

Инструкция по работе с сервисом VIDAU Quantum Connector v.1.0.

Общая информация

ПО Quantum Connector предназначено для транслирования запросов от системы Quantum PFR (SOAP-запросы) в обновлённый интерфейс StorNext REST API и обратной трансляции ответов на запросы. Таким образом, обеспечивается совместимость Quantum PFR и Quantum StorNext версии 6.2.0 и выше.

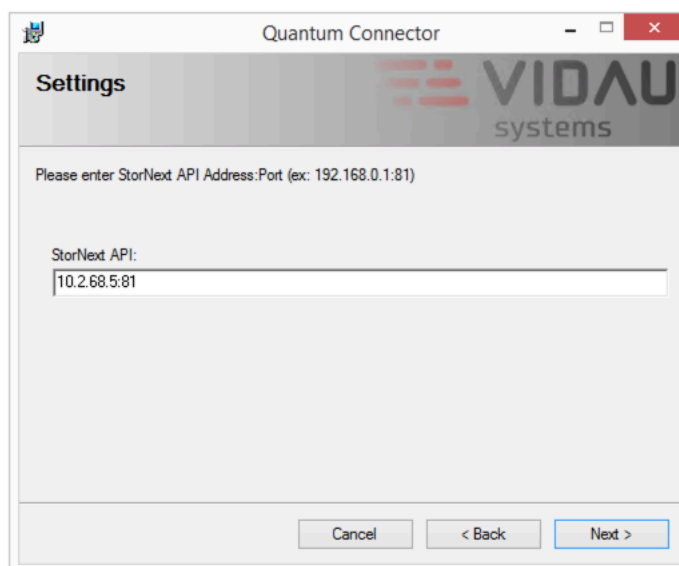
Установка

Перед установкой ПО, убедитесь что на сервере установлен пакет Microsoft .NET Framework версии 4.6.1 или более поздней.

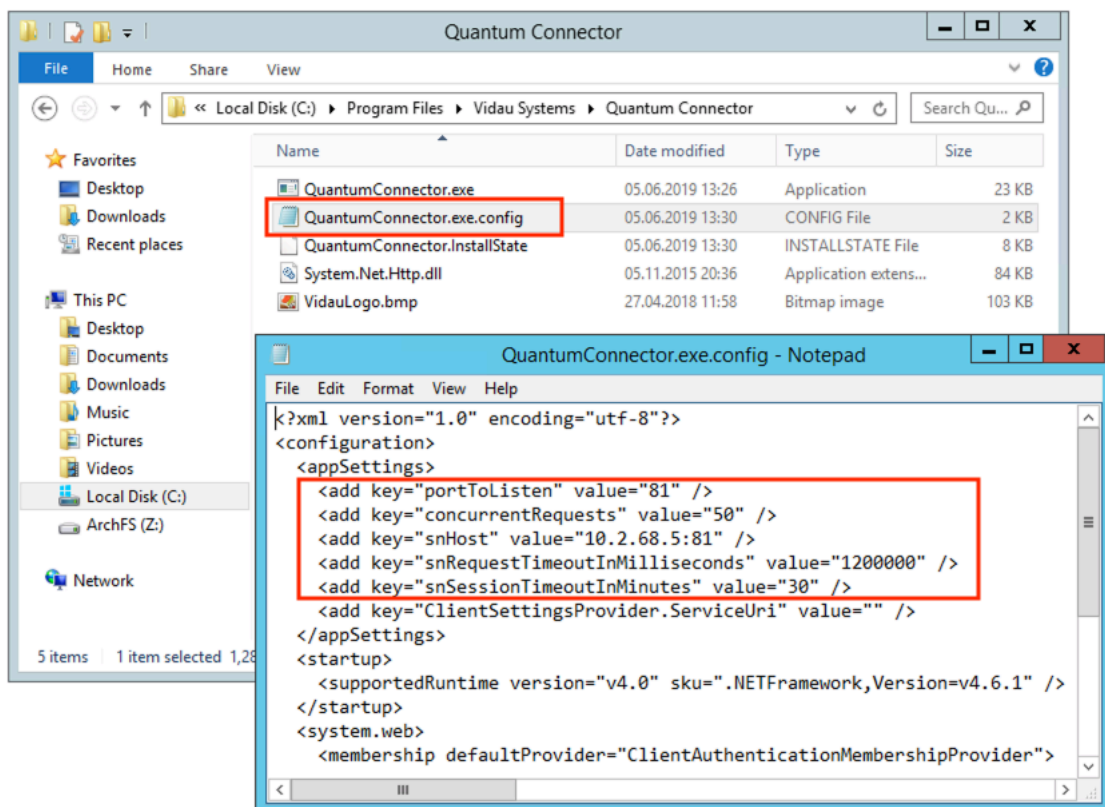
Для установки сервиса запустите инсталляционный пакет Quantum Connector Setup.msi и следуйте появляющимся на экране инструкциям.

