

Инструментальная Система управления потоками данных (ИСУПД). Архитектура Системы.

1 Состав Системы

Основными компонентами Системы являются:

А) **Сервер обработки**, реализован модулем ISUPD_Srv.exe.

Сервер является исполняющей средой Системы и выполняет следующие функции:

- управление Системой и организация всех этапов обработки данных;
- реализация и предоставление программных интерфейсов по взаимодействию с Системой;
- обеспечение информационного взаимодействия между различными экземплярами Системы, функционирующими на различных узлах локальной или глобальной сети;
- ведение журналов функционирования Системы.

Б) **Монитор Системы** – реализован модулем ISUPD_Monitor.exe. Выполняет следующие функции:

- предоставление диалоговых информационных и управляющих инструментов (механизмов) для оператора Системы;
- отображение журнала выполнения запросов к Системе;
- предоставление механизмов просмотра результатов выполнения запросов;
- предоставление механизмов просмотра данных Операций;
- предоставление механизмов активации сервисных обеспечивающих утилит Системы;

В) **WEB-сервисы** обеспечения API к функциональности Сервера по протоколу “SOAP”, реализованные модулями среды “ASP.net” из папки “BIN\WEBServer”;

Г) **Командный Процессор** ИСУПД – реализован модулями ISUPD_Comm.exe и ISUPD_CommC.exe.

Процессор является исполняющей средой – посредником, исполняющим вызовы API сервера ИСУПД через интерпритацию специальных команд и используется в ПО, не имеющем механизмов доступа через COM/OLE интерфейсы.

Д) **Агент удалённого доступа к серверам ИСУПД**, реализован модулем ISUPD_ToRemote.exe.

Агент является исполняющей средой – посредником, функционирующим на ПК, где отсутствует среда исполнения ИСУПД. Т.е. ему не требуется доступ к SQL-серверу, реестру, файлам и библиотекам Арбат. Агент самодостаточен и ему только требуется удалённый доступ к одному или нескольким серверам ИСУПД по одному из 3-х поддерживаемых сетевых протоколов. Агент выполняет следующие функции:

- реализация и предоставление программных интерфейсов по взаимодействию с Системой;
- обеспечение информационного взаимодействия с различными экземплярами Системы, функционирующими на различных узлах локальной или глобальной сети.

Е) **Редактор системных объектов**, реализован модулем ISUPDObjEdit.exe.

Редактор объектов ИСУПД служит для:

- описания (конструирования) информационных объектов Системы и схем их связей и взаимодействия;
- описания взаимодействия объектов Системы с дополнительным программным обеспечением (схемы обработки, связь с событиями, сопоставление информационных объектов с модулями обработки);
- генерации программной и информационной инфраструктуры Системы для работы в среде MS SQL Server;
- импорт/экспорт описаний Объектов ИСУПД;
- предоставления дополнительных сервисных функций по обслуживанию Системы.

Ж) **Обработчики** - программные модули (библиотеки, классы, сборки, хранимые процедуры и т.д.), реализующие конкретные алгоритмы обработки данных. Подключаются к Системе динамически по мере её настройки и развития.

З) **Подсистема событий.**

Подсистема обеспечивает функционирование Системы по независимой событийной схеме. Она реализована в виде следующих элементов:

- Редактор описания событий (модуль EV_Edit.exe);
- встроенные программные и информационные единицы БД (таблицы, представления, триггеры, хранимые процедуры);
- Сервер обработки событий (системный монитор – модуль EV_Agent.exe);
- специальное API;
- вспомогательные библиотеки обработчиков для обеспечения взаимодействия с Подсистемой.

И) **Подсистема разграничения доступа.**

Подсистема управляет доступностью данных, объектов и действий над ними для конкретных лиц из числа обслуживающего персонала (операторы) или пользователей внешних систем. Она реализована в виде следующих элементов:

- Редактор описания объектов, субъектов и действий, являющихся участниками процесса по разграничению доступа (модуль Security.exe);
- встроенные программные и информационные единицы БД (таблицы, представления, триггеры, хранимые процедуры);
- специальное API.

К) **Подсистема конфигураций.**

Подсистема позволяет динамически (только за счёт описаний) настраивать Систему на выполнение определённого функционала, подключая в управляющий диалог требуемые вызовы или программные модули, создавать специализированные АРМ. Она реализована в виде следующих элементов:

- Редактор описания меню/АРМ (модуль Menu.exe);
- встроенные программные и информационные единицы БД (таблицы, представления, триггеры, хранимые процедуры);
- специальное API.

Л) **Подсистема отчётности и представлений** (модули AReports.exe, DBImages.exe).

М) **Подсистема администрирования** (обслуживания – модуль SysManager.exe).

Н) Прочие Подсистемы и вспомогательные программные единицы.

2 Информационная модель Системы

Система построена по объектно-ориентированному принципу. Вся информационная и программная структура ориентирована на обработку строго типизированных программных и информационных **Объектов**, которые предварительно должны быть описаны (сконструированы) в Системе с помощью специального Редактора объектов. Кроме того, должны быть описаны связи Объектов между собой, а также определены схемы их обработки.

Ниже приводится описание информационных и программных (выполняемых) Объектов Системы.

2.1 Атрибут

Информационный Объект «Атрибут» используется для типизации описания характеристик и способов обработки отдельных элементов хранения атомарных данных в других объектах Системы (объектах – агрегатах данных). Фактически, под объектом «Атрибут» можно понимать типизированное поле таблицы БД. Причём, в разных агрегатах данных один и тот же «Атрибут» всегда имеет один и тот же смысл, одни и те же характеристики и обрабатывается по одним и тем же правилам (алгоритмам).

«Атрибут» описывается рядом характеристик:

- **идентификатор**: должен быть уникальным и состоять не более чем из 32 алфавитно-цифровых символов (русские буквы не допускаются), составленных по правилам идентификации информационных объектов MS SQL Server;
- **наименование**: может быть произвольное, но не более 128 символов;
- **тип хранимых данных**: ‘bit’, ‘int’, ‘bigint’, ‘decimal’[(p [,s])], ‘money’, ‘float’[(n)], ‘date’, ‘time’, ‘datetime’, ‘varchar’[(n)], ‘guid’, ‘array’;
- **размер**:
 - количество символов для ‘varchar’ (если не задано, то ‘max’);
 - количество десятичных цифр для ‘decimal’ (если не задано, то 38);
 - количество двоичных разрядов для ‘float’ (если не задано, то 53);
- **точность**: количество цифр после запятой для ‘decimal’ (если не задано, то 0);
- **допустимость** хранения пустых данных (NULL);
- **умалчиваемое значение**;
- **признак** специального Атрибута–Параметра (идентификатор будет начинаться с символа ‘@’).

