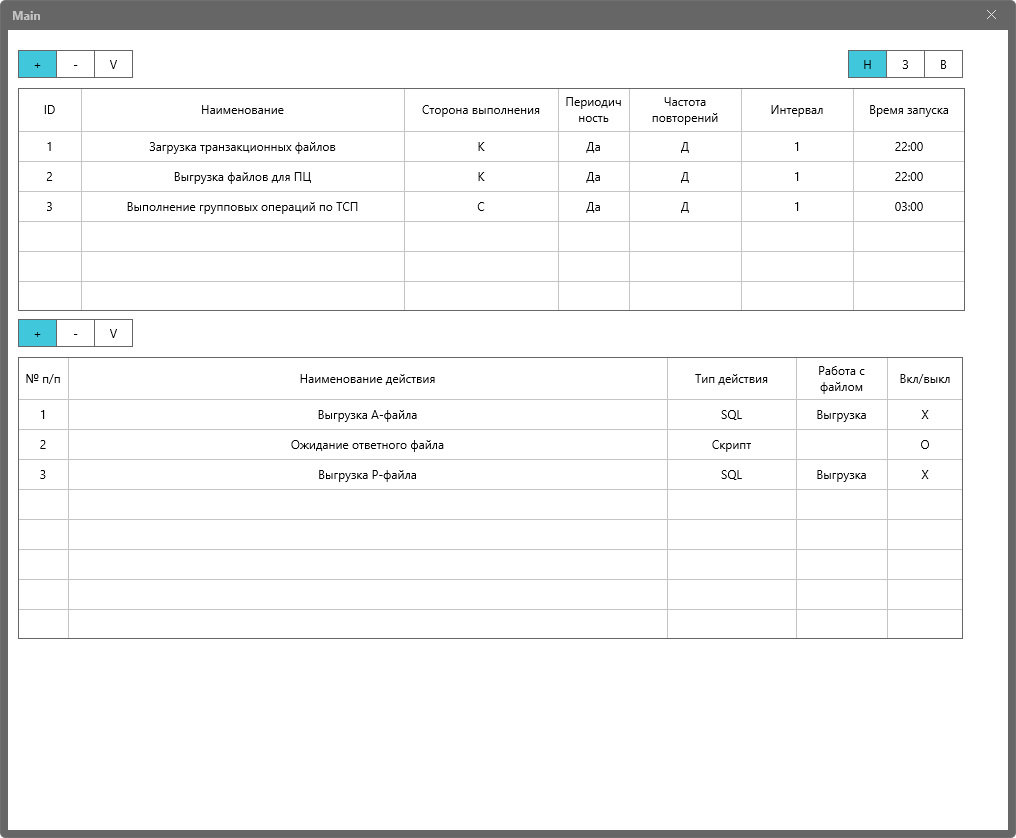
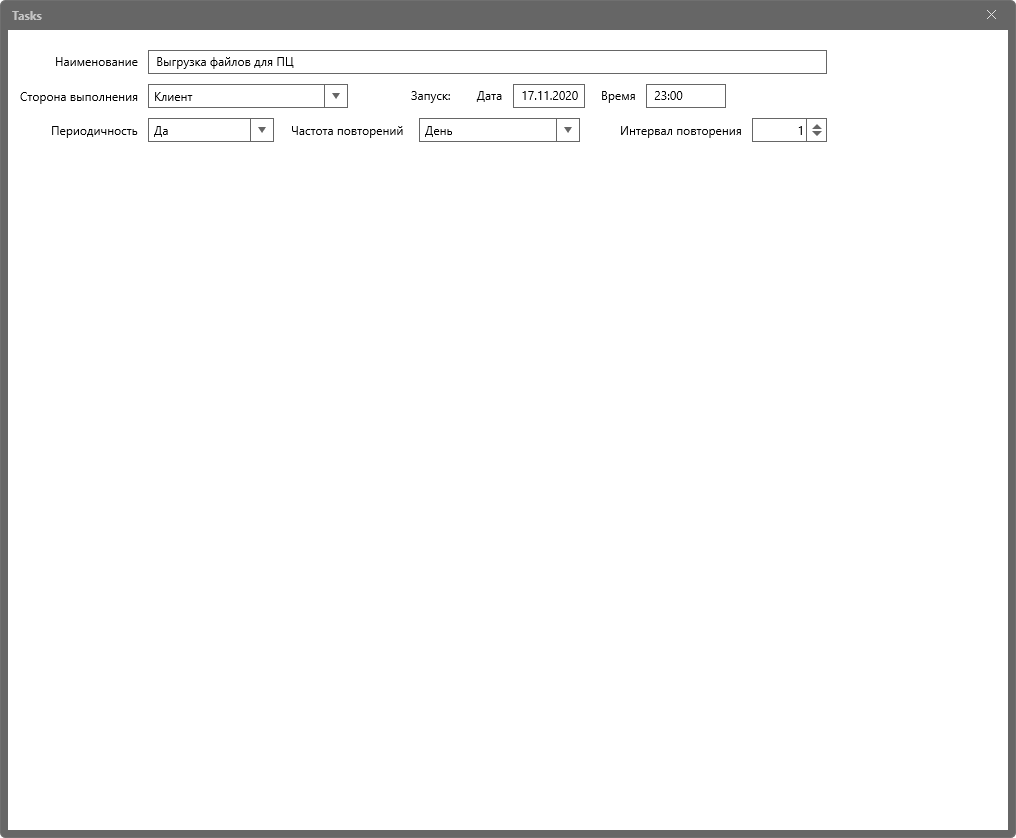
**Основная форма модуля:**