Во время установки от вас потребуется ввести адрес (имя хоста или IP) и порт StorNext API (StorNext Web Services V2).

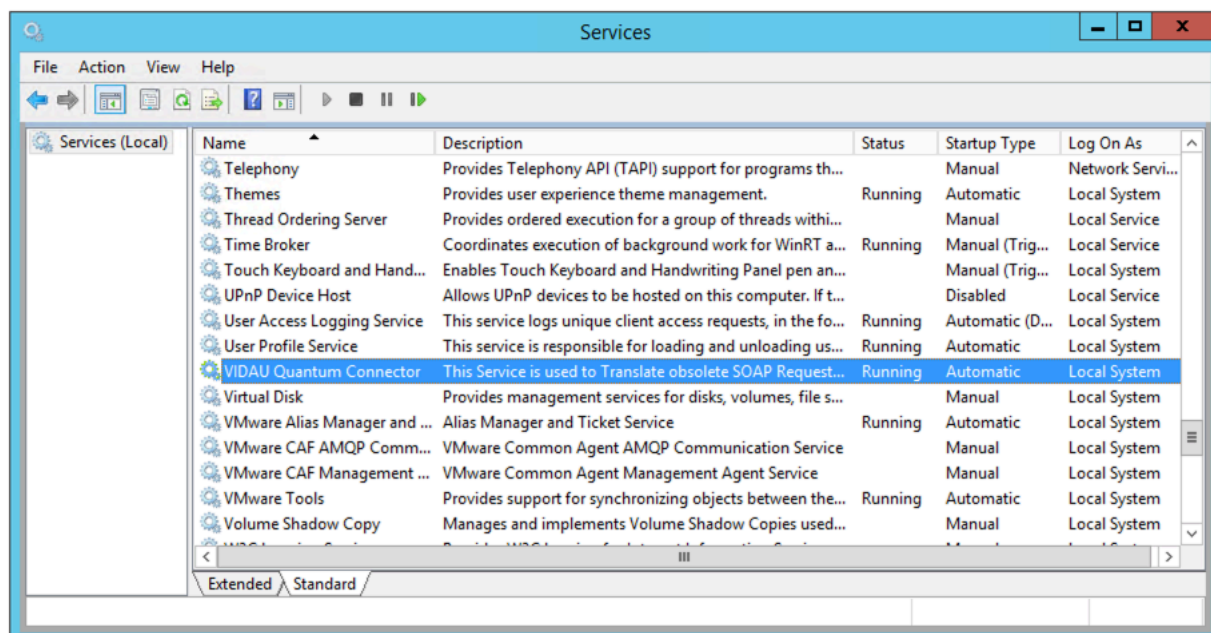
После ввода значений в окне инсталлятора, можно нажать кнопку «Next»:



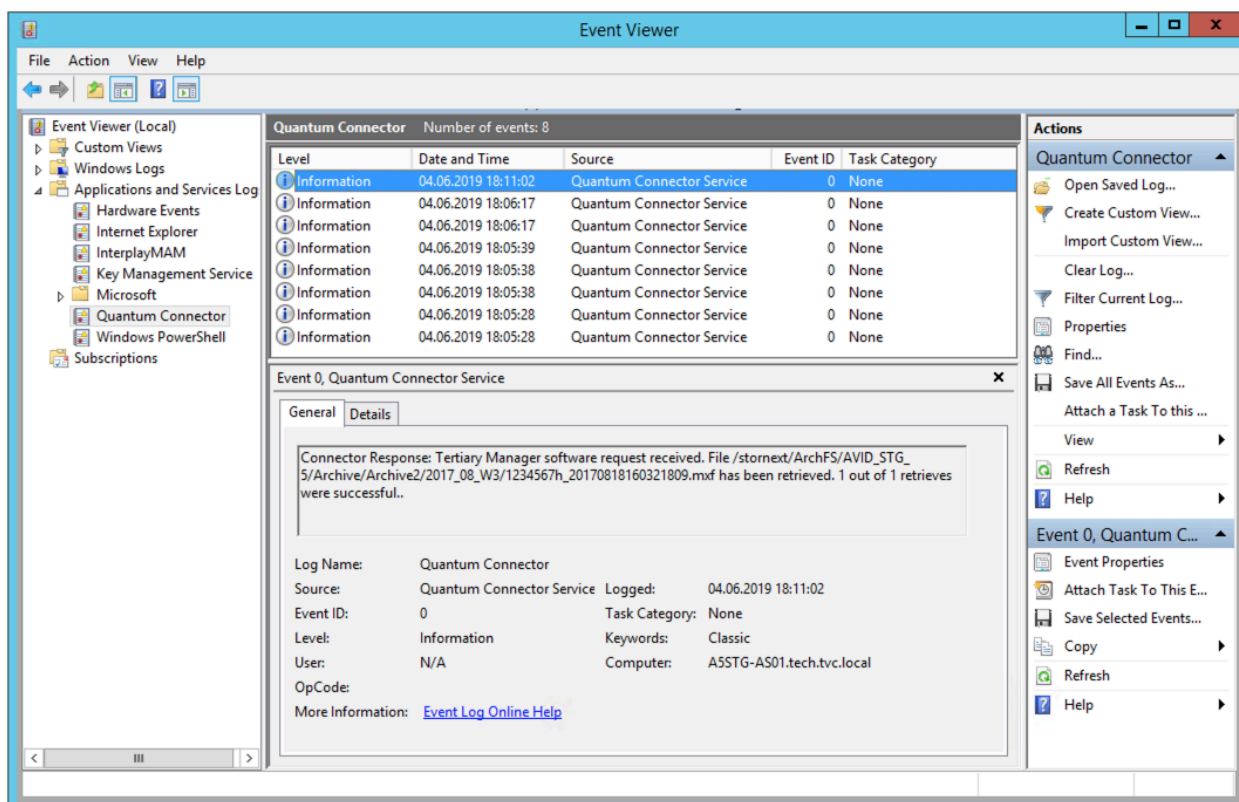
(!) Внимание: при установке сервиса, значение адрес:порт не проверяется на актуальность, а просто сохраняется в файле конфигурации. Этот файл можно будет найти после установки в папке C:\Program Files\Vidau Systems\Quantum Connector:



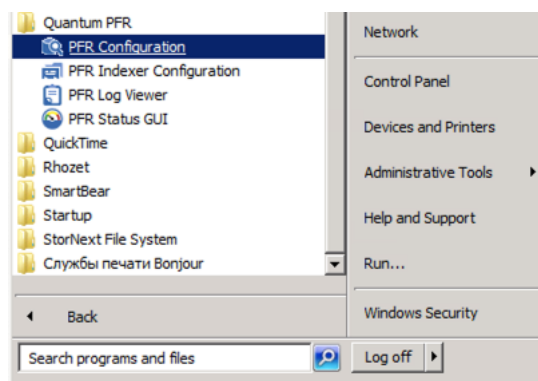
Если возникла необходимость поменять эту настройку или другие имеющиеся настройки, то сделать это можно при помощи любого текстового редактора. Для вступления изменений в силу необходимо выполнить перезапуск сервиса VIDAU Quantum Connector. Этот сервис должен появиться после инсталляции в списке работающих сервисов Windows и иметь статус Running:



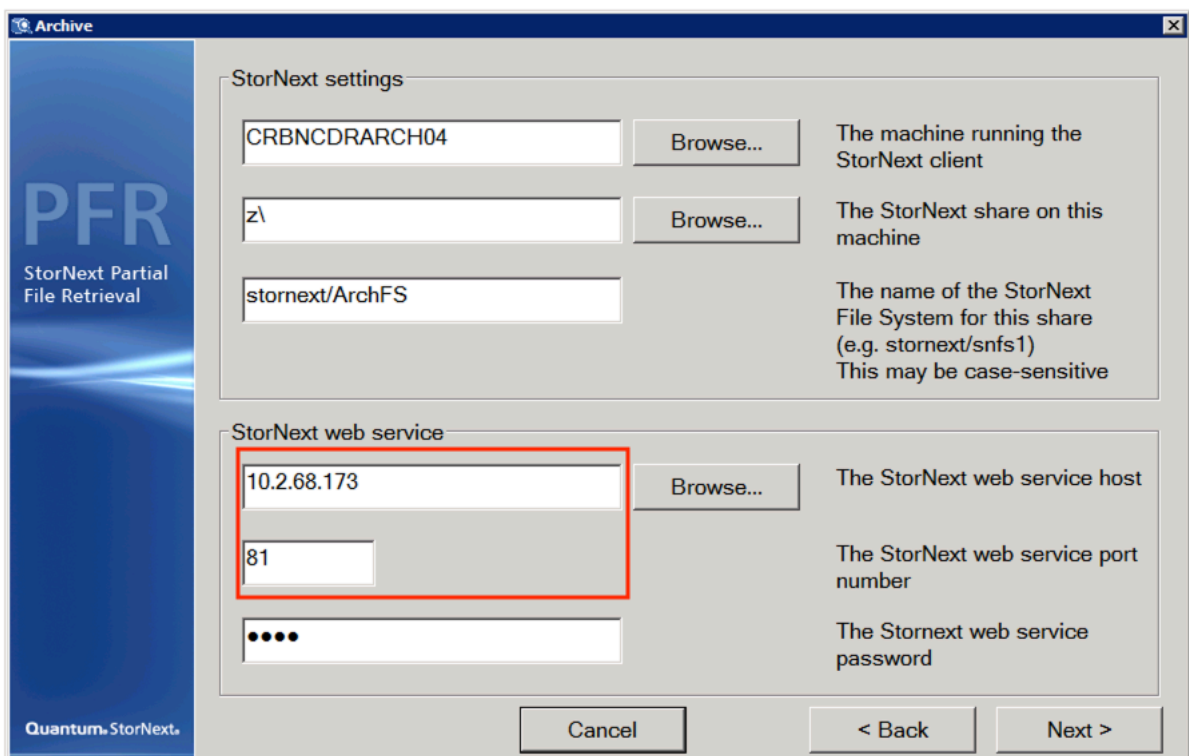
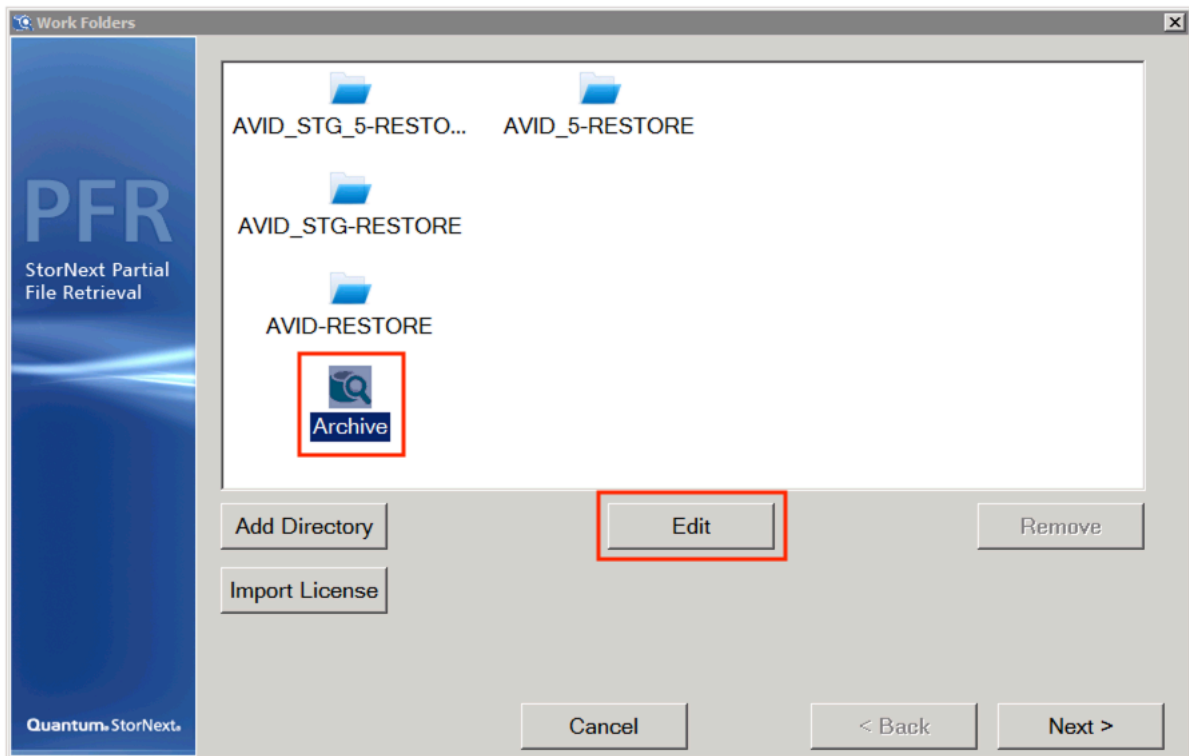
Работающий сервис записывает основные события в отдельный журнал, который можно найти в административной утилите Windows Event Viewer (Event Viewer (Local)\Application and Services Logs\Quantum Connector):