Параметр является разновидностью Атрибута. Он используется не для вхождения в Объекты-агрегаты, а для передачи данных в Обработчики при их активизации.

Значение Параметра берется из следующих источников (по приоритетности):

- последнее введенное в диалоге значение в рамках Процесса (параметр диалога);
- значение, заданное Параметру для конкретного Обработчика при проектировании;
- умалчиваемое значение соответствующего Атрибута-параметра, заданное при проектировании;
- значение Параметра в предыдущих Обработчиках в рамках текущего Процесса (наследование значений параметров);
- **признак Атрибута–Параметра диалога** (при активизации обработки вручную (по команде оператора) организуется диалог ввода значений таким параметрам);
- **признак ссылочного** (кодируемого) Атрибута с привязкой его к конкретному Справочнику Системы (значение Атрибута является кодом Справочника);
- **перечень специальных Обработчиков** Системы для выполнения действий:
 - проверка значения Атрибута при импорте;
 - получение (вычисление) значения Атрибута при импорте;
 - получение (вычисление) значения Атрибута при экспорте.
- **ссылка на Атрибут–родитель** для наследования характеристик (Атрибут может быть самостоятельным или наследником от другого (только самостоятельного) Атрибута. В этом случае он наследует все характеристики типизации своего старшего Атрибута-родителя).

Атрибут может быть простым или составным (агрегат). Агрегатный Атрибут (тип назван как «массив», но реально - это запись) имеет тип ‘array’ и он непосредственно не содержит никаких данных. Ему должны быть сопоставлены (подключены) другие Атрибуты (только простые), которые будут составлять элементы (записи) массива:

```
<Атрибут типа ‘array’> :=
  <Атрибут1> <Атрибут 2> ... <Атрибут n>
```

Атрибут с типом 'agray' может входить в Объект «Операция» (допускается иметь в своем составе только один Атрибут с типом 'agray') и служит для создания «Кубов» данных.

В Системе изначально определены следующие **служебные** Атрибуты:

atObjectName – базовый Атрибут описания идентификаторов Объектов Системы;

@OperName – параметр - идентификатор Операции (наследуется). Если тип обрабатываемого объекта у Процесса «Операция» и значение не задано, то значение параметра используется как значение объекта обработки;

@DataName – параметр - идентификатор набора данных (наследуется). Если тип обрабатываемого объекта у Процесса «Набор» и значение не задано, то значение параметра используется как значение объекта обработки;

@TabName – параметр - идентификатор таблицы справочника (наследуется). Если тип обрабатываемого объекта у Процесса «Таблица» и значение не задано, то значение параметра используется как значение объекта обработки;

@FileName – параметр - имя/путь файла импорта/экспорта (наследуется). Если тип обрабатываемого объекта у Процесса «Файл» и значение не задано, то значение параметра используется как значение объекта обработки;

@FileDir – параметр - каталог файлов импорта/экспорта (наследуется);

@FileExt – параметр – перечень расширений имён файлов импорта/экспорта (через запятую);

@Flags0 – параметр - битовые флаги отбора данных по критерию: $(fiFlags \& @Flags0) = 0$ (не наследуется). Имеет умалчиваемое значение, вычисляемое по виду выполняемой обработки;

@Flags1 – параметр - битовые флаги отбора данных по критерию: $(fiFlags \& @Flags1) \neq 0$ (не наследуется). Имеет умалчиваемое значение, вычисляемое по виду выполняемой обработки;

@Source – параметр – GUID Системы-источника данных Операции: используется при отборе данных (наследуется);

@Destination – параметр – GUID Системы-приёмника данных Операции: используется при экспорте данных (наследуется);

@Where – параметр – дополнительное выражение (ограничение) для отбора данных Операции (не наследуется);

@Timeout – параметр – задаёт максимальное время в **секундах** (0 – нет, -1 – бесконечно) выполнения Обработчика (кроме Обработчиков, реализованных как DLL, OLE, COM). Параметр не наследуется и действует только на текущий Обработчик (у каждого Обработчика должен быть свой параметр). Если параметр не определён, то используется умалчиваемое значение 5 минут (для всех SQL запросов и хранимых процедур минимальное время выполнения в секундах может быть установлено в файле Arbat.ini: секция “[Shell]”, параметр “MinDBTimeOut”);

@Parameters – параметр – задаёт дополнительные параметры командной строки, передаваемые в среду исполнения Обработчика, реализованного как EXE или Windows-скрипт, например, для активизации отладчика Windows-скрипт (как правило, MS Visual Studio) необходимо задать: “//D//X”. Параметр не наследуется и действует только на текущий Обработчик (у каждого Обработчика должен быть свой параметр).

@Remote – параметр – задаёт протокол доступа к данным на удалённом компьютере: “DCOM” | “IPSOC” | “WEB” (наследуется);

@ComputerName – параметр – задаёт идентификатор удалённого компьютера для доступа через DCOM или IP-Socket (наследуется);

@IP – параметр – задаёт IP адрес удалённого компьютера (вместо идентификатора компьютера) для доступа по IP-Socket (наследуется);

@Port – параметр – задаёт № порта IP-Socket на удалённом компьютере (умолчание 211, наследуется);

@InterceptRouter – параметр (наследуется) – задаёт GUID (заклѳчен в фигурные скобки: {}) Com-класса шифрования/сжатия трафика при использовании протокола удалённого доступа IP Socket или идентификатор Обработчика, который выполняет эти функции;

@URL – параметр – задаёт http (https) путь к виртуальной папке удалённого компьютера, где находится модуль “httpsrvr.dll” для доступа через WEB (наследуется);

@Proxy – параметр – задаёт перечень Proxy-серверов для доступа к удалённому компьютеру через WEB (наследуется);

@UserName – параметр – задаёт Login пользователя для удалённого доступа. Если не задано, то используется текущий пользователь (наследуется);

@Password – параметр – задаёт пароль пользователя для удалённого доступа (наследуется);

@Strategy[x] – параметр – задаёт специальный вариант обработки – выбор некоторого возможного алгоритма из ряда предусмотренных в Системе (индивидуально оговаривается при описании Обработчиков. Наследуется);

@JobEvent – параметр – задаёт событие подсистемы событий Арбат, которое устанавливается в готовность после завершения Задания. Событие задаётся в виде:

“<Идентификатор события>[<Идентификатор объекта «соединение с БД»>][@<Представление>[.<Просмотр>]]”.

В среду управления событиями передаются данные события (функция API: “EV_SetEvReady” - см. описание подсистемы событий Арбат):

DataText=

[[<Представление>[.<Просмотр>]]&]0=<LogID>;1=<Операция>;2=<Задание>[<Параметры командной строки “DBImages.exe”>][@<Подстановочные данные для комментария через запятую>]]”;

DataID=<Код завершения Задания>;

DataComment=<Диагностическое сообщение>.

