

Автоматический запуск заданий

Оглавление

1. Введение _____	3
2. Принцип работы модуля _____	4
3. Комплект поставки _____	5
4. Начало работы _____	6
5. Системные пакеты модуля IKP_ADM _____	9

1. Введение.

Компания «Инверсия-Кавказ» разработала модуль «Автоматического выполнения заданий»

Модуль предназначен для автоматизации периодических процессов и процессов по условию. Таких как загрузка/выгрузка файлов, запуск процедур и функций БД.

Модуль представлен в виде java приложения и серверной части на языке PL/SQL.

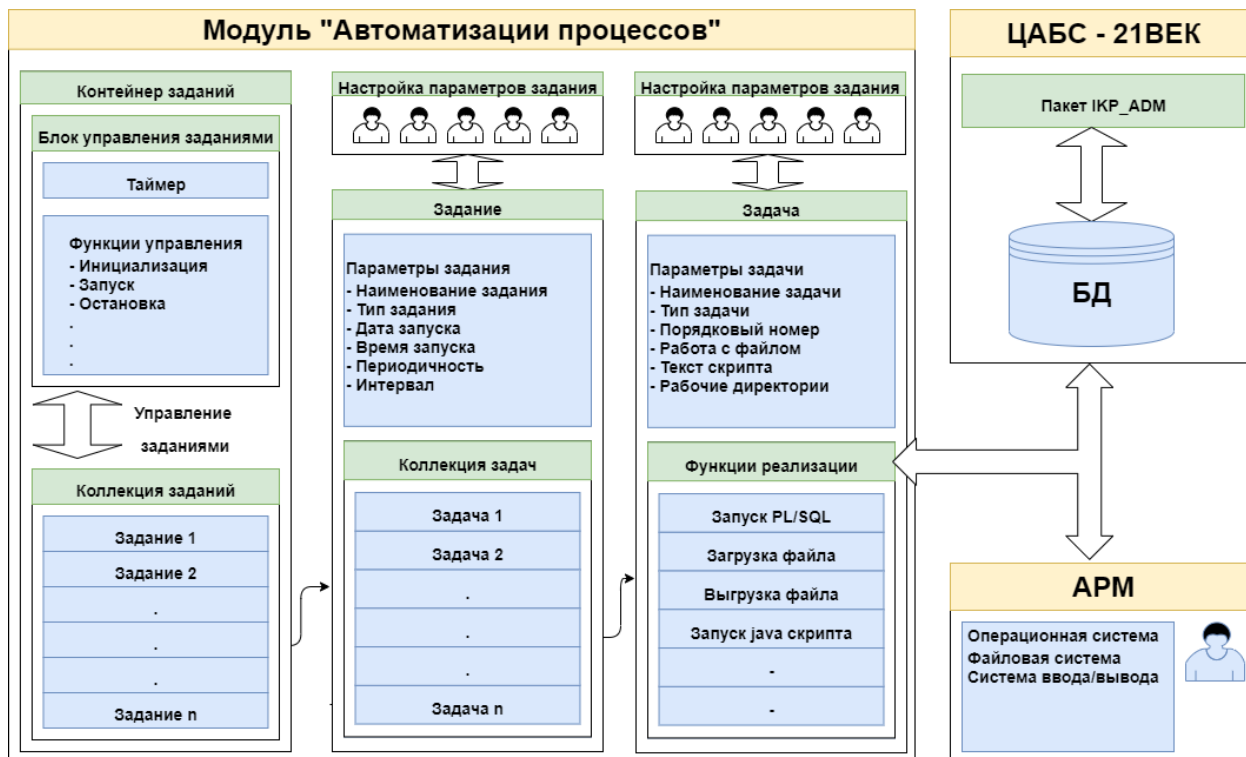
В клиентской части приложения реализован интерфейс по настройке модуля, управление заданиями, просмотр журнала выполнения заданий. Так же на стороне клиента решены задачи по работе с файлами.

Серверная часть модуля отвечает за обработку данных разбор загруженных файлов и последующую работу с полученной информацией. Также серверная часть отвечает за подготовку данных для выгрузки в файл.

Запуск задания может производиться по расписанию или принудительно.

Для каждого задания задается последовательность атомарных задач, выполняемых по порядку.

Структурная схема модуля:



2. Принцип работы модуля.

Для начала работы необходимо сформировать новое задание. Задание имеет перечень настроек необходимых для его работы. Таких как:

- Наименование задания.
- Периодичность выполнения.
- Частота выполнения.
- Сторона выполнения.
- Дата выполнения.
- Время запуска.
- Интервал запуска.

Что означает что событие будет выполняться, начиная с указанной даты в указанное время с указанной периодичностью через указанный интервал.