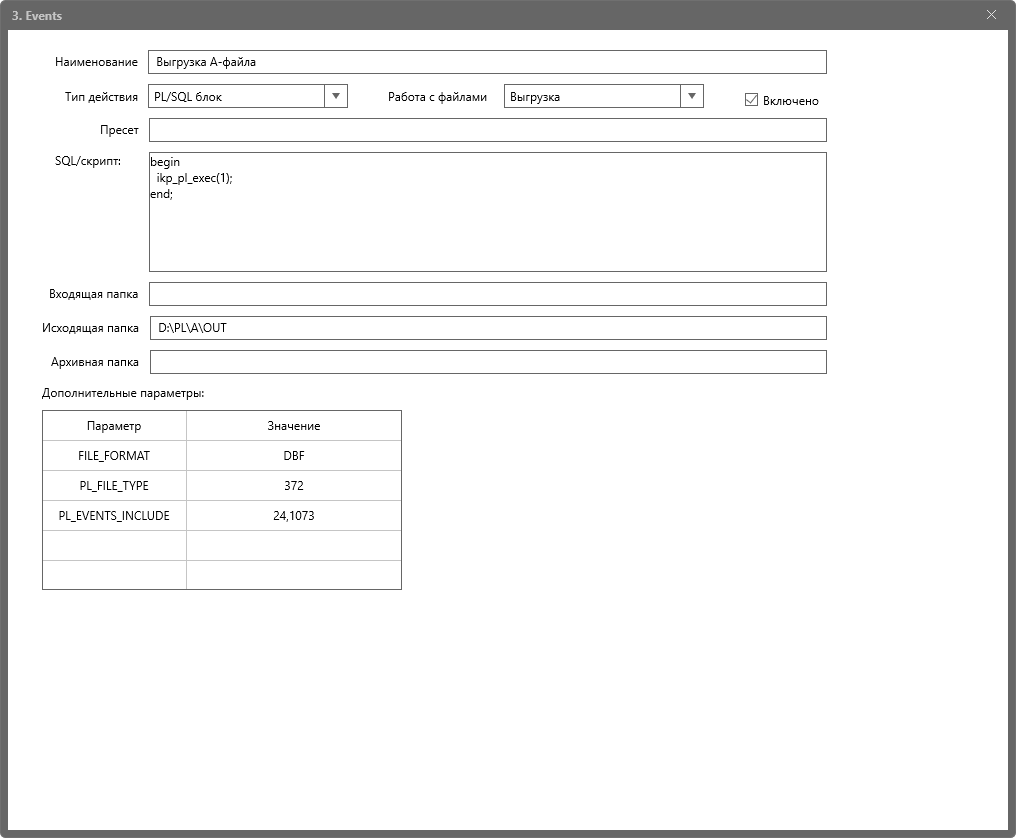
Две таблицы – задания и действия в них. Для каждой таблицы тулбар (добавить/удалить/редактировать). Справа сверху общий тулбар – настройки/запуск задания/выход – чисто для примера, могут быть отдельные кнопки, может быть меню и т.д.