@LoadData_Date – параметр Обработчика типа “IGD”, задающий дату актуализации последних загруженных данных в Операцию. Вычисляется системой по журналу перед активизацией Обработчика импорта данных. Обработчик использует значение параметра для принятия решения о загрузке данных и должен обновить его значение на реальную дату загруженных данных.

2.2 Справочник

Объект Системы «Справочник» предназначен для типизации значений ссылочных Атрибутов. Такие Атрибуты в качестве своего значения содержат ссылку на значение (код значения), хранящееся в определённом Справочнике. Все Справочники Системы представляют из себя виртуальные таблицы (представления БД), состоящие из двух, трёх или более полей: «Поле кода» (идентификатор “fv_ID”), «Поле текста-расшифровки» (идентификатор “fc_Text”, тип “VARCHAR(???)”) и «Поле ID» (идентификатор “fi_ID” - только при наличии IDENTITY-поля в исходной таблице) с типами данных, как у исходной таблицы. Если при описании справочника было указано «Включать все поля исходной таблицы», то дополнительно к стандартным полям добавляются и все остальные поля.

Физически объект «Справочник» базируется на внешней (другая БД и/или другой сервер) или внутренней таблице (или представлении) БД. При создании объекта «Справочник» в редакторе объектов ИСУПД задаётся конкретная таблица (исходная таблица может содержать любое количество полей) и два её поля: поле кода и поле значения. Система автоматически сгенерирует представление БД и синоним для Справочника.

Для ссылочных Атрибутов Система автоматически обеспечивает обращение к соответствующему справочнику при работе с агрегатами данных, например Оперция: кроме поля “<Имя ссылочного Атрибута>”, содержащего код значения, автоматически обеспечивается поле “<Имя Атрибута>_Text”, содержащего расшифровку кода по справочнику. Кроме того система автоматически будет подставлять соответствующие значения в связанное поле при заполнении одного из них при импорте данных (при заполнении кода подставляется значение и наоборот).

Для ручного обращения к Справочникам используйте методы объекта среды (TShell) и представления БД:

[dbo].[vPDFFT_Gen_Dict_<Идентификатор Справочника>]

или:

[PDFТ].[<Идентификатор Справочника>]
или хранимую процедуру:
[dbo].[spPDFТ_Dicts].

Разные Справочники могут базироваться на одной и той же исходной таблице (использовать разные поля).

Импорт данных для Справочника (только Справочники, базирующиеся на внутренней таблице) осуществляется в базовые внутренние таблицы, из которых черпают данные Справочники. Для Справочников, базирующихся на внешних данных, импорт данных не выполняется.

«Справочник» описывается рядом характеристик:

- **идентификатор**: должен быть уникальным и состоять не более чем из 32 алфавитно-цифровых символов (русские буквы не допускаются), составленных по правилам идентификации информационных объектов MS SQL Server;

- **наименование**: может быть произвольное, но не более 128 символов;

- при создании нового объекта «Справочник» ему автоматически присваивается уникальный **GUID**, который наряду с идентификатором идентифицирует объект «Справочник»;

- для привязки Справочника к физической Таблице (или Представлению) задаются **идентификатор таблицы** (для внешних таблиц дополнительно – сервер или соединение, схема и БД), **поле идентификации** записей (код) и **поле значения**;

- **where-условие** для отбора записей Справочника из исходной таблицы.

Автоматическое сопоставление поступающих значений для ссылочных атрибутов и кодов справочника при импорте данных корректно работает только при полном соответствии поступающих значений и значений, находящихся в соответствующих справочниках. Если это не так, например, на входе поступает текст «обыкновенные», а в справочнике содержится «Акция обыкновенная именная» необходимо проводить перекодировку либо при загрузке самих справочников, либо при импорте данных. Одним из возможных механизмов, предоставляемых Системой, являются альтернативные справочники. Альтернативный справочник используется для автоматической перекодировки поступающих текстов значений ссылочных атрибутов и должен иметь идентификатор: “<Основной справочник>\$”. Такой справочник базируется на внутренней таблице (представление), состоящей из двух или более полей: первое поле, указываемое в качестве поля «Код значения» справочника, содержит тексты (только отличающиеся), поступающие при импорте данных Операции, а второе поле, указываемое в качестве поля «Текст значения» справочника, содержит тексты, содержащиеся в основном справочнике. Кодировка в таком случае выполняется в два этапа: сначала текст ищется в основном справочнике и, если он не найден, делается попытка его перекодировки по связанному альтернативному справочнику.

2.3 Операция

Описывает основной информационный объект Системы, представляющий собой агрегат данных, состоящий из набора Атрибутов и множества реализаций данных – записей.

Каждая Операция (или набор Операций) описывает определённый информационный процесс, реализуемый в одной из внешних прикладных систем, характеризующийся определённым набором данных и связей, подлежащих контролю (анализу данных) в Системе ИСУПД с целью выявления подмножества записей с использованием заложенных критериев (алгоритмов). Т.е. при планировании использования Системы ИСУПД в рамках (совместно с) конкретной прикладной системы выполняется анализ предметной области для выявления процессов, подлежащих контролю, после чего проектируется перечень и структура Операций Системы ИСУПД, моделирующих эти прикладные процессы.

Информационный объект «Операция» описывается рядом характеристик:

- **идентификатор**: должен быть уникальным и состоять не более чем из 32 алфавитно-цифровых символов (русские буквы не допускаются), составленных по правилам идентификации информационных объектов MS SQL Server;

- **наименование**: может быть произвольное, но не более 128 символов;
- при создании нового объекта «Операция» ему автоматически присваивается уникальный **GUID**, который наряду с идентификатором идентифицирует объект «Операция»;
- **источник данных** Операции:
 - **внутренние данные** импортируются (загружаются) из внешнего источника. Такие данные хранятся и обрабатываются во внутренних таблицах БД Системы, периодически подгружаясь/модифицируясь специальными системными Обработчиками;
 - **внешние данные**. Такие данные не переносятся в БД Системы. Доступ к ним осуществляется Системой в режиме Online путём активизации специального Обработчика получения внешних данных;
- разрешить **исключать данные** Операции из БД Системы:
 - допустимо исключение данных вручную оператором Системы;
 - допустимо автоматическое исключение помеченных данных сразу после их анализа;
 - допустимо автоматическое исключение данных при повторном импорте данных Операции (остаются только «новые» данные);
 - допустимо автоматическое исключение помеченных данных после их экспорта (передачи в другую Систему);
- описание **способа исключения** данных Операции из БД Системы:
 - Физическое удаление записей из БД;
 - Простановка специального признака исключения записей из БД (поле (fiFlags & 1) != 0);
 - Физическое удаление записей из БД с переносом данных в архивную БД;
- **атрибутный состав** Операции. Только один Атрибут каждой Операции может иметь тип 'массив';
- **перечень Обработчиков** по проверке (тестирования) данных Операции;
- **признак отключения** Операции (в этом случае она не обрабатывается Системой).

