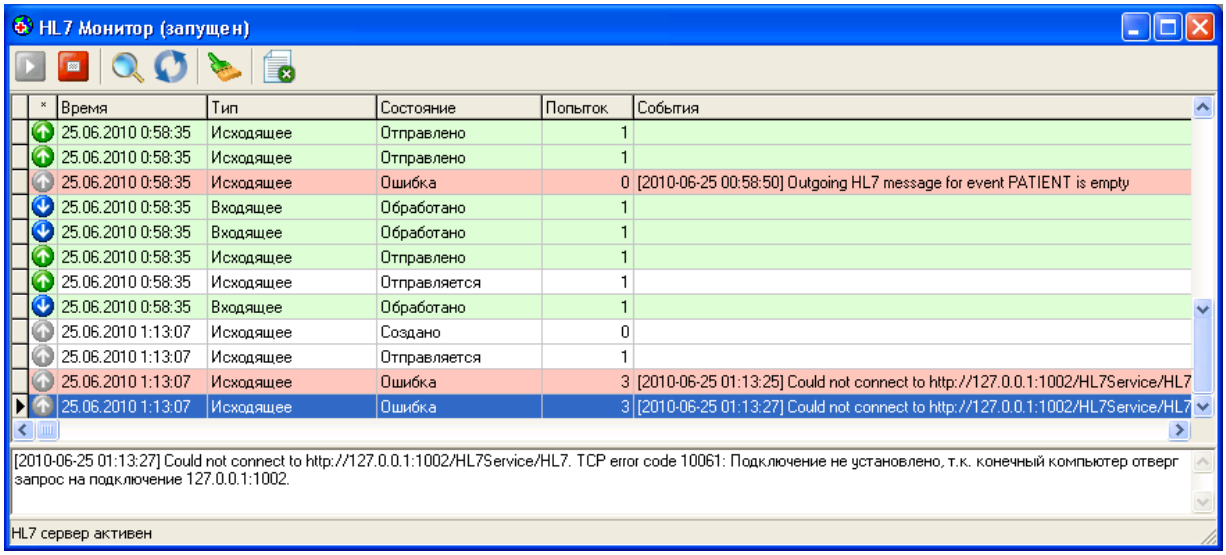

Рисунок 8.1. HL7 монитор

Для каждого сообщения в таблице отображается следующая информация:

- **Время** — время принятия входящего сообщения / создания исходящего сообщения,
- **Тип** — *исходящее* или *входящее*,
- **Состояние** — текущее состояние сообщения.
  Исходящее сообщение последовательно принимает состояния *Создано, Отправляется, Отправлено* (или *Ошибка*, если сообщение не удается отправить);
  Входящее сообщение принимает состояния *Обрабатывается, Обработано* (или *Ошибка*, если сообщение нельзя обработать).
- **Попыток** — количество попыток отправки / принятия сообщения,
- **События** — описание события. Полностью текст события отображается в нижней части окна.

На панели инструментов монитора расположены кнопки, которые позволяют:

- **Запустить монитор (F9)** — запустить мониторинг событий HL7 сервера (запрос у HL7 сервера новых событий);

- **Остановить монитор (F8)** — остановить мониторинг событий HL7 сервера;

- **Просмотр сообщения (F3)** — открыть выделенное сообщение в отдельном окне. HL7-сообщение может быть недоступно для просмотра, например, в случае, когда HL7 сервер был перезагружен.

- **Обновить лог (F5)** — обновить список событий. Команда запрашивает список событий с сервера, не дожидаясь окончания промежутка времени, заданного в параметре Interval (файл HL7.ini).

- **Очистить лог** — очистить список событий в окне HL7 Монитора. При этом сообщения не удаляются, и если перезапустить программу HL7 монитор или открыть ее копию, то список событий будет запрошен с HL7 сервера заново.
  Примечание 1. На сервере хранится ограниченное число последних событий. Оно определяется параметром в настройках сервера (файл HL7.ini).
  Примечание 2. Количество событий, которое отображается в логе, также ограничено. Оно определяется параметром в настройках HL7 монитора (файл HL7Mon.ini).

- **Закрыть** — свернуть HL7 монитор в трей.

Состояние монитора (запущен / остановлен) отображается в заголовке окна монитора и в подсказке иконки монитора в трее.

Свернутый в трей монитор всегда можно открыть двойным щелчком по его иконке или выполнив команду контекстного меню **Восстановить**; чтобы снова свернуть монитор — выполнить команду **Скрыть** (рис. 8.2).

Также в контекстном меню доступны команды, аналогичные командам в окне монитора: запустить или остановить монитор, обновить список сообщений, очистить лог.

Закрывается монитор командой **Выход**, при этом монитор в трее также будет закрыт.

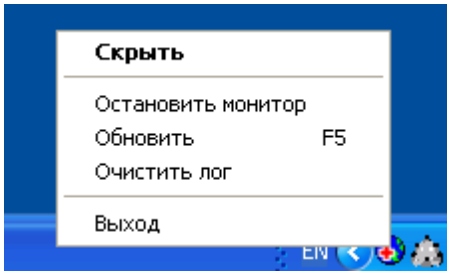

Рисунок 8.2. HL7 монитор в трее

9. ПАРАМЕТРЫ МОДУЛЯ HL7

Параметры модуля доступны из меню **Настройка / Параметры модулей**, закладка **PACS** (рис. 9.1).