**Форма добавления/редактирования задания:**



Задаем основные параметры задания. Сторона выполнения – Клиент/Сервер. Периодичность – Да/Нет. Частота повторений – Месяц/Неделя/День/Час/Минута. Интервал повторений задается количеством предыдущей величины (1 День, 15 минут и т.п.)

**Форма добавления/редактирования действия:**

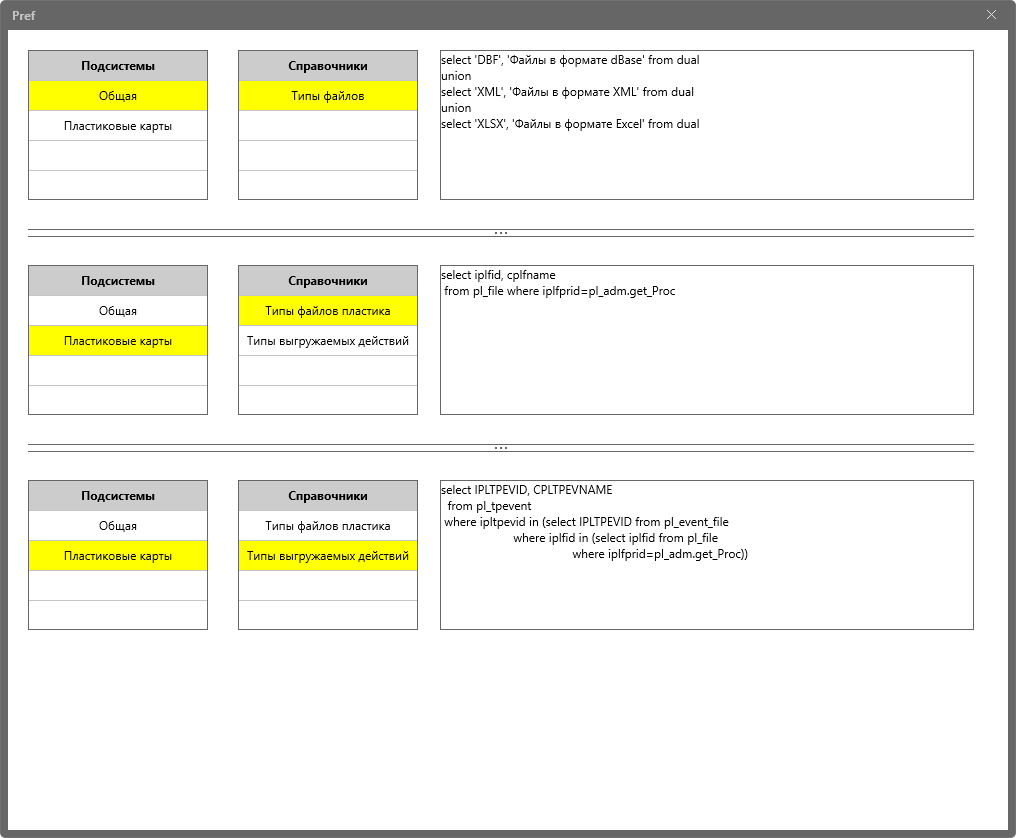


Задаются общие параметры действия – тип (PL/SQL, скрипт, команда OS, пресет). Работа с файлами – Выгрузка/загрузка/не файл. Признак вкл/выкл – выполнять ли в общем цикле по заданию. Пресет – в случае типа действия «Пресет» – выбор из справочника пресетов (пока отложим). В мемо-поле SQL/скрипт – задаем текст sql-блока или скрипта для соответствующего типа действия. Папки для работы с файлами – понятно.