При описании Операции в редакторе объектов ИСУПД Система автоматически создаёт ряд таблиц и представлений в БД, позволяющих вести их обработку.

Имеется общая таблица данных для всех Операций ("tPDFT_OpersData"), состоящая из стандартных полей, описывающих время и источник поступления данных, флаги обработки, ...:

```

fcName      varchar(32),          -- Идентификатор Операции
fgID        uniqueidentifier, -- GUID записи Операции
fvID        sql_variant,      -- ID записи Операции в Системе-источнике
fiFlags     int,              -- Флаги контроля/управления
fgSource    uniqueidentifier, -- GUID Системы-источника данных
fdImport    datetime,        -- Дата/время получения данных
fdLastCheck datetime,        -- Дата/время последней проверки данных
fdFirstExpot datetime,       -- Дата/время первой передачи (экспорта) помеченных данных
fiLog_REF   bigint           -- ID записи журнала событий (tPDFT_EventsLog) последнего
                                     -- изменения

```

Используются следующие значения флагов (поле 'fiFlags') в этой таблице:

```

$00000001; // Данные удалены/отключены
$00000002; // Признак внешних данных Операции
$01000000; // Признак "имеются ошибки/отклонения,... в данных записи"
$06000000; // Маска серьёзности ошибки/отклонения,... в данных записи (0:3)
$08000000; // Признак "выполнен экспорт в заданный приёмник (@Destination и tPDFT_OpersExtDestList)"
$10000000; // Признак "тестирование данных записи выполнено"
$20000000; // Признак "установлена пометка записи для последующего экспорта"
$40000000; // Признак "выполнен экспорт помеченных данных"

```

Каждая запись общей таблицы соответствует одной записи данных Операции и идентифицируется уникальным GUID (поле 'fgID'). Даже Операции, базирующиеся на внешнем Online-источнике данных, имеют в ней свои записи.

Для каждой Операции создаётся таблица "tPDFT_Gen_Oper_<идентификатор Операции>", содержащая данные Атрибутов Операции. Для каждой записи этой таблицы соответствует своя запись в общей таблице данных Операций ("tPDFT_OpersData"), имеющие одно значение поля 'fgID'. Таблица состоит из полей, соответствующих Атрибутам Операции, а также ряда служебных полей:

```

fgID        uniqueidentifier, -- GUID записи Операции

```

```
fcArray_REF    varchar(32)          -- Идентификатор Атрибута-массива или NULL
```

При наличии в Операции Атрибута-массива, дополнительно создаётся таблица данных элементов массива “tPDFT_Gen_OperArr_<идентификатор Операции>”, в которой для каждой записи основной таблицы данных этой Операции имеется одна или несколько записей – элементов данных Атрибута-массива. Таблица состоит из полей, соответствующих Атрибутам, составляющим Атрибут-массив Операции, а также ряда служебных полей:

```
fgID           uniqueidentifier,    -- GUID записи Операции (как в таблице "tPDFT_OpersData")
fiArrIndex     smallint             -- Индекс элемента массива Атрибута-массива (от 0)
```

Также Системой создаются представления БД, позволяющие в одном запросе получать весь комплекс данных Операции (включая поля расшифровки для Атрибутов, кодируемых по Справочнику):

```
dbo.vPDFT_Gen_Oper_<идентификатор Операции>    - общее представление Операции;
PDFT_<идентификатор Операции>                 - синоним общего представления;
dbo.vPDFT_Gen_Oper0_<идентификатор Операции>   - представление основных данных Операции;
dbo.vPDFT_Gen_OperA_<идентификатор Операции>   - представление данных Атрибута-массива Операции.
```

2.4 Задание

«Задание» является программным объектом - исполняемой единицей Системы. Описывает поток действий (является контейнером) по обработке Объектов ИСУПД, задаваемый в виде перечня активизируемых «Процессов», режимов и способов их активизации. Именно этот объект определяет жизненный цикл функционирования Системы.

«Задание» «известно» вне Системы и, фактически, работа Системы и заключается в периодической активизации тех или иных «Заданий» через предоставляемый программный интерфейс (API), вручную оператором Системы или по событиям.

«Задание» описывается рядом характеристик:

- **идентификатор**: должен быть уникальным и состоять не более чем из 32 алфавитно-цифровых символов (русские буквы не допускаются), составленных по правилам идентификации информационных объектов MS SQL Server;

- **наименование**: может быть произвольное, но не более 128 символов;

- при создании нового объекта «Задание» ему автоматически присваивается уникальный **GUID**, который наряду с идентификатором идентифицирует объект «Задание»;

- разрешённые **способы активизации**:

- Вручную оператором из Монитора Системы;
- Автоматически по заданному Событию Подсистемы Событий;
- Из внешних Систем по специальному API (через COM/OLE объекты);
- Из внешних Систем по специальному API из удалённого компьютера;

- **перечень** и очерёдность активизируемых **Процессов** в рамках «Задания»;

- **признак отключения** «Задания» (в этом случае оно не обрабатывается Системой).

2.5 Процесс

«Процесс» является программным объектом - исполняемой единицей Системы. Описывает набор конкретных действий (является контейнером) над конкретным объектом, выполняемыми для достижения определенной законченной цели, например, выполнение импорта данных конкретной «Операции» или таблицы Справочника.

Процесс состоит из набора действий, выполняемых путем активизации подключенных «Обработчиков» в порядке, определяемом их нумерацией в редакторе Объектов ИСУПД или автоматически в зависимости от их типов (в любом случае при смешанной реализации однотипных Обработчиков сначала активизируются «Обработчики», реализованные для исполнения в среде Windows, а затем – для среды MS SQL Server).

В рамках «Процесса» запускаются «Обработчики»:

А). Обработчики, напрямую подключенные к Процессу;

Б). Обработчики, подключенные к заданному в Процессе Объекту, например, Обработчики, подключенные к заданной в Процессе Операции.

В). В случае импорта и экспорта данных запускаются также Обработчики, подключенные к Атрибутам, входящим в состав Операции или таблицы Справочника (Обработчики преобразования/проверки значений Атрибутов).

Для активизации Обработчика может понадобиться задание значений Атрибутам-параметрам Обработчика. Поэтому все Параметры, помеченные как «Параметры диалога», всех Обработчиков, подключенных к Процессам, выполняемым в рамках Задания, объединяются в один пул. В случае запуска Задания вручную по команде оператора в Мониторе ИСУПД для таких Параметров организуется диалог по вводу их значений. В других случаях запуска Задания используются либо умалчиваемые значения (заданы при описании Атрибутов или при описании Параметров конкретного Обработчика), либо последние значения, заданные в рамках этого Процесса при предыдущих запусках.

