Модуль «Автоматическое выполнение заданий Пластикового Модуля»

Модуль предназначен для автоматизации обмена файлами с ПЦ (загрузка файлов с последующей обработкой, выгрузка файлов, выполнение сопутствующих задач по обработке данных).

Модуль должен быть представлен, как клиент-серверное решение.

Разработка должна вестись на Java, с использованием библиотек/API, предоставляемых компанией Инверсия.

В клиентской части должен быть реализован интерфейс по настройке модуля, управлению заданиями, просмотру журналов выполнения заданий, также на стороне клиентского модуля должны быть решены некоторые задачи по работе с файлами.

Серверная часть отвечает за обработку данных – разбор загруженных файлов и дальнейшая обработка полученных данных; подготовку данных для выгрузки и приведение этих данных в необходимый для выгрузки вид (формат).

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

Атомарная 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 (JS, Perl и т.д. – нужно определиться что мы можем сделать), 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 может иметь входящие и исходящие параметры, последующая атомарная задача может использовать в качестве входящего параметра исходящий параметр из предыдущей.

У атомарной задачи должен быть признак «Дожидаться завершения задачи», т.е. переход к следующей по порядку задаче осуществляется после завершения предыдущей или не дожидаясь его. К примеру, одной из задач в списке может быть архивирование сформированного файла. Мы можем запустить этот процесс скриптом и сразу переходить к следующему пункту, не дожидаясь окончания архивирования.

Таким образом, для каждой атомарной задачи могут быть заданы следующие параметры:

1. Наименование
2. Тип задачи
3. Исходя из типа задачи – предустановленное действие/PLSQL блок/скрипт/команда ОС.
4. Входящие и исходящие параметры
5. Признак ожидания завершения
6. Входящая директория
7. Исходящая директория
8. Архивная директория
9. Признак вкл/выкл

Каждое задание должно иметь параметры:

1. Наименование
2. Период выполнения
3. Интервал повторения
4. Сторона выполнения (клиент/сервер)
5. Признак вкл/выкл

Пояснения для п.4:

1. Задание может быть не связанным с файлами. Например, выполнение групповых операций после загрузки всех входящих файлов, таких как формирование действий по торговым точкам, начисление комиссий и т.д. Подобные задачи не требуют каких либо действий на стороне клиента и могут быть оформлены как JOB.
2. Есть возможность загрузки и выгрузки файлов на стороне сервера (с использованием ORACLE DIRECTORY), в этом случае мы также можем не использовать клиентскую часть постоянно, а вынести задание в JOB. Однако, в данном случае мы сталкиваемся с позицией службы безопасности банка – ведь для такого подхода на сервере должна быть примонтирована общая папка, через которую будет происходить обмен файлами. Если безопасность согласна с таким подходом – нагрузка на клиентскую часть ограничится только настройками, если нет – значит вся работа с файлами будет на клиенте.

Открытые вопросы:

В случае реализации работы с файлами (загрузка/выгрузка) на стороне клиента сама клиентская часть должна быть постоянно запущена. Как реализовать параллельное выполнение нескольких заданий одновременно в одном клиентском приложении? Разделить в разные потоки? Один поток работает по расписанию задания №1 (загрузка TR-файлов к примеру), второй поток – задание №2 (формирование и выгрузка файлов для ПЦ с учетом условий – формировать следующий файл после получения подтверждения об обработке в ПЦ предыдущего (существует такая задача в Т1, нужно после формирования одного файла ждать пока в определенной папке появится файл определенным именем – ответ от ПЦ, после этого приступаем к формированию следующего файла)). По-хорошему, для соединения с БД эти потоки также должны иметь разные коннекты, разные сессии.

В каком объеме переносим логику по загрузке/выгрузке файлов с форм пластика в пакеты? Понятно, что начинать надо с Т1, можно обкатать Т2 и абхазов, крымчане тоже в принципе доступны. Остальные ПЦ – нет ни примеров, останется только расписать схему как реализовывать механизм (благо, все выносится в пакеты, и можно вызывать любые самописные или готовые функции/процедуры).