Тестовый сервер. Описание

Формат конфигурационного файла

1) Комментарии начинаются с символа ‘#’ и заканчиваются концом строки (мы рассматриваем только такие комментарии, которые занимают всю строку целиком). Комментарии в разборе конфигурационного файла не участвуют.

2) В первых 4 строках (без учета комментариев) расположены следующие параметры (в том же порядке, в котором они приведены здесь):

* интервал между обработкой сообщений в миллисекундах;
* IP адрес слушающего сокета сервера;
* номер TCP порта (номер порта потока управления)
* номер UDP порта (номер порта потока данных).

3) Во всех следующих строках (без учета комментариев) расположены данные сообщений следующего вида: *Message-Type Data-Type Data*. Здесь *Message-Type* одно из следующих значений: *REQUEST*, *RESPONSE*, *DATA*, *EVENT*; *Data-Type* - одно из следующих значений: *HEX*, *BASE64*; *Data* - строка в формате hex (если *Data-Type* = *HEX*) или base64 (если *Data-Type* = *BASE64*). Формат hex - это строковое представление массива байт, в котором каждый байт представлен в 16-ном виде, занимает 2 символа (если значение байта меньше 16, то перед значением записывается ведущий 0) и не содержит разделителей между строковыми представлениями отдельных байтов.

Поведение при обработке текущего сообщения:

1) Если *Message-Type* = *REQUEST*, то тестовый сервер ожидает запрос от клиента через поток управления (через TCP соединение); при этом происходит сравнение данных из тела запроса и из текущего сообщения. Если данных из тела запроса и из текущего сообщения не совпадают, то клиенту посылается ответ (сообщение с *Message-Type* = *RESPONSE*) с пустыми данными; сервер при этом продолжает обрабатывать текущий запрос (ожидает от клиента запрос с данными, совпадающими с данными текущего сообщения). Если данных из тела запроса и из текущего сообщения совпадают и следующим сообщением в конфигурации является ответ (сообщение с *Message-Type* = *RESPONSE*), то именно он и будет послан клиенту. Если данных из тела запроса и из текущего сообщения совпадают и следующим сообщением в конфигурации является не ответ (сообщение с *Message-Type* = *RESPONSE*), то клиенту посылается ответ (сообщение с *Message-Type* = *RESPONSE*) с пустыми данными.

2) Если *Message-Type* = *RESPONSE* и предыдущее сообщение было не запрос (сообщение с *Message-Type* = *REQUEST*), то такое сообщение отбрасывается.

3) Если *Message-Type* = *DATA*, то такое сообщение посылается клиенту через поток данных.

4) Если *Message-Type* = *EVENT*, то такое сообщение посылается клиенту через поток управления.

После обработки очередного сообщения, если следующее сообщение имеет тип (*Message-Type*) *DATA* или *EVENT*, то это сообщение будет обработано (послано клиенту) через промежуток времени, равный интервалу между обработкой сообщений. После обработки очередного сообщения, если следующее сообщение имеет тип (*Message-Type*) *REQUEST*, то это сообщение будет обработано через неопределенный промежуток времени (когда клиент пришлет запрос на сервер). После обработки очередного сообщения, если следующее сообщение имеет тип (*Message-Type*) *RESPONSE*, то это сообщение будет сразу же отброшено.

Замечания по обработке конфигурационного файла:

Случай 2 поведения при обработке сообщений (если *Message-Type* = *RESPONSE* и предыдущее сообщение было не запрос) будет обрабатываться на этапе чтения конфигурационного файла: такое сообщение будет отброшено. Также на этапе чтения конфигурационного файла будет обрабатываться ситуация случая 1 (если *Message-Type* = *REQUEST*), когда следующим сообщением в конфигурации является не ответ (сообщение с *Message-Type* = *RESPONSE*): в список сообщений будет добавляться ответ (сообщение с *Message-Type* = *RESPONSE*) с пустыми данными.