В ходе выполнения каждого Процесса по мере активизации подключенных к ним Обработчиков идёт последовательное подключение (накопление) к среде обработки всех их параметров. При этом, если параметр уже встречался у предыдущего Обработчика, его значение заменяется на новое. Таким образом, для последнего Обработчика Процесса будут доступны все параметры всех Обработчиков, косвенно или напрямую подключенных к Процессу. Следует также отметить, что любой Обработчик, используя предоставленное API, может самостоятельно изменить значение любого доступного параметра (это значение действует только в рамках текущего Процесса).

«Процесс» описывается рядом характеристик:

- **идентификатор**: должен быть уникальным и состоять не более чем из 32 алфавитно-цифровых символов (русские буквы не допускаются), составленных по правилам идентификации информационных объектов MS SQL Server;

- **наименование**: может быть произвольное, но не более 128 символов;

- **обрабатываемый информационный Объект**:

- Операция;
- Таблица/представление;
- Файл;
- Набор данных;
- Не задан – в этом случае тип объекта и сам объект обработки в Процессе сохраняется от предыдущего Процесса Задания или определяется Обработчиком самостоятельно по заданным параметрам.

- **перечень** и очерёдность активизируемых **Обработчиков** (к Процессу непосредственно могут быть подключены только нейтральные Обработчики, Обработчики по импорту данных, проверке/преобразованию данных, экспорту данных);

- **признак отключения** Процесса (в этом случае он не обрабатывается Системой).

2.6 Обработчик

«Обработчик» является программным объектом - исполняемой единицей Системы. Связывает конкретную обработку объекта с конкретной программной единицей – реализующим модулем, который разрабатывается отдельно от Системы по определённым правилам и реализует требуемый алгоритм обработки.

Модуль Обработчика может быть реализован на различных системах программирования (компилирующих или интерпретирующих типов) с использованием следующих способов реализации (программных сред):

- Хранимая процедура Transact-SQL MS SQL Server;
- Скрипт на Transact-SQL MS SQL Server (программный текст, хранимый в таблице описания конкретного Обработчика в БД Системы);
- Процедура-функция в DLL-библиотеке;

- Исполняемый Ехе-файл;
- Исполняемый программный код (скрипт) для одной из поддерживаемых интерпретирующих систем, например, “Visual Basic Script” или “Java Script”. Код может храниться в файле или в таблице описания конкретного Обработчика в БД Системы и выполняется (активизируется) стандартным обработчиком Windows, связанным с заданным расширением (типом) имени файла, например, “*.VBS” - для “Visual Basic Script”, а “*.JS” - для “Java Script” (см. Примечание ниже);
- Интерфейс объекта COM/OLE класса, зарегистрированного в Системе;
- Стандартная встроенная процедура обработки ИСУПД.

Примечание: для Обработчиков, реализуемых в виде скриптов для среды Windows, рекомендуется использовать формат “WSF” – универсальный “Windows script file”, который позволяет создавать пакеты, состоящие из различных языков (Java-script, VB-script, REXX, Perl и т.д.), причём одновременно. Тексты скриптов при этом могут располагаться как в отдельных файлах, так и в теле WSF-файла. Кроме того, возникает возможность использования констант, описанных в Type Library системы ИСУПД (ссылка: `reference guid="{CF0F6855-9A9D-4DB1-AD7C-F3B4BC64978E}"`) и других системных объектов Windows.

Пример возможного фрагмента WSF-файла:

```
<job id="ruIGD_PRCT01">
<reference guid="{CF0F6855-9A9D-4DB1-AD7C-F3B4BC64978E}"/>
<reference object="Scripting.FileSystemObject"/>
<script language="JScript" src="ruIGD_PRCT01.js">
</script>
</job>
```

При создании объекта «Обработчик» в редакторе объектов ИСУПД система предлагает стандартные заготовки (шаблоны) программных текстов для разных сред (SQL, VBS, JS, WSF) разных типов.

«Обработчик» описывается рядом характеристик:

- **идентификатор:** должен быть уникальным и состоять не более чем из 32 алфавитно-цифровых символов (русские буквы не допускаются), составленных по правилам идентификации информационных объектов MS SQL Server. Если идентификатор начинается с символов: “Dummy” (без учёта регистра), то такой Обработчик не будет обрабатываться (аналогично признаку отключения Обработчика).

Если идентификатор Обработчика имеет префикс: “Params” (без учёта регистра), имеет тип ‘NOP’ и Обработчик не подключен ни к одному Процессу, то параметры такого Обработчика становятся глобальными - они при старте любого Задания загружаются первыми не зависимо от Процессов и Обработчиков, подключенных к ним.

- **наименование:** может быть произвольное, но не более 128 символов;

- **тип Обработчика:**

<code>otNoOperation</code>	= ‘NOP’ : “No Operation” - Нейтральный - без привязки к стандартным действиям
<code>otCheckOperData</code>	= ‘TOD’ : “Test Oper Data” - Проверка (анализ) данных Операции (Правила)
<code>otImpGetData</code>	= ‘IGD’ : “Import Get Data” - Получение данных для импорта или получение внешних Online-данных Операции
<code>otImpTestData</code>	= ‘ITD’ : “Import Test Data” - Проверка данных импорта
<code>otImpGetField</code>	= ‘IGF’ : “Import Get Field” - Получение/преобразование значения Атрибута при импорте
<code>otImpTestField</code>	= ‘ITF’ : “Import Test Field” - Проверка значения Атрибута при импорте
<code>otExpGetData</code>	= ‘EGD’ : “Export Get Data” - Получение данных для экспорта
<code>otExpGetField</code>	= ‘EGF’ : “Export Get Field” - Получение/преобразование значения Атрибута при экспорте

- **способ описания** (реализации);

- ссылка (идентификация) на программный модуль:

- Путь к DLL, EXE или Скрипт-файлу;
- Идентификатор процедуры или метода реализации;
- Идентификатор OLE-класса или GUID OLE/COM-класса;
- GUID интерфейса;
- Текст скрипта;

- Командная строка;
- перечень **Атрибутов-параметров** и их значений;
- **признак отключения** Обработчика (в этом случае он не обрабатывается Системой).

Обработчик должен вернуть код завершения одним из следующих способов:

- как результат вызова (DLL, интерфейс, SP);
- установка значения свойству Shell.RetCode (доступно везде);
- возврат кода завершения процесса (EXE, Windows-скрипт);
- установка значения переменной @rc (SQL-скрипт).

Для Обработчиков, реализованных в виде EXE, Windows-Скрипт или SQL, установлено ограничение на время обработки (время до их завершения) в 5 минут. Для изменения этого параметра используйте Атрибут-параметр “@Timeout” для конкретного Обработчика.