После установки ПО Quantum Connector, необходимо выполнить настройку системы Quantum PFR, запустив для этого PFR Configuration:



Далее следует открыть настройки ресурса «Archive» и изменить значения полей «The StorNext web service host» и «The StorNext web service port number» на соответствующие адресу сервера, на котором установлен Quantum Connector и номеру порта, который открыт для запросов в Quantum Connector (по умолчанию это порт 81, но в случае конфликтов с другим ПО, это значение может быть изменено на произвольное):



Удаление

Для удаления сервиса можно воспользоваться тем же установочным пакетом Quantum Connector Setup.msi или стандартной утилитой панели управления Control Panel\Programs\Programs and Features.

Настройка

В процессе установки Quantum Connector существует только одна настройка адреса StorNext API, упомянутая выше. Для нормальной работы сервиса достаточно корректно указать этот параметр. Остальные настройки могут понадобиться в случае нестабильной работы сервиса и осуществляются посредством ручного изменения файла конфигурации QuantumConnector.exe.config и перезапуска сервиса.

В приведенной ниже таблице перечислены все параметры, доступные для изменения:

Key	Value	Type	Описание
portToListen	81	Integer	Порт, по которому сервис принимает запросы от Quantum PFR. Такое же значение должно быть установлено в настройках PFR Configuration, как показано в разделе «Установка». Номер порта может быть изменён, если возникают конфликты с другим программным обеспечением.
concurrentRequests	50	Integer	Количество запросов от Quantum PFR, которые сервис может обработать параллельно. Каждый запрос обрабатывается в отдельном потоке. Если в единицу времени поступит количество запросов, превышающее значение concurrentRequests (от множества параллельных задач частичного восстановления), то сервис ответит ошибкой Internal Server Error (500).
snHost	192.168.0.1:81	String	Адрес и порт хоста, на котором работает StorNext API (StorNext Web Services V2).
snRequestTimeoutInMilliseconds	1200000 (20 минут)	Integer	Таймаут единичного запроса к StorNext API в миллисекундах, после которого сервис ответит ошибкой Internal Server Error (500). При этом сам запрос может быть успешно выполнен системой StorNext позднее.
snSessionTimeoutInMinutes	30 (30 минут)	Integer	Время жизни одной сессии обработки запроса в минутах. Этот параметр используется для контроля над потенциально «зависшими» сессиями, т.к. каждая из них обрабатывается в отдельном потоке. По истечении заданного периода времени сессия принудительно закрывается.