Задание к модулю Messaging

# Общее

В данной работе мы несколько расширим службу обработки результатов потокового сканирования, которую разрабатывали в модуле Windows Services.

Традиционно принято подобного рода системы разбивать на отдельные независимые службы, которые могут устанавливаться как на один компьютер, так и на разные. Например, может существовать следующая конфигурация:

* **Сервера захвата (ввода) документов**. Обычно их бывает несколько, и они устанавливаются на разные компьютеры – туда, где происходит ввод документов (изображений). Их задача – собирать документы и передавать на сервера трансформации
* **Сервера трансформации**. Их также может быть несколько, но по другой причине – таким образом балансируют нагрузку. Такие сервера могут производить различные типы обработок: конвертация, извлечение текста (OCR), отправка в СЭД, …
* **Центральный управляющий сервер**. Как правило он один и его задача – мониторинг состояния работы остальных серверов и передавать им настройки.

Мы будем реализовывать чуть более простую модель, состоящей из 2-х элементов: служб ввода (на базе уже реализованной службы для потокового сканирования) и центрального сервера. Очередь должна базироваться на MSMQ.

# Задание 1. Централизация сбора результатов работы служб ввода

Разработай службу центрального сервера, которая будет делать следующее:

* При установке (или первом запуске) создавать очередь для приема результатов (готовых документов) от служб ввода
* Слушает эту входящую очередь и сохраняет на диск все пришедшие документы

Доработайте службу ввода так, чтобы вместо того, чтобы сохранять готовые документы на диск, она отсылала их через службу сообщений на центральный сервер.

Для упрощения архитектуры примем, что в сети одновременно могут работать несколько агентов (на разных компьютерах), но только 1 центральный сервис.

**Примечание!!!** Одна из сложностей данного задания: лимит на размер одного сообщения. Суть в том, что как правило, очереди сообщений лимитируют размер одного сообщения (особенно это касается облачных платформ) и получаемый файл, скорее всего, этим лимитам удовлетворять не будет.

Для пересылки больших объемов данных в очереди (там, где имеется лимит и невозможно использовать другие способы передачи) используется подход [Message Sequence](http://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageSequence.html) (по ссылке приведен только рисунок, но общую идею он проясняет).

# Задание 2. Механизм централизованного контроля и управления

Разработайте механизм централизованного управления через очереди сообщений.

В рамках этого механизма:

* От служб ввода на центральный сервер с некоторой периодичностью приходит текущий статус службы:
  + Чем занята служба (ждет новых файлов/обрабатывает последовательность/обнаружен новый документ…)
  + Текущие настройки:
    - Таймаут ожидания очередной страниц
    - Шаблон имени документа
* От сервера ко всем службам уходят команды:
  + Обновить статус (т.е. не дожидаясь времени обновления статуса)
  + Поменять настройки:
    - Таймаут ожидания очередной страниц
    - Шаблон имени документа

Для упрощения предлагается не разрабатывать никакого UI (хотя это не обязательное требование – если удобнее разработать UI, то можно сделать и так):

* Статусы служб центральный сервер пишет в EventLog
* Для рассылки новых настроек просто следит за обновлением локального файла с настройками – если он поменялся, то рассылает новые настройки.