Обработчики могут быть 3-х видов.

Нейтральные Обработчики (не выполняющие стандартных действий, предусмотренных Системой). Перед активизацией таких Обработчиков Система не предпринимает никаких специальных действий и они функционируют в среде, созданной на предыдущих этапах работы. Такие Обработчики могут быть вспомогательными и служат для выполнения каких-то системных действий, например, выполнять обращение по API к конкретной внешней системе.

Типизированные Обработчики предназначены для выполнения строго определённых функций, предусмотренных в Системе. Перед активизацией таких Обработчиков Система, в зависимости от типа Обработчика и этапа обработки, выполняет определённые действия по подготовке данных и среды выполнения, а по завершении – выполняет стандартные действия по обработке подготовленных Обработчиками данных. Предусмотрены следующие типы таких Обработчиков:

- Контроль (анализ) данных Операций;
- Импорт данных для Операций или таблиц Справочников;
- Проверка импортируемых данных;
- Проверка значений Атрибутов импортируемых данных;
- Вычисление значений Атрибутов импортируемых данных;
- Экспорт данных Операций;
- Вычисление значений атрибутов экспортируемых данных.

Обработчики получения **внешних данных**. Специальный вид Обработчиков, которые обеспечивают получение в режиме Online данных из внешних источников для Операций, которые не хранят данные во внутренней оперативной БД Системы.

2.7 Соединение с БД

«Соединение» является информационным объектом, описывающим строку ADO-соединения с БД, и может содержать ссылку на UDL-файл, описывающий соединение, или непосредственно строку соединения. Дополнительными параметрами являются идентификатор пользователя и его пароль. «Соединение» используется в Обработчиках ИСУПД для создания альтернативных подключений к произвольным БД.

2.8 Удалённый Узел

«Удалённый Узел» является информационным объектом, описывающим рабочую станцию (ПК) или сервер, на котором функционирует ИСУПД, и способ удалённого доступа к нему (один узел с ИСУПД обращается к другому узлу с ИСУПД). Удалённый Узел может находиться как в локальной сети организации, так и располагаться в глобальной сети с доступом, например, через интернет. Объект «Удалённый Узел» используется в Обработчиках ИСУПД для установления связи с удалённым Узлом с целью совместного функционирования и использования удалённых ресурсов.

3 Функционирование Системы

Основная идея Системы (жизненный цикл) и, соответственно, направленность её функционирования это: сбор данных из различных источников данных, их проверка, очистка, преобразование и обобщение, запись данных во внутреннюю оперативную БД с последующим анализом этих данных. По результатам анализа данные помечаются Системой. Помеченные данные в последующем подлежат передаче на дальнейшую обработку в другой экземпляр Системы или экспортируются во внешние системы, после чего переданные данные уничтожаются и/или архивируются.

В соответствии с этим Система реализует следующие основные функции:

- предоставление механизмов по созданию и настройке информационных Объектов Системы;
- получение (импорт) данных для наполнения таблиц Справочников Системы;
- получение (импорт) данных для Операций, описанных в Системе;
- проверка, очистка и преобразование получаемых (импортируемых) данных;
- хранение импортируемых данных;
- анализ (тестирование) данных Операций и выработка заключений по ним;
- подготовка и выдача (экспорт) данных Операций по результатам их анализа (тестирования);
- архивирование/исключение уже использованных данных;
- генерация различного рода справочных данных о ходе работы Системы и результатах обработки данных.

Система может быть развёрнута на одном или нескольких компьютерах в локальной и/или глобальной сети. Для функционирования Системы также требуется MS SQL Server версии не ниже 2008-R2 (кроме узлов, где используется Агент удалённого доступа). Каждый узел, где развёрнута Система, может функционировать независимо или взаимодействовать друг с другом. ПО, функционирующее на разных узлах, может использовать один и тот же или различные SQL серверы, а также использовать одну и ту же или разные БД.

Ядром Системы является Сервер ИСУПД, реализованный в виде сервера COM/OLE классов в модуле "ISUPD_Srv.exe". Сервер может быть запущен заранее (режим службы Windows или процесс автозапуска) либо активизируется автоматически при создании объекта внешнего управления, например, приходе внешнего запроса на выполнение Задания на обработку данных. Запрос может выдать либо Оператор Системы, либо Подсистема Событий, либо другой узел при совместной работе в сети, либо другая внешняя Система, использующая API Системы. Все поступающие запросы выполняются одновременно в мульти-режиме (строго последовательно выполняются запросы от Подсистемы Событий). После выполнения всех запросов Сервер автоматически завершает свою работу (только при не активном Мониторе ИСУПД и, если он не является службой Windows и процессом автозапуска). Все поступающие запросы выполняются в одном экземпляре Сервера ИСУПД.

Монитор ИСУПД предоставляет диалоговое окно по визуальному контролю за работой Системы, а также предоставляет различные информационные и управляющие функции оператору Системы. Монитор функционирует на ПК, где развёрнута среда и Сервер ИСУПД. Монитор отслеживает запуск Заданий ИСУПД:

- локальных Заданий на локальном Сервере ИСУПД;
- Заданий на локальном Сервере ИСУПД, запущенных из удалённого сервера, сервиса или Агента ИСУПД;
- Заданий, запускаемых на удалённом Сервере по инициативе локального Сервера ИСУПД;
- Заданий, запускаемых на удалённых Серверах ИСУПД, находящихся в одной локальной сети и работающих с одной БД одного и того же SQL-сервера.

Для настройки Системы на конкретное применение осуществляется конструирование (описание) объектов Системы в редакторе Объектов ИСУПД, а также программируются и создаются необходимые Обработчики Системы. После чего Система готова к функционированию.

Функционирование Системы осуществляется путём обработки поступающих запросов на выполнение одного из описанных в Системе Заданий или получения внутренней среды (класс TShell) с последующим использованием её свойств и методов. Для выполнения «Запроса» создаётся COM/OLE объект класса “ISUPD_Srv.TExtCntr” (CLASS_TExtCntr: TGUID = '{868BFB76-EFBC-4B0A-9F3B-E96787E5A2BE}'), после чего осуществляется вызов метода ‘Execute’ или ‘ExecuteRemote’ с указанием идентификатора запускаемого Задания, либо на прямую используются методы внутренней среды. Альтернативой прямому созданию и использованию объекта класса “TExtCntr” является использование WEB-сервисов ИСУПД, публикуемых на “MS Internet Information Services”, доступных через протокол “Soap” и механизмов “ASP.net”, или использование методов объекта клиентского ПО класса “ISUPD_ToRemote.TRemCntr”.(CLASS_TRemCntr: TGUID = '{868BFB76-EFBD-4B0A-9F3B-E96787E5A2BE}'), реализуемых модулем “ISUPD_ToRemote.exe”;

Запросы порождаются:

- **Подсистемой Событий** при наступлении События, к которому подключено Задание. При этом Подсистема Событий может функционировать в режимах:

- Встроенный режим: Подсистема Событий функционирует в рамках модуля “ISUPD_Monitor.exe”;
- Внешний режим: Подсистема Событий реализует свои функции в рамках отдельно запущенного модуля “Ev_Agent.exe”;

- По **команде оператора** из Монитора Системы;

- **Самой Системой**, запущенной на другом компьютере в сети (удалённый доступ);

- **Другой Системой**, запущенной на этом или другом компьютере в сети.