Каждое задание имеет перечень задач, выполняемых последовательно в соответствии с их порядковым номером.

Каждая задача имеет перечень настроек необходимых для его работы таких как:

- Наименование действия.
- Номер по порядку.
- Тип действия.
- Работа с файлом.
- Текст скрипта.
- Директории входящая, исходящая, архивная.

Задача может быть:

- Предустановленным действием. К примеру, загрузка/выгрузка разных типов файлов под разные процессинги, потребуется перенос логики из форм пластика в пакеты, тогда каждое действие может быть связано с процедурой/функцией в пакете.
- PL/SQL блоком.
- Скриптом(Java), при помощи которого можно выполнять задачи, не связанные с БД – к примеру, наложение цифровой подписи на сформированный файл; архивирование выгруженного/разархивирование входящего файла; ожидание файла в указанной директории и т.д.
- запуском команды ОС

Каждая атомарная задача имеет входящие и исходящие параметры, последующая атомарная задача может использовать в качестве входящего параметра исходящий параметр из предыдущей.

Поле текст скрипта содержит код написанный в синтаксисе java и может вызывать некоторые проблемно ориентированные встроенные функции. Полное описание встроенного языка см. Приложение1.

Запуск задания производится в соответствии с настройками и начинает последовательное выполнение заданий. Каждое задание выполняется в собственном потоке и создает отдельное подключение к БД.

Следует обратить внимание на тот факт, что принудительное закрытие приложение остановить выполнение текущих задач и может привести к непредсказуемым последствиям.

3. Комплект поставки.

В комплект поставки входит один jar файла основного модуля и набор необходимых библиотек.

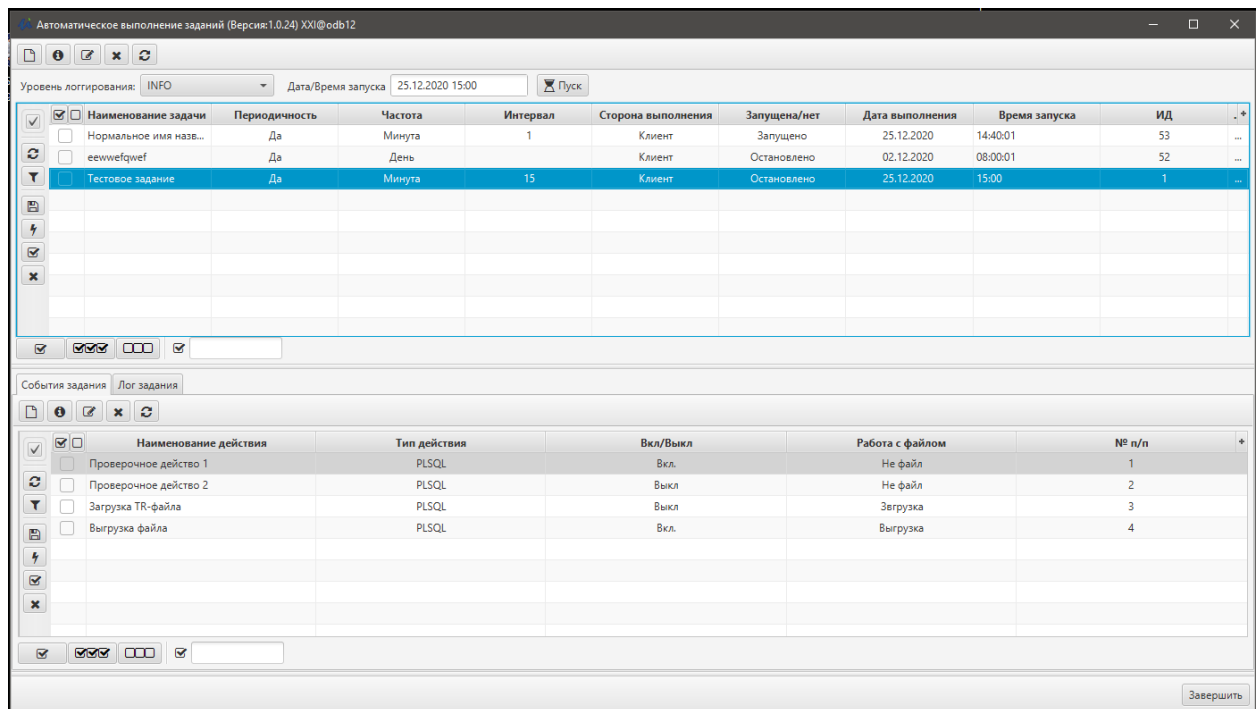
1) PLShed.jar – Это основной модуль.

Библиотеки должны быть размещены в папке /JAPP/lib

- manifest-reader.jar
- lovUtils.jar
- tools.jar

4. Начало работы.