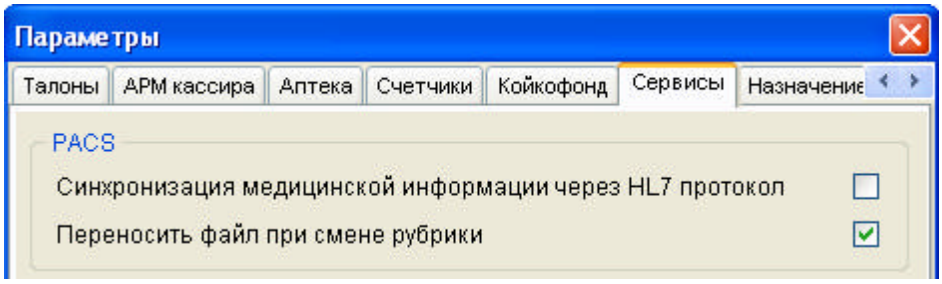

Рисунок 9.1. Параметры модуля "Протокол HL7"

Флажок **Синхронизация медицинской информации через HL7 протокол** позволяет выполнять триггеры, создающие записи в таблице HL7_EVENTS. См. подробнее НАСТРОЙКА ТРИГГЕРОВ ТАБЛИЦ.

10. ТАБЛИЦЫ БАЗЫ ДАННЫХ МОДУЛЯ "ПРОТОКОЛ HL7"

Таблица HL7_EVENTS

HL7_EVENTS — таблица событий МЕДИАЛОГ. Записи в ней создаются специальными триггерами. Если нужно синхронизировать данные в одной из таблиц с удаленной системой по HL7, то на изменяемую таблицу ставится триггер, создающий запись в таблице HL7_EVENTS. В каждой базе имеется неактивированный триггер на создание/изменение/удаление пациента.

При выполнении операции строка из этой таблицы удаляется и переносится в таблицу HL7_EVENTS_LOG.

Структура таблицы:

Имя поля	Тип поля	Данные
HL7_EVENTS_ID	int	Первичный ключ
REC_ID	int	Первичный ключ таблицы, в которой сработал триггер (PATIENTS_ID, PLANNING_ID и т.д.). Заполняется триггером.
REC_TYPE	varchar(20)	Текстовый код события, на который будет настроен HL7 сервер (например, PATIENT, MERGE, MOTCONSU_UPDATE). Заполняется триггером.
ACTION_TYPE	varchar(20)	Дополнительный код события (устаревшее поле).
EXT_CODE	varchar(100)	В эти поля записывается любая информация, которая может понадобиться при обработке события. Например, при объединении пациентов в REC_ID пишется основной PATIENTS_ID, а в EXT_CODE — ID объединяемого пациента. Заполняется триггером.
EXT_CODE1	varchar(100)	
EXT_CODE2	varchar(100)	
CREATION_DATE	datetime	Дата и время события. Заполняется триггером.
SEND_COUNT	int	Количество попыток сервера отослать сообщение на адрес выгрузки данных. Число увеличивается HL7-сервером автоматически при отправке сообщения. Сообщение будет периодически отправляться, пока не будет получено подтверждение получения или SEND_COUNT не достигнет значения MaxSendCount (определяется в файле HL7.ini).
MESSAGE_ID	varchar(32)	ID сообщения. Заполняется сервером при отправке сообщения.
PRIORITY	int	Приоритет. Заполняется автоматически сервером при создании событий, на которые надо создать сообщение в первую очередь. Чем больше значение поля, тем выше приоритет и тем выше будет расположено сообщение в очереди на отправку.
COMMENT	text	Комментарий. Используется сервером для сохранения текста возможной ошибки.
KRN_CREATE_DATE	datetime	
KRN_CREATE_USER_ID	int	
KRN_MODIFY_DATE	datetime	
KRN_MODIFY_USER_ID	int	

Таблица HL7_EVENTS_LOG

HL7_EVENTS_LOG — таблица обработанных событий МЕДИАЛОГ. Когда запись в HL7_EVENTS будет обработана, отправлено сообщение и получен на него ответ, эта запись переносится из HL7_EVENTS в HL7_EVENTS_LOG.

Структура таблицы:

Имя поля	Тип поля	Данные
HL7_EVENTS_ID	int	Первичный ключ.
REC_ID	int	Первичный ключ таблицы, в которой сработал триггер.
REC_TYPE	varchar(20)	Текстовый код события.
ACTION_TYPE	varchar(20)	Дополнительный код события.
EXT_CODE	varchar(100)	Дополнительная информация о событии.
CREATION_DATE	datetime	Дата и время события.
SUCCESS	bit	Признак успешно обработанного события.
COMMENT	text	Комментарий.
SEND_COUNT	int	Количество попыток сервера отослать сообщение на адрес выгрузки данных.
MESSAGE_ID	varchar(32)	ID сообщения.
KRN_CREATE_DATE	datetime	
KRN_CREATE_USER_ID	int	
KRN_MODIFY_DATE	datetime	
KRN_MODIFY_USER_ID	int	

Таблица HL7_EVENTS_ERR

HL7_EVENTS_ERR — таблица ошибок, возникших при обработке HL7 сообщений.

Структура таблицы:

Имя поля	Тип поля	Данные
HL7_EVENTS_ERR_ID	int	Первичный ключ
HL7_MESSAGE_TYPE	int	Тип сообщения
HL7_MESSAGE	text	Текст сообщения
ERROR_CODE	int	Код ошибки
ERROR_MESSAGE	text	Текст сообщения об ошибке
KRN_CREATE_DATE	datetime	
KRN_CREATE_USER_ID	int	
KRN_MODIFY_DATE	datetime	
KRN_MODIFY_USER_ID	int	