При активизации Задания Система выявляет перечень Атрибутов-Параметров, помеченных признаком «Параметр диалога», всех Обработчиков, которые будут исполняться в рамках всех Процессов Задания, и, в случае запуска Задания оператором из Монитора Системы, организуется диалог ввода значений этим Параметрам. Остальным Параметрам присваиваются значения (в порядке приоритетности) следующим образом:

- последние запомненные значения Параметра в рамках конкретного Процесса или значение, установленное Параметру предыдущим Обработчиком;

- если таких значений нет, то значения по умолчанию, заданные при описании соответствующих Параметров для конкретного Обработчика;

- если таких значений нет, то значения по умолчанию, заданные при описании соответствующих Атрибутов-Параметров;

- если значения не заданы, а Атрибут не может быть NULL, то возбуждается состояние ошибки.

Далее Система активизирует Процессы, подключенные к данному Заданию (строго в соответствии с порядком списка подключения, если только сами Обработчики не изменяют порядок Процессов через вызовы методов/свойств объектов API). Активизируются только разрешенные (с отсутствием признака «Отключено») Процессы.

Каждый Процесс ориентирован на обработку одного строго определенного Объекта, заданного при его описании в редакторе Объектов ИСУПД. Для Процесса могут быть определены следующие Объекты:

- Операция;
- Таблица/Представление БД Справочников;
- Файл;
- Поименованный сохранённый ранее Набор данных.

Объект для Процесса может быть и не задан, тогда Объекты обработки должны быть определены (программно или по описанию) в самих Обработчиках и/или их Параметрах. Для этого могут использоваться индивидуальные Параметры или стандартные Параметры Системы (при

использовании стандартных параметров Система будет правильно выдавать значения при использовании соответствующих методов/свойств объектов API):

- @OperName – параметр - идентификатор Операции;
- @TabName – параметр - идентификатор таблицы справочника;
- @FileName – параметр - имя/путь файла импорта/экспорта;
- @FileDir – параметр – путь к папке файлов импорта/экспорта;
- @FileExt – параметр – перечень расширений имени обрабатываемых файлов (через запятую);
- @DataName – параметр - идентификатор сохранённого набора данных.

При активизации Процесса Система автоматически создаёт внутренние программные объекты и заполняет среду необходимыми данными, после чего выявляется перечень Обработчиков, которые должны быть активизированы в рамках этого Процесса:

А). Явно подключенные к Процессу при его описании (Обработчики типов: ‘NOP’, ‘IGD’, ‘ITD’, ‘EGD’, ‘TOD’);

Б). Обработчики тестирования/вычисления значений Атрибутов, входящих в импортируемую Операцию/Таблицу (Обработчики типов: ‘ITF’, ‘IGF’);

В). Обработчики вычисления значений Атрибутов, входящих в экспортируемую Операцию (Обработчики типа: ‘EGF’);

Применяется следующий порядок активизации Обработчиков:

- Обработчики, заданные в источнике А), в порядке строгой очерёдности, заданном в редакторе Объектов ИСУПД при подключении Обработчиков к Процессу, но с учётом ранжирования по типам Обработчиков в порядке: ‘IGD’, ‘IGF’, ‘ITF’, ‘ITD’, ‘TOD’, ‘EGF’, ‘EGD’, ‘NOP’;

- если в процессе активизации Обработчиков из источника А) встретился Обработчик типа ‘IGD’ и других Обработчиков данного типа в рамках Процесса больше нет, то после его выполнения последовательность активизации Обработчиков из источника А) приостанавливается и для всех Атрибутов, входящих в импортируемый Объект (Операция или Таблица), выполняются индивидуальные для Атрибутов Обработчики в очерёдности:

1. Вычисление значений вычисляемых Атрибутов в среде Windows;
 2. Проверка значений контролируемых Атрибутов в среде Windows;
 3. Вычисление значений вычисляемых Атрибутов в среде MS SQL Server;
 4. Проверка значений контролируемых Атрибутов в среде MS SQL Server;
- Далее обработка продолжается с точки приостановки;

- если в процессе активизации Обработчиков из источника А) встретился Обработчик типа ‘EGD’ и он является первым Обработчиком данного типа, то последовательность активизации Обработчиков из источника А) приостанавливается и для всех Атрибутов, входящих в экспортируемый Объект (Операция) выполняются индивидуальные для Атрибутов Обработчики в очерёдности:

1. Вычисление значений вычисляемых Атрибутов в среде Windows;
 2. Вычисление значений вычисляемых Атрибутов в среде MS SQL Server;
- Далее обработка продолжается с точки приостановки.

Обработчики могут самостоятельно изменить порядок (или даже пропуск) активизации Обработчиков в рамках Процесса с помощью специальных методов/свойств объектов API.

Более подробно о действиях Системы при активизации Обработчиков рассматривается при описании создания конкретных типов Обработчиков.

Обработчик реализуется отдельно от Системы по определённым правилам и подключается к Системе при его описании в Редакторе Объектов ИСУПД. Именно в Обработчике реализуются требуемые алгоритмы по обработке данных.

При активизации Обработчику передаётся среда Системы (в виде ссылки на интерфейс объекта типа “TShell”), из которой он получает сведения из описания Процесса, значения Параметров, построенные Системой или предыдущими Обработчиками наборы данных. Обработчик (не зависимо от способа реализации) принимает входные данные, выполняет над ними определенные действия (проверка и/или преобразование) и создает выходные данные, которые следующий Обработчик использует как входные (конвейерная обработка), либо вырабатывает сообщения

(заклучения) по анализу (тестированию) данных Операции с занесением результатов в журнал. Доступ к данным и выполнение над ними своих действий Обработчик выполняет с использованием свойств и методов объектов среды Системы, передаваемой в Обработчик при его запуске. По завершении Обработчик должен вернуть код возврата (способ зависит от средств реализации Обработчика):

= 0 – всё в норме;

= 1 – нет Обработчиков (обработка не выполнялась) – код Системы;

= 2 – нет требуемых данных (обработка не выполнялась);

> 2 < 10 – устранимые ошибки/ситуации, не приводящие к прекращению или изменению

алгоритма обработки;

= 10 – данный код соответствует конкретной ситуации: при анализе (тестировании) данных

Операции обнаружены отклонения (нарушения) в данных;

> 10 – ошибки, не приводящие к прекращению обработки, но отменяющие сохранение

изменённых данных в БД (кроме журналов обработки);

< 0 – серьёзные ошибки/сбои, требующие прекращения обработки и отмены всех

выполненных действий, включая отметки в журналах действий.