Основное окно программы разделено на две области и представляет из себя отображение двух связанных таблиц в которых отображаются Задачи и события каждой задачи.



Каждая таблица имеет панель управления:



С помощью данной панели вы можете выполнять следующие действия над строками таблиц:

- Создать новый.
- Просмотр.
- Редактирование.
- Удаление.
- Обновить данные таблицы.



Так же обе таблицы имеют стандартную панель фильтра.

Которая позволяет производить стандартные инверсионные операции фильтрации над данными таблицами.

- Установка фильтра.
- Удаление фильтра.
- Сохранить фильтр.
- Установить фильтр по умолчанию.

Таблица заданий содержит так же содержит командно – информационную панель.

Уровень логгирования: **INFO** Дата/Время запуска **13.01.2021 12:13** **Пуск**

На данной панели можно установить:

- Уровень логгирования для текущей задачи.
- Принудительно запустить задачу вне расписания.
- Посмотреть дату время следующего запуска текущей задачи.

Также на основном экране имеется таблица логов заданий:

События задания **Лог задания**

При переключении вкладки Лог задания мы увидим таблицу логов:

Автоматическое выполнение заданий (Версия: 1.0.24) XH@odb12

Уровень логгирования: **INFO** Дата/Время запуска **13.01.2021 10:02** **Пуск**

Наименование задачи	Периодичность	Частота	Интервал	Сторона выполнения	Запущена/нет	Дата выполнения	Время запуска	ИД
Нормальное имя наз...	Да	Минута	1	Клиент	Запущено	25.12.2020	14:40:01	53
eevwefqwe	Да	День		Клиент	Остановлено	02.12.2020	08:00:01	52
Тестовое задание	Да	Минута	15	Клиент	Остановлено	25.12.2020	15:00	1

События задания **Лог задания**

№	Тип события	Текст	Дата	Логин пользователя	ИД сессии
1	INFO	Начинаем работу по заданию 53	25.12.2020 14:54	XOI	102501136
2	INFO	Завершаем задание 53	25.12.2020 14:54	XOI	102501136
1	INFO	Начинаем работу по заданию 53	26.12.2020 11:00	XOI	102549826
2	INFO	Завершаем задание 53	26.12.2020 11:00	XOI	102549826
1	INFO	Начинаем работу по заданию 53	28.12.2020 09:54	XOI	102620599
2	INFO	Завершаем задание 53	28.12.2020 09:54	XOI	102620599
1	INFO	Начинаем работу по заданию 53	30.12.2020 10:02	XOI	102750569
1	INFO	Начинаем работу по заданию 53	30.12.2020 10:49	XOI	102750674
1	INFO	Начинаем работу по заданию 53	30.12.2020 11:10	XOI	102750733
1	INFO	Начинаем работу по заданию 53	09.01.2021 12:43	XOI	106139778

Завершить

Таблица логов содержит свой более удобный фильтр:

Дата ИД сессии

Однако можно воспользоваться и стандартным фильтром.

Для формирования новой или редактирования существующей задачи открывается окно редактирования, которое имеет следующий внешний вид:

Редактирование

Наименование задачи: Тестовое задание

Периодичность: Да Частота: Минута Сторона выполнения: Клиент Запущен/Остановлен: Остановлено

Дата выполнения: 25.12.2020 Время запуска: 15:00:00 Интервал: 15 Событие запуска: Таймер

OK Отмена

После создания задачи необходимо сформировать перечень заданий для этого нажимаем кнопку создать на командной панели таблицы заданий:

Редактирование

Наименование действия: 11111111111111111111

№ п/п: 1

Тип действия: script Работа с файлом: Не файл Вкл/Выкл: Вкл.

ID пресета:

Тестировать код Результат предыдущего события:

Текст скрипта:

```
1 String result = "OK";
2 String fileName = null;
3 int counter = 10;
4
5 fileName = (String) CallSqlFunc(connection, "XXI.IKP_ADM.GetEventFileName", 1);
6 /** Эту строку можно убрать она только для отладки */
7 System.out.println(String.format("fileName = %s", fileName));
8
9 while(!isFileExist(fileName) && counter-- > 0 )
10     wait(10);
11
12 return isFileExist(fileName) ? result : "StopTask";
```

Компиляция Вывод

Входящая директория: Исходящая директория: Архивная директория:

OK Отмена

После настройки всех параметров задания и событий. Задание автоматически начнет запускаться в соответствии с настройками.