Дополнительные параметры – тут интереснее. Связанная таблица IKP\_EVENT\_PARAMS с таблицей действий IKP\_TASK\_EVENTS по eventID. По сути 2 поля – наименование параметра и его значение, оба текстовые. Можем иметь обязательный набор параметров (для пластиковых загрузок/выгрузок например – тип файла, формат, для выгрузки – какие события пластиковые выгружаем и т.д.), такие наборы можно контролировать на наличие и заполнение при старте выполнения действия. Никто не запрещает добавлять свои параметры (название/значение) и при запуске блоком каких-то своих пользовательских функций также в общем списке – получать значения этих параметров через пакет IKP\_ADM.

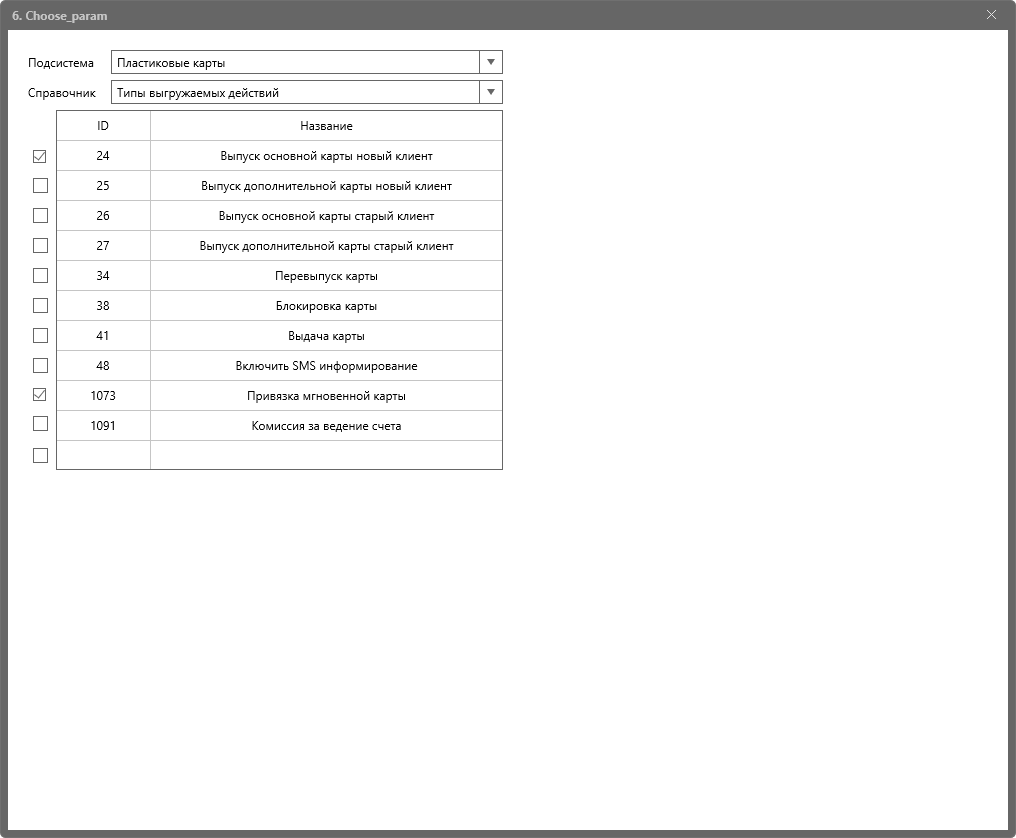
Для автоматизации заполнения значений параметров привяжем справочники, которые можно построить по примеру курсоров для параметров в альт.печати.

**Форма настройки справочников:**



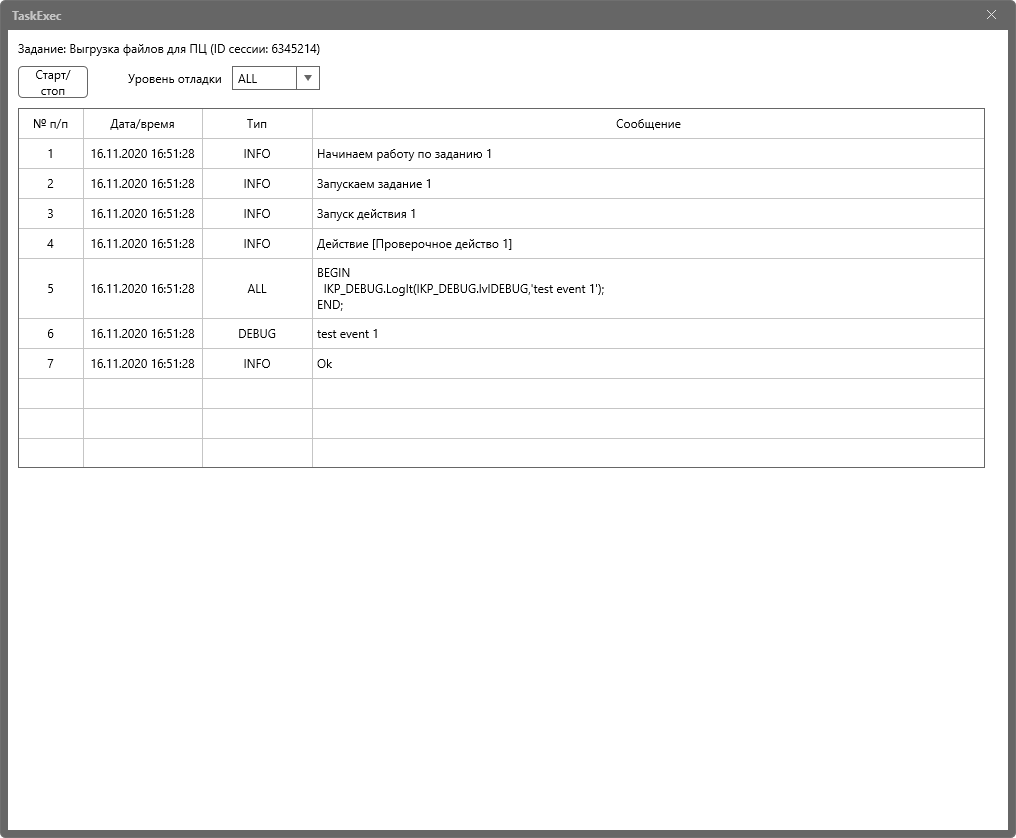
В один скрин собрал 3 примера (активная строка в таблицах подсвечена желтым). В первом случае – набор констант, второй и третий – запросы разного уровня сложности. Общий принцип построения запросов для построения справочника – возвращает 2 поля, 1 поле уходит в значение параметра, второе – комментарий.

Пример выбора значения доп.параметров из справочника:



При заполнении значения параметра из справочника появляется эта форма, где пользователь выбирает подсистему, справочник и видит перечень доступных значений. При выборе нескольких – возвращаем в поле «Значение параметра» значения первого поля из справочника через запятую.

**Форма запуска задания:**



Тут имеем информационный блок (для примера тут название задания, ИД сессии, можно еще придумать что выводить).

Далее – кнопка старт/стоп и выбор уровня отладки. При изменении значения комбобокса зовем процедуру IKP\_DEBUG.SetLvl(уровень).

Ниже на форме отображаем таблицу отладки IKP\_LOG запросом

**select** strno, strype, dt, msgtxt

**from** ikp\_log

**where** sessid=:SessionID **order** **by** strno

Т.е. по ИД сессии выбираем чего понаписали в отладку по заданию.

Можно обновлять запросик для отображения текущей ситуации после каждого шага в цикле инициализации/выполнения действий в рамках задачи – так по шагам сразу будет видно что происходит.