Предусмотрены следующие типы Обработчиков:

otNoOperation = 'NOP': Нейтральный без привязки к стандартным действиям

otCheckOperData = 'TOD': Проверка (анализ) данных Операции (Правила)

otImpGetData = 'IGD': Получить данные для импорта или получение Online-данных Операции из внешнего источника

otImpTestData = 'ITD': Проверить данные импорта

otImpGetField = 'IGF': Получить/преобразовать значение Атрибута при импорте (вычисляемые Атрибуты)

otImpTestField = 'ITF': Проверить значение Атрибута при импорте

otExpGetData = 'EGD': Получить данные для экспорта

otExpGetField = 'EGF': Получить/преобразовать значение Атрибута при экспорте (вычисляемые Атрибуты)

После завершения работы Обработчика Система выполнит проверку кода завершения и, в случае отсутствия ошибок, выполнит стандартные действия (эти действия зависят от типа обрабатываемого Объекта и типа Обработчика) по окончательной обработке данных, после чего осуществляется переход к следующему Обработчику в Процессе, а по их завершению – к следующему Процессу.

Т.к. Обработчики могут быть созданы для функционирования в разных средах (Windows, SQL-Server), Система автоматически обеспечивает передачу (синхронизацию) Параметров и подготовленных данных между этими средами, что может привести к замедлению её работы.

По завершении Задания при отсутствии ошибок (сбоев) все изменённые данные фиксируются в БД. Не зависимо от результатов формируются записи в журналах работы и выдаются отладочные сообщения Обработчиков.

4 Работа в распределённой вычислительной среде

4.1 Удалённый доступ к Системе

Предполагает использование механизмов DCOM и RPC Windows.

Система предоставляет программный интерфейс (класс "ISUPD_Srv.TExtCntr") в среде MS Windows через механизм COM / OLE объектов, позволяющий активизировать Задания Системы из стороннего ПО. Этот же механизм может быть использован для доступа к API Системы из стороннего ПО, которое функционирует на другом компьютере локальной вычислительной сети (ЛВС), путём применения DCOM интерфейса (требуется настройка системы безопасности Windows для использования механизмов DCOM и RPC). В этом случае Система может и не быть установлена

на компьютере-клиенте (откуда осуществляется вызов), а объект класса “ISUPD_Srv.TExtCntr” должен создаваться через специальные вызовы механизма DCOM (см. «Разработка»). Альтернативой прямому созданию и использованию объекта класса “TExtCntr” является использование WEB-сервисов ИСУПД, публикуемых на “MS Internet Information Services”, доступных через протокол “Soap”, а также использование объекта “TRemCntr” Агента удалённого доступа ИСУПД.

4.2 Взаимодействие экземпляров Системы различных узлов сети

Взаимодействие экземпляров Системы использует механизм «удалённого доступа к данным» посредством брокера удалённого доступа, который обеспечивает взаимодействие экземпляров Системы, функционирующих на разных компьютерах сети (удалённые Узлы), и предполагает использование одного из трёх предоставляемых (в данной версии Системы) протоколов:

- DCOM: требуется настройка системы безопасности Windows для использования механизмов DCOM и RPC;

- IP-Socket: требуется настройка Брандмауэра Windows для разрешения доступа к удалённому компьютеру через IP-порт (по умолчанию – 211). На удалённом компьютере также требуется запустить модуль “Scktsrvr.exe” из комплекта поставки;

- WEB: в качестве удалённого компьютера в этом случае выступает WEB-сервер. Требуется создать на нём виртуальную папку или «Приложение» и поместить туда модуль “httpsrvr.dll” из комплекта поставки. WEB-сервер должен быть настроен на использование протоколов http или https для этой папки (Приложения) для выполнения вызовов посредством механизма ISAPI.

В любом случае на всех взаимодействующих компьютерах должна быть установлена Система ИСУПД в полном или сокращённом (только Агент удалённого доступа) объеме.

Ниже рассмотрены различные варианты такого взаимодействия (см. «Разработка»):

А). Использование методов “TExtCntr.ExecuteRemote” и “TRemCntr.Execute” позволяет любому ПО выполнять активизацию Заданий на удалённом компьютере.

Б). Использование метода “TShell.ExecuteRemote” позволяет Обработчику, функционирующему на одном узле сети, активизировать Задание на удалённом компьютере.

В). Использование методов “TShell.LinkToRemote” и “TRemCntr.LinkToRemote” позволяет получить ссылку на интерфейс среды (объект типа TShell), созданной на удалённом компьютере. С некоторыми ограничениями данный интерфейс можно использовать для выполнения действий, предусмотренных API, на удалённом компьютере (надо представлять, что все выполняемые действия через такой интерфейс реально выполняются в среде Системы, созданной на другом узле в другом экземпляре Системы).

Г). Использование метода “TShell.SendFile” позволяет копировать файл из локальной системы на удалённый компьютер и обратно. Этим же целям служат методы клиентской среды “TRemCntr.SendFile” и “TRemCntr.ReceiveFile”

Д). Использование методов “TDataSet.SaveToFile” и “TDataSet.LoadFromFile” с указанием удалённого узла позволяет выполнять экспорт/импорт данных на/с удалённого компьютера.

Е). Использование метода “TDataSet.LoadFromData” при указании в качестве источника интерфейса набора данных, расположенного на удалённом ПК, позволяет получать данные из удалённого ПК.

Ж). Использование встроенных Обработчиков “irRemoteOperRead” и “irRemoteTableRead”, позволяет иметь доступ одному экземпляру Системы к данным (Операции или Таблицы), обслуживаемым другим экземпляром Системы, функционирующим на другом компьютере локальной или глобальной сети. Инициатором выступает узел, получающий (импортирующий) данные, в процессе выполнения Задания по импорту данных. Т.е. активизируемый Обработчик удалённого доступа осуществляет соединение с удалённым компьютером, активизирует (при необходимости) сервер Системы на нём, выполняет соответствующий запрос и получает данные для их импорта в локальную Систему.

4.3 Удалённый доступ к данным

Ещё одним случаем взаимодействия различных узлов может быть механизм Online-доступа к данным (Операций или Таблиц), находящимся в разных базах данных разных SQL Серверов. В этом случае могут быть использованы любые механизмы, предусмотренные в MS SQL Server (Link-Server, Репликация, OPENDATASOURCE, OPENROWSET и др.).