5. Системные пакеты модуля «Автоматический запуск заданий»

IKP_ADM

Основной пакет для работы с модулем. Содержит ряд функций, использующихся в интерфейсе для запуска заданий и действий в них:

Наименование функции	Назначение
FUNCTION initTask(taskID NUMBER) RETURN NUMBER	Инициализация задания (очистка всех переменных, используемых в пакете в рамках сессии работы с записью в журнал выполнения)
FUNCTION runTask(taskID NUMBER) RETURN NUMBER	Запуск задания (установка флага выполнения задания для исключения параллельного запуска с записью в журнал выполнения)
FUNCTION init_event(NPP NUMBER) RETURN number	Инициализация действия в рамках задания (установка переменных с номерами предыдущего/текущего действия, очистка временной таблицы для работы с файлами – загрузки/выгрузки с записью в журнал выполнения)
FUNCTION ExecEvent (NPP NUMBER) RETURN NUMBER	Запуск действия (выполнение PL/SQL блока с записью в журнал выполнения)
PROCEDURE finishTask(taskID NUMBER)	Завершение задания (снятия флага выполнения задания с записью в журнал выполнения)

Кроме системных функций, в пакете есть ряд переменных и функций/процедур, которые могут быть использованы в PL/SQL блоках действий в рамках заданий:

Наименование функции	Назначение
PROCEDURE SetEventFileName (NPP NUMBER , FN VARCHAR2)	Установка имени файла для действия № <NPP> (в основном используется при загрузке файлов – клиентская часть передает серверной имя загруженного файла для дальнейшей обработки)
FUNCTION GetEventFileName (NPP NUMBER) RETURN VARCHAR2	Получение имени файла для действия № <NPP> (в основном используется при выгрузке файлов – серверная часть передает в клиентскую имя файла, который необходимо выгрузить)

FUNCTION GetEventErrM (NPP NUMBER) RETURN VARCHAR2	Получение сообщения об ошибке (либо о результате работы) действия № <NPP>
PROCEDURE set_event_param(npp number , p_param_name in varchar2 , dval in date); PROCEDURE set_event_param(npp number , p_param_name in varchar2 , nval in number); PROCEDURE set_event_param(npp number , p_param_name in varchar2 , cval in varchar2);	Процедуры для установки дополнительных параметров действия № <NPP>, p_param_name – название доп.параметра (строковое, произвольное), dval/nval/cval – значение доп.параметра в зависимости от типа данных
FUNCTION get_event_param_c(npp number , p_param_name in varchar2) return varchar2 ; FUNCTION get_event_param_n(npp number , p_param_name in varchar2) return number ; FUNCTION get_event_param_d(npp number , p_param_name in varchar2) return date ;	Функции получения значения дополнительных параметров действия № <NPP> в зависимости от типа данных доп.параметра

IKP_DEBUG

Пакет IKP_DEBUG используется для ведения журнала работы модуля. В нем устанавливается уровень логгирования для текущей сессии работы модуля (процедура IKP_DEBUG.SetLvl), а также реализована основная процедура для занесения записи в журнал работы.

Процедура установки уровня логгирования:

PROCEDURE SetLvl (lvl **number**)

Входящий параметр lvl – устанавливаемый уровень логгирования, может принимать значения от 1 до 4, можно пользоваться константами из пакета:

```

lvlALL number(1,0) := 1;
lvlDEBUG number(1,0) := 2;
lvlINFO number(1,0) := 3;
lvlERROR number(1,0) := 4;

```

IKP_DEBUG.SetLvl(IKP_DEBUG.lvlINFO) – установка уровня INFO.

Штатно установка уровня производится в интерфейсе модуля, однако можно менять его и из PL/SQL блоков действий в рамках задания (к примеру, все задание выполняется с одним уровнем INFO, установленным в модуле, а одно из действий необходимо отладить более детально, тогда в начале PL/SQL блока данного действия сохраним текущий уровень (IKP_DEBUG.curLvl), установим нужный нам уровень отладки (к примеру, lvlDEBUG), в конце – вернем уровень, сохраненный ранее.

Процедура вывода строки в журнал отладки:

PROCEDURE LogIt (lvl **number**, msg **varchar2**);

Входными параметрами являются уровень отладки для строки и само отладочное сообщение. ID задания, пользователь, сессия будут определены автоматически внутренними механизмами модуля. Если заданный в процедуре вывода уровень отладки ниже, чем заданный в модуле – сообщение в журнал добавлено не будет. Т.е. если текущий уровень отладки INFO – сообщения с уровнями DEBUG или ALL в журнал не попадут, т.о. можно добавлять данный вывод в PL/SQL блоки и вызываемые в них функции/процедуры без опасений вывода лишних данных в журнал работы.

Максимальная длина отладочного сообщения – 4000 символов. Если размер сообщения будет превышать это значение – при занесении сообщения в журнал оно будет обрезано до 4